Université Libre des Pays de Grands Lacs

Faculté de Sciences et Technologie Appliquée

Département de Génie Informatique

BP. 368 GOMA

www.ulpgl.net

Conception et réalisation d'un système de reconnaissance en temps réel de la langue des signes : cas de l'ASL

Présenté par : Mugisa Panza Emmanuel

Travail présenté en vue de l'obtention du Diplôme de Bachelor en Sciences de l'ingénieur

Mention : Génie Informatique

Directeur : DI. DR. Tech AKWIR Alain NKIEDIEL

Encadré par : Msc Ir MUGHOLE KALIMUMBALO Daniella

Année académique : 2024–2025

Épigraphe

"Le plus grand problème dans la communication, c'est l'illusion qu'elle a eu lieu."

George Bernard Shaw

Dédicace

À nos chers parents, MAKI PANZA et UKELA ADUBA, pour leur amour et leur soutien inébranlable.

Mugisa Panza Emmanuel

Remerciements

Au terme de ce mémoire, je tiens à exprimer ma profonde gratitude envers toutes les personnes qui, de près ou de loin, ont contribué à sa réalisation.

Mes premières pensées vont à Dieu Tout-Puissant, pour le don de la vie, sa grâce quotidienne et la force qu'il m'a accordée tout au long de ce parcours.

Je remercie sincèrement Monsieur le Professeur Alain AKWIR NKIEDEL, Directeur de ce mémoire, pour sa rigueur et ses orientations éclairées, ainsi que le Master Ir MUGHOLE KALIMUMBALO Daniella, mon encadreur, pour son accompagnement attentif et ses conseils avisés.

Ma reconnaissance va également aux autorités académiques de l'ULPGL-Goma, pour la qualité du cadre d'apprentissage offert, ainsi qu'à Monsieur le Professeur BARAKA Olivier et à tous les enseignants de la Faculté de Science et Technologie Appliquée, dont l'engagement a nourri mon parcours universitaire.

À mes parents bien-aimés, MAKI PANZA et UKELA ADUBA, je dédie une pensée toute particulière. Leur amour, leur patience et leur soutien inébranlable ont été les piliers silencieux de mon avancement.

Enfin, ma gratitude s'adresse à mes amis et collègues pour leur amitié, leur bienveillance et les nombreux moments de soutien partagés. Une mention spéciale à VUNDE LUKENGU Gratien, AKSANTI BYALAHIRE François, MAKENGU MUSAFIRI Benit, ainsi qu'à toutes celles et ceux, non cités ici, mais qui ont contribué, chacun à sa manière, à la réussite de ce travail.

À vous tous, merci.

Mugisa Panza Emmanuel

Résumé

La barrière linguistique entre les personnes entendantes et celles malentendantes ou sourdes reste un défi majeur pour l'inclusion sociale et professionnelle. Ce mémoire propose une solution technologique innovante pour répondre à ce besoin : un système de reconnaissance en temps réel de la langue des signes en texte, combinant MediaPipe Holistic, un pipeline de traitement vidéo asynchrone et l'architecture SPOTER. Nous avons mené une étude expérimentale basée sur la langue des signes américaine (ASL), utilisée comme benchmark de validation méthodologique, avec un pipeline allant de la capture vidéo à la prédiction du texte correspondant. L'architecture SPOTER, adaptée et entraînée sur des gestes manuels, a permis d'atteindre des résultats intéressants, notamment en précision et en réactivité. Les tests ont montré que notre système pouvait reconnaître les signes avec une précision de 80,61 % au Top-1 et de 94,35 % au test de précision Top-3, offrant ainsi un retour instantané compréhensible pour les personnes non signantes.

Ce travail a démontré que l'intégration de l'apprentissage profond avec la vision par ordinateur peut significativement améliorer la communication entre les communautés malentendantes et leur environnement. Il ouvre également des perspectives pour des systèmes embarqués ou mobiles dédiés à l'accessibilité universelle.

Mots-clés : Langue des signes, apprentissage profond, MediaPipe, traduction en temps réel, accessibilité, reconnaissance gestuelle.

Abstract

The linguistic barrier between hearing individuals and those who are hard of hearing or deaf remains a major challenge to social and professional inclusion. This thesis proposes an innovative technological solution to address this need : a real-time sign language recognition system that combines MediaPipe Holistic, an asynchronous video-processing pipeline, and the SPOTER architecture.

We conducted an experimental study based on American Sign Language (ASL), using a processing pipeline that ranges from video capture to the prediction of the corresponding textual output. The SPOTER architecture, adapted and trained on manual gestures, achieved promising results, particularly in terms of accuracy and responsiveness. Experimental evaluations showed that the proposed system can recognize signs with a Top-1 accuracy of 80.61% and a Top-3 accuracy of 94.35%, providing instant and comprehensible feedback for non-signing users.

This work demonstrates that integrating Deep Learning with computer vision can significantly enhance communication between deaf and hard-of-hearing communities and their surrounding environment. It also opens new perspectives for embedded or mobile systems dedicated to universal accessibility.

Keywords : Sign language, Deep Learning, MediaPipe, real-time translation, accessibility, gesture recognition.

Sommaire

Sigles et Abréviations

3D	Trois dimensions
API	Application Programming Interface (Interface de programmation applicative)
ASL	American Sign Language (Langue des signes américaine)
CV	Computer Vision (Vision par ordinateur)
DL	Deep Learning (Apprentissage profond)
FPS	Frames Per Second (Images par seconde)
GPU	Graphics Processing Unit (Processeur graphique, utilisé pour l'entraînement des IA)
LSC	Langue des Signes Congolaise
LSTM	Long Short-Term Memory (Type de réseau RNN pour les séquences)
MediaPipe	Bibliothèque Google pour la détection de mains, visages et poses
ML	Machine Learning (Apprentissage automatique)
MoCap	Motion Capture
OpenCV	Open Source Computer Vision Library (Bibliothèque de vision par ordinateur)
PyTorch	Framework d'apprentissage profond développé par Facebook (Meta)
RNN	Recurrent Neural Network (Réseau de neurones récurrents)
SPOTER	Sign POse-based TransformER
TensorFlow	Framework d'apprentissage profond développé par Google
UI	User Interface (Interface utilisateur)
UX	User Experience (Expérience utilisateur)
WLASL	Word-Level American Sign Language

Liste des tableaux

Liste des figures

Introduction générale

Dans un monde où la communication est essentielle à toute interaction humaine, la langue reste un outil fondamental d'expression. Pour les personnes sourdes ou malentendantes, la langue des signes est vitale, leur permettant de participer activement à la société. Pourtant, sa méconnaissance généralisée limite encore fortement leur inclusion sociale, éducative et professionnelle.

À l'ère du numérique et de l'intelligence artificielle, il devient possible de concevoir des outils innovants pour surmonter cette barrière. L'analyse vidéo combinée à l'apprentissage profond offre aujourd'hui de nouvelles perspectives pour automatiser la reconnaissance des gestes et traduire la langue des signes en texte compréhensible.

Ce mémoire s'inscrit dans cette dynamique. Il vise à concevoir une architecture logicielle capable de reconnaître en temps réel les signes captés par une caméra et de les restituer sous forme de texte, en s'appuyant sur un pipeline vidéo et sur l'adaptation de l'architecture SPOTER. L'avatar 3D est abordé comme un objectif d'ingénierie d'interface et d'architecture système : il sert à explorer la restitution visuelle des glosses et à préparer une expérience utilisateur plus complète, sans constituer l'hypothèse centrale de recherche.

0.1 Généralités sur le thème

La langue des signes est le principal outil de communication pour les personnes sourdes ou malentendantes. Cependant, son usage reste peu compris par la majorité, ce qui limite fortement l'inclusion sociale et professionnelle de cette communauté. Dans un monde de plus en plus connecté, l'intelligence artificielle et la vision par ordinateur ouvrent de nouvelles perspectives pour briser ces barrières. Grâce à des outils comme MediaPipe, OpenCV et l'apprentissage profond, il devient possible de concevoir des systèmes capables de traduire automatiquement la langue des signes en texte, en temps réel.

Ce mémoire vise à concevoir un prototype fonctionnel de reconnaissance automatisée

de la langue des signes, intégré à une interface de restitution textuelle et visuelle. L'animation d'un avatar 3D est envisagée comme une composante UI/UX complémentaire destinée à améliorer l'accessibilité et la lisibilité du système.

0.2 Identification et formulation du problème

L'absence de dispositifs de traduction automatique fiables et accessibles pour la langue des signes continue de constituer un obstacle majeur à la communication pour les personnes sourdes ou malentendantes. Dans une société où l'information circule principalement à l'oral ou à l'écrit, cette barrière de communication génère des répercussions considérables :

— **Sur le plan social**, elle alimente l'exclusion, limite l'accès à l'éducation, complique les démarches administratives et freine la participation citoyenne équitable.

— **Sur le plan économique**, elle entrave l'insertion professionnelle, réduit l'autonomie des personnes concernées et prive la société d'un potentiel humain important.

— **Sur le plan technologique**, bien que les progrès récents en intelligence artificielle et en traitement vidéo aient permis l'émergence de projets inclusifs, ceux-ci restent encore souvent peu accessibles ou mal adaptés aux contextes locaux. Il subsiste donc un besoin réel de solutions robustes, déployables en temps réel, et capables de combler efficacement ce fossé communicatif.

Il persiste une lacune dans la disponibilité de systèmes pratiques et performants capables de reconnaître, en temps réel, la langue des signes et d'en produire une restitution textuelle exploitable, adaptée surtout au contexte local. Cette carence accentue les inégalités sociales, limite les perspectives professionnelles des personnes malentendantes et souligne l'urgence de fournir davantage d'efforts pour concevoir des systèmes intelligents fondés sur l'analyse vidéo, l'apprentissage profond et des interfaces de restitution accessibles.

0.3 Questions de recherche

Face au besoin croissant d'outils technologiques pour faciliter la communication avec les personnes sourdes ou malentendantes, ce travail soulève les questions suivantes :

1. Comment concevoir un système capable de traduire, en temps réel, la langue des signes en langage écrit clair et compréhensible ?
2. Quels modèles d'apprentissage profond et techniques d'analyse vidéo sont les plus adaptés à la reconnaissance fiable de gestes issus de la langue des signes ?
3. Comment garantir la précision, la rapidité et la robustesse de ce système dans un environnement réel, avec des variations de conditions visuelles (éclairage, arrière-plan, vitesse d'exécution des gestes, etc.) ?
4. Comment structurer un moteur de rendu hybride pour restituer visuellement les glosses de manière fluide à travers un avatar 3D ?

0.4 Formulation des hypothèses

Ce mémoire repose sur les hypothèses suivantes :

1. L'intégration de MediaPipe, d'un pipeline de traitement vidéo asynchrone et de l'architecture SPOTER pourrait permettre la mise en place d'un système fiable de reconnaissance en temps réel de la langue des signes.
2. Les Transformers, en particulier SPOTER, devraient mieux capturer les dépendances temporelles des gestes que les modèles récurrents de type BiLSTM, grâce à leurs mécanismes d'attention temporelle.
3. L'extraction de points clés via MediaPipe pourrait permettre d'isoler le geste du bruit visuel (éclairage, arrière-plan), garantissant ainsi la robustesse du système en conditions réelles.

Ces hypothèses orienteront la conception du pipeline de traitement, la mise en œuvre de l'architecture de reconnaissance et l'évaluation du système informatique visant à atténuer la barrière de communication. Elles seront soumises à des tests afin de valider leur pertinence et leur applicabilité dans le contexte spécifique de la reconnaissance de la langue des signes.

Au-delà de ces hypothèses de recherche, le travail poursuit également un objectif secondaire d'implémentation : structurer une interface de restitution visuelle par avatar 3D, capable d'exploiter les glosses produits par le système afin d'améliorer l'expérience utilisateur et la compréhension du résultat.

0.5 Justification du choix du sujet et motivations

0.5.1 Motivation et intérêt pour le sujet

Le choix de ce sujet découle à la fois d'une forte sensibilité sociale et d'un intérêt pour l'innovation technologique. L'inclusion des personnes sourdes ou malentendantes demeure un défi majeur dans de nombreuses sociétés, et les outils de communication qui pourraient améliorer leur interaction avec le monde restent encore insuffisants. Allier les compétences en programmation et en apprentissage automatique à un projet à dimension sociale constitue une démarche pertinente dans le domaine de l'informatique. L'observation du milieu environnant met en évidence le potentiel d'un tel projet à avoir un impact réel sur la vie des individus et sur la société dans son ensemble. La traduction en temps réel de la langue des signes, rendue possible grâce à des techniques avancées telles que la vision par ordinateur et l'apprentissage profond, apparaît comme une solution prometteuse à ce défi.

0.5.2 Pertinence scientifique du sujet

Le sujet s'inscrit dans un champ de recherche très dynamique et actuel : l'application de l'intelligence artificielle (IA) et des technologies vidéo à des problématiques sociales complexes. La traduction de la langue des signes par des systèmes informatiques est un domaine encore en pleine évolution, avec de nombreux défis à relever en termes de reconnaissance des gestes, d'interprétation contextuelle, et de réactivité en temps réel. Ce projet se distingue par son intégration de l'apprentissage profond, de MediaPipe et d'OpenCV, des technologies largement utilisées dans d'autres domaines (comme la reconnaissance faciale et la vision par ordinateur), mais encore peu appliquées à la traduction de la langue des signes. L'originalité de ce travail réside dans la méthodologie et l'approfondissement de l'implémentation de ces technologies pour aboutir à un système précis et fiable, répondant à un besoin social pressant.

0.5.3 Pertinence sociale du sujet

La portée sociale de ce sujet est évidente : il vise à favoriser l'inclusion des personnes sourdes ou malentendantes en facilitant leur communication avec les autres. En effet, l'isolement social et les barrières de communication restent des obstacles majeurs pour cette population, que ce soit dans les contextes scolaires, professionnels ou même dans

les interactions quotidiennes. En facilitant l'accès à la langue des signes pour ceux qui ne la connaissent pas, ce système peut contribuer à améliorer leur qualité de vie, leur accès à l'éducation et à l'emploi, et réduire les discriminations. Ainsi, ce projet ne se limite pas à une question technique, il revêt une importance capitale pour le bien-être social de nombreux individus.

0.6 Énoncé des objectifs de recherche

0.6.1 L'objectif général

L'objectif principal de cette recherche est de concevoir et de développer un système informatique capable de traduire en temps réel la langue des signes en texte, en utilisant des techniques d'analyse vidéo et d'apprentissage profond. Ce système devra non seulement permettre la traduction fluide de la langue des signes en texte, mais également s'adapter aux variations de gestes dans des environnements variés, tout en offrant une interface accessible et intuitive pour les utilisateurs. Par la suite, la mise en œuvre de ce système pourrait devenir un outil clé pour l'inclusion sociale des personnes malentendantes ou sourdes.

0.6.2 Les objectifs opérationnels/spécifiques

Pour atteindre cet objectif général, les objectifs spécifiques sont les suivants :

1. **Analyser les technologies existantes :** Étudier les approches actuelles en matière de traduction automatique de la langue des signes et identifier leurs forces, limites et opportunités d'amélioration.
2. **Implémenter, adapter et entraîner l'architecture SPOTER :** Utiliser l'apprentissage profond pour reconnaître les gestes de la langue des signes, en intégrant MediaPipe pour l'extraction des landmarks et la capture de mouvements en temps réel.
3. **Exploitation d'un dataset open-source :** Sélectionner et prétraiter une base de données vidéo de signes annotés (ASL) afin d'entraîner et de valider les performances du modèle.
4. **Implémenter un pipeline de traduction en temps réel :** Mettre en place un système complet qui prend en entrée un flux vidéo et le traduit en texte, en

optimisant la précision et la réactivité du système.

5. **Valider les performances du système :** Tester le système sur différents ensembles de données, mesurer ses performances en termes de précision, de vitesse et de robustesse, et l'ajuster en fonction des résultats obtenus.
6. **Explorer la restitution visuelle par avatar 3D :** Étudier et développer un moteur de rendu hybride permettant de restituer les glosses sous forme d'animations, en tant que composante d'interface et d'architecture système.

0.7 Méthodologie et délimitation du travail

Ce travail adopte une démarche expérimentale et de recherche-développement (R&D) articulée en plusieurs étapes. Il débute par une revue de littérature permettant d'identifier les approches existantes liées à la reconnaissance gestuelle, à l'apprentissage profond et à l'analyse vidéo. Sur le plan méthodologique, un jeu de données existant en langue des signes américaine (ASL), déjà annoté et prétraité, est utilisé comme benchmark de validation méthodologique afin d'implémenter, d'adapter et d'entraîner l'architecture SPOTER dans les délais impartis pour ce cycle académique. La conception porte principalement sur le pipeline de traitement de données asynchrone au backend, l'orchestration des flux vidéo, l'extraction des landmarks et l'architecture globale du système. Une phase d'évaluation permettra de tester la précision, la rapidité et la robustesse du système. En parallèle, une composante exploratoire d'interface sera envisagée pour restituer visuellement les glosses à travers un avatar 3D. Le projet se limite à la reconnaissance de vidéos ASL, sans prise en charge expérimentale d'autres langues de signes. L'objectif est de produire un prototype fonctionnel et démonstratif en réponse à une problématique sociale réelle.

0.8 Subdivision du travail

Ce mémoire est structuré en trois chapitres principaux, en plus de l'introduction générale et de la conclusion générale.

1. Le **premier chapitre "Revue de la littérature"**. Il présente l'état de l'art sur la traduction de la langue des signes, les techniques d'analyse vidéo et les principes de l'apprentissage profond dans ce domaine. Il vise à établir les fondements théoriques nécessaires au développement du système proposé.

2. Le **deuxième chapitre "Méthodologie et conception du système"** aborde la méthodologie adoptée ainsi que la conception du système. Il détaille les besoins techniques, le processus de collecte et de préparation des données, le choix des modèles, ainsi que l'architecture générale du système.
3. Le **troisième chapitre "Résultats et Analyse Critique"** présente les résultats obtenus. Il décrit les étapes de mise en œuvre, les performances du système à travers différents tests, ainsi qu'une analyse critique des résultats et des pistes d'amélioration.

Chapitre 1

Revue de la littérature

1.1 Introduction partielle

L'essor des technologies d'intelligence artificielle et de vision par ordinateur a ouvert de nouvelles perspectives dans la communication entre personnes entendantes et personnes sourdes. La langue des signes, en tant que système linguistique complet, reste cependant difficile à interpréter automatiquement en raison de sa complexité visuelle et grammaticale. Ce chapitre propose une revue de la littérature sur les travaux existants liés à la reconnaissance et à la traduction de la langue des signes à l'aide de techniques d'apprentissage profond.

L'objectif est de situer le présent travail par rapport à l'état de l'art, de mettre en lumière les principales approches, les jeux de données disponibles, ainsi que les limites actuelles qui justifient la conception d'un système innovant fondé sur MediaPipe et des architectures de Deep Learning.

1.2 Contexte général

1.2.1 Importance de la langue des signes

La langue des signes est un outil de communication essentiel pour la communauté sourde et malentendante. Contrairement à une idée répandue, elle ne se limite pas à une série de gestes mais constitue un langage complet, possédant une grammaire et une syntaxe propres [1]. On estime à plus de 70 millions le nombre de personnes utilisant une langue des signes dans le monde, et environ 80% d'entre elles vivent dans des pays en développement [2].

Chacune de ces communautés dispose de sa propre langue des signes, telle que l'American Sign Language (ASL), la Langue des Signes Française (LSF) ou la Langue des Signes Congolaise (LSC), entre autres [3]. Bien que ces langues partagent des structures visuo-gestuelles communes, elles divergent radicalement par leur lexique et leur syntaxe, rendant chaque système unique. Dans le cadre de ce travail, nous nous focaliserons spécifiquement sur l'American Sign Language.

Ces langues constituent un vecteur d'identité culturelle et sociale fort pour leurs locuteurs. Cependant, leur reconnaissance officielle varie d'un pays à l'autre. Dans de nombreux contextes, notamment africains, la langue des signes reste peu documentée, non normalisée et insuffisamment intégrée dans les politiques éducatives et technologiques [4].

1.2.2 Enjeux de l'inclusion et de l'accessibilité

L'absence d'outils technologiques capables d'assurer la communication entre personnes sourdes et entendantes engendre une barrière majeure à l'inclusion sociale, éducative et professionnelle [5].

Les interactions quotidiennes (accès aux services publics, soins de santé, éducation, emploi) demeurent limitées pour la majorité des personnes sourdes.

Selon le World Report on Hearing publié par l'Organisation mondiale de la Santé (OMS, 2021), plus de 1,5 milliard de personnes dans le monde présentent une perte auditive, dont environ 430 millions ont besoin de services de rééducation [6]. Ce chiffre ne cesse de croître, soulignant l'urgence de développer des solutions innovantes favorisant la communication universelle.

L'essor de l'intelligence artificielle et de la vision par ordinateur offre une opportunité unique pour répondre à ces défis. Les systèmes de traduction automatique de la langue des signes en texte ou en parole représentent un pas important vers une société plus inclusive, où les barrières linguistiques sont atténuées par la technologie [7] [8].

1.2.3 Initiatives et efforts institutionnels

Plusieurs institutions internationales œuvrent pour la reconnaissance et la normalisation des langues des signes.

La Convention des Nations Unies relative aux droits des personnes handicapées (CRPD, 2006) stipule, dans son article 21, la nécessité de reconnaître et de promouvoir l'usage de la langue des signes dans tous les domaines de la vie publique [9].

De plus, la Journée internationale des langues des signes, célébrée chaque 23 septembre, rappelle l'importance de cette reconnaissance [10].

Au niveau local, des associations et organisations comme Handicap International ou encore des centres linguistiques spécialisés entreprennent des actions de sensibilisation et de formation [11]. Toutefois, la documentation numérique et les corpus visuels restent encore insuffisants, freinant les avancées scientifiques et technologiques dans la région [12].

1.2.4 Vers une approche technologique inclusive

Dans ce contexte, la recherche sur la traduction automatique de la langue des signes vise à réduire les inégalités communicationnelles. L'idée est de combiner la vision par ordinateur (MediaPipe, OpenCV,...) et l'apprentissage profond (Deep Learning) pour analyser les gestes, les expressions faciales et les postures, puis traduire ces informations en texte [13] [14].

Ce type de système peut être utilisé dans divers environnements : éducation, santé, services publics ou encore communication en ligne. L'objectif final est de favoriser l'intégration sociale des personnes sourdes tout en valorisant la richesse linguistique de la langue des signes locale [15].

1.3 Analyse structurelle de la langue des signes : les paramètres chérémiques

Pour concevoir un système de reconnaissance efficace, il est impératif de comprendre la structure phonologique de la langue cible. Contrairement aux langues orales basées sur des phonèmes sonores, la langue des signes (LSF, ASL, LSC) repose sur des unités visuelles appelées "chérèmes" ou paramètres de Stokoe [16]. La reconnaissance automatique (SLR) doit être capable de capturer et de discriminer simultanément cinq paramètres fondamentaux :

1. **La Configuration (Handshape) :** Elle désigne la forme prise par la main (doigts ouverts, fermés, pincés, etc.). Il existe plus de 50 configurations standard en ASL. Pour notre système, cela implique que le module MediaPipe doit extraire avec une précision millimétrique la position relative des phalanges pour distinguer, par exemple, la lettre "A" (poing ferme, pouce sur le côté) de la lettre "S" (poing ferme, pouce devant).

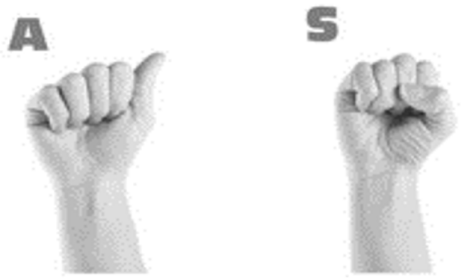

FIGURE 1.1 – Signe A et S

2. **L'Emplacement (Location) :** Il s'agit de la zone du corps où le signe est exécuté (devant le torse, sur le front, près de la bouche). Cette dimension spatiale justifie l'utilisation de la composante "Pose" de MediaPipe en plus de la composante "Hands", afin de situer la main par rapport au repère corporel.

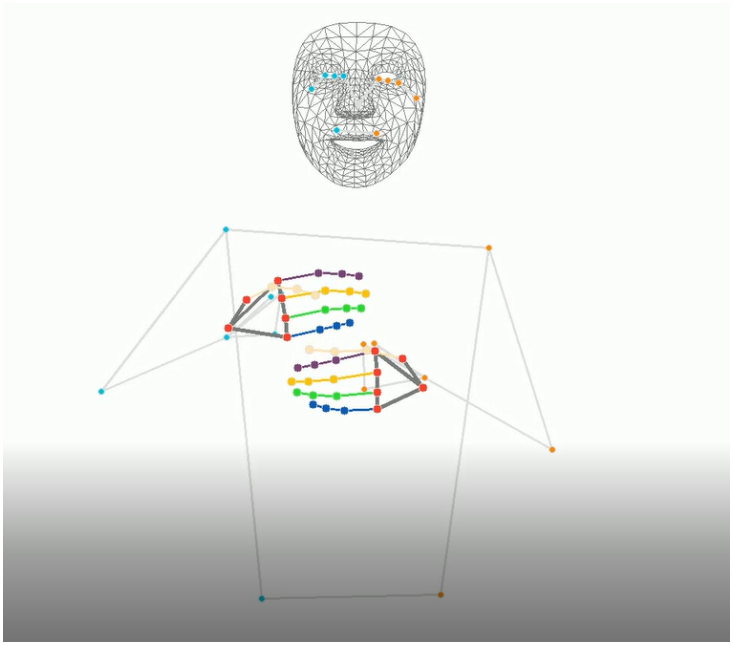

FIGURE 1.2 – Illustration complète des landmarks

3. **Le Mouvement (Movement) :** Il décrit l'action cinétique (droite, circulaire, saccadée). C'est ici que l'architecture temporelle du réseau de neurones (Transformer ou LSTM) joue son rôle crucial, car une image statique ne peut capturer cette dynamique.
4. **L'Orientation (Orientation) :** C'est la direction de la paume de la main (vers le haut, le bas, l'interlocuteur). Une erreur d'orientation peut inverser le sens d'un signe (ex. : "Moi" vs "Toi").

5. **Les Composantes Non-Manuelles (Non-Manual Features - NMF) :** Souvent négligées, elles portent la grammaire (interrogation, négation). Bien que notre prototype se concentre sur les mains, l'intégration des landmarks du visage (Face Mesh) dans notre vecteur de caractéristiques prépare le terrain pour la détection de ces subtilités linguistiques.

1.4 Travaux sur la reconnaissance de la langue des signes

1.4.1 Évolution historique de la recherche

Les premiers travaux de reconnaissance automatique de la langue des signes (RALSi) remontent aux années 1990. Ces recherches utilisaient principalement des gants de données (data gloves) pour capturer les mouvements des doigts et des mains [17].

Bien que ces approches aient permis d'obtenir des résultats encourageants, elles présentaient des limites importantes, notamment en termes de coût, de portabilité, et d'acceptabilité sociale [18].

L'avènement de la vision par ordinateur a marqué un tournant décisif. Des travaux tels que ceux de Starner et Pentland (1997) ont proposé une reconnaissance des signes à partir de vidéos en temps réel, en utilisant des méthodes de suivi de mouvement et de modélisation statistique [19]. Historiquement, la probabilité d'observer une séquence de gestes sachant un modèle était calculée ainsi :

$$P(O|\lambda) = \sum_{Q} P(O|Q,\lambda)P(Q|\lambda) \tag{1.1}$$

Ces approches ont ouvert la voie à des systèmes plus naturels, capables d'interpréter la langue des signes sans équipement spécialisé.

1.4.2 Reconnaissance basée sur la vision par ordinateur

L'utilisation d'images issues de caméras RGB et de capteurs de profondeur (Kinect, Leap Motion) a permis d'importantes avancées dans la capture des gestes [20].

Des modèles classiques tels que les réseaux de neurones convolutifs (CNN), les modèles de Markov cachés (HMM) et les réseaux de neurones récurrents (RNN) ont été largement utilisés pour reconnaître des signes à partir d'images ou de séquences vidéo [21] [22].

Par exemple, Pigou et al. (2015) ont utilisé un CNN pour la reconnaissance de signes

isolés à partir de vidéos, atteignant des taux de précision supérieurs à 90 % sur des bases de données limitées [23].

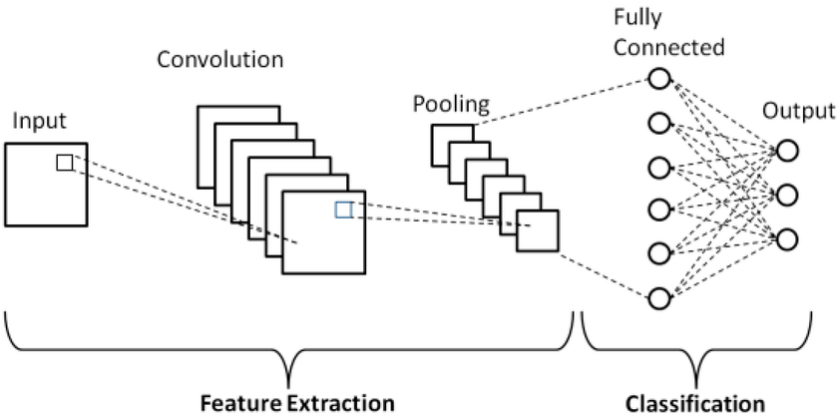

FIGURE 1.3 – Schéma d'architecture de réseau neuronal CNN

L'opération fondamentale d'extraction de caractéristiques dans un CNN s'exprime par le produit de convolution :

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n) \cdot K(m,n) \tag{1.2}$$

De leur côté, Camgoz et al. (2018) ont introduit un système de traduction neuronale complet capable de convertir des séquences de gestes en texte, intégrant une approche de type encoder-decoder similaire à celles utilisées en traitement automatique du langage naturel (NLP) [24].

1.4.3 Intégration de l'apprentissage profond

Avec l'essor du Deep Learning, les systèmes modernes de reconnaissance de la langue des signes exploitent des architectures hybrides combinant CNN, LSTM et Transformers pour modéliser à la fois l'espace et le temps [25] [26].

Les modèles CNN-LSTM permettent de reconnaître des séquences de gestes continus en analysant simultanément les caractéristiques spatiales (formes des mains) et temporelles (mouvements successifs) [27].

Des réseaux plus récents tels que I3D (Inflated 3D ConvNet) ou Vision Transformers (ViT) offrent une meilleure compréhension contextuelle des gestes, notamment dans les phrases continues de langue des signes [28].

L'objectif est de passer de la simple reconnaissance de mots isolés à une traduction fluide et sémantiquement cohérente, ce qui constitue aujourd'hui l'un des défis majeurs

du domaine.

1.4.4 Utilisation de landmarks et de MediaPipe

Afin d'alléger la charge computationnelle et d'améliorer la robustesse face aux variations d'éclairage, certaines approches récentes utilisent les landmarks extraits via MediaPipe (Google) ou d'autres outils de détection de points clés [29]. Ces landmarks représentent les coordonnées des articulations de la main, du visage et du corps, permettant d'entraîner des modèles plus légers et plus rapides. Par exemple, Marczak et al. (2023) ont utilisé MediaPipe Hands et Holistic pour réduire la taille des données d'entraînement de 80 % tout en maintenant une précision supérieure à 92 % [30]. Cette approche ouvre la voie à des systèmes de traduction temps réel fonctionnant sur des dispositifs embarqués ou mobiles [31].

1.4.5 Limites et perspectives des travaux existants

Malgré ces avancées, plusieurs défis persistent.

Premièrement, la rareté de jeux de données de grande qualité, surtout pour les langues des signes locales (comme la LSC), freine l'entraînement de modèles généralisables [32].

Dans ce contexte, l'utilisation d'un corpus ASL ne constitue pas un renoncement à l'objectif d'adaptation locale, mais un choix méthodologique de validation. L'ASL, mieux documentée et plus largement disponible sous forme de corpus annotés, offre un benchmark de validation méthodologique permettant de tester le pipeline MediaPipe–SPOTER, d'évaluer les choix architecturaux et de mesurer les performances du système avant une extension future vers des langues des signes locales telles que la LSC.

Deuxièmement, les modèles actuels peinent à gérer les variations individuelles des gestes, la coarticulation entre signes, et les expressions faciales essentielles à la sémantique [33].

Enfin, l'intégration complète d'un système de reconnaissance, traduction et visualisation (par avatar 3D ou texte) demeure encore un domaine de recherche actif.

Des projets récents comme RWTH-PHOENIX-2014T ou Sign2Text Transformer montrent qu'il est désormais possible d'atteindre une traduction fluide de séquences continues, mais souvent au prix d'une forte dépendance à des corpus massifs et coûteux [34] [35].

1.5 Traduction automatique des signes en texte

La traduction automatique des signes en texte vise à transformer les glosses reconnus en une langue naturelle fluide, ce qui dépasse la simple reconnaissance de gestes. Les premières tentatives utilisaient des architectures seq2seq avec RNN, LSTM/GRU pour traiter les séquences temporelles [36] [37]. Cependant, ces modèles souffraient de pertes de contexte sur les phrases longues et d'une difficulté à généraliser à des locuteurs différents.

La révolution est survenue avec le Transformer introduit par Vaswani et al. (2017) [38], basé sur le mécanisme d'attention. Ce modèle a permis de capturer efficacement les dépendances à long terme dans les séquences de gestes, améliorant ainsi la qualité de la traduction. Le cœur du transformer repose sur le mécanisme d'attention, permettant de pondérer l'importance des différents mots dans une phrase :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.3}$$

Nous y reviendrons dans la suite. Camgoz et al. (2018) ont présenté Sign2Text, le premier modèle capable de traduire automatiquement des séquences vidéo en texte écrit [24]. Cette approche a été améliorée en 2020 par le Sign Language Transformer, combinant reconnaissance de signes et traduction en un modèle unifié [34].

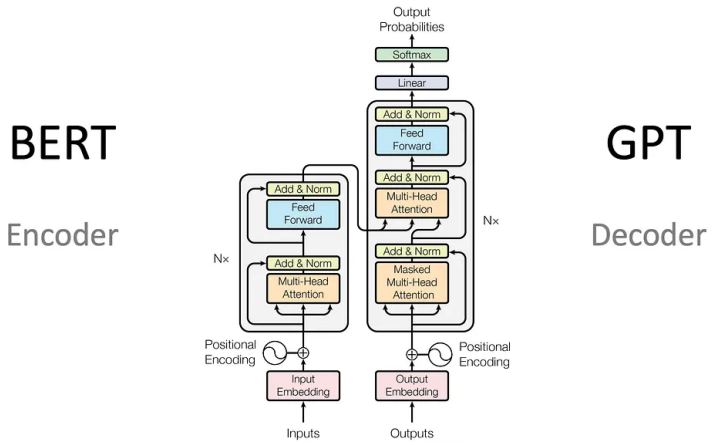

FIGURE 1.4 – Architecture Transformer (www.medium.com)

Yin, Read et Cohn (2021) ont proposé le STMC-Transformer, optimisé pour mieux exploiter les indices spatio-temporels dans les séquences vidéo, tandis que Zhou et al. (2021) ont développé le Spatial-Temporal Multi-Cue Network, qui intègre simultanément les informations de mouvement et d'apparence pour produire une traduction plus cohérente

[39] [40].

Enfin, l'utilisation de modèles de langage pré-entraînés comme BERT (Devlin et al., 2019) a permis d'améliorer la fluidité linguistique des traductions glosse → texte naturel, rapprochant les résultats de la qualité humaine [41].

1.6 Extraction de caractéristiques biomécaniques (MediaPipe)

Contrairement aux approches classiques qui traitent l'image entière (pixels), l'approche moderne privilégiée dans ce travail repose sur la squelettisation. L'outil MediaPipe de Google représente l'état de l'art dans ce domaine. Il ne voit pas la main comme une image, mais comme un graphe de nœuds et d'arêtes.

1.6.1 Architecture du pipeline MediaPipe

Le fonctionnement repose sur deux modèles en cascade pour optimiser la vitesse :

— **Détecteur de paume (Palm Detector) :** il identifie la région d'intérêt (ROI) de la main dans l'image globale.

— **Modèle de repères (Landmark Model) :** il opère sur la région recadrée pour prédire les coordonnées 3D de 21 points clés (jointures).

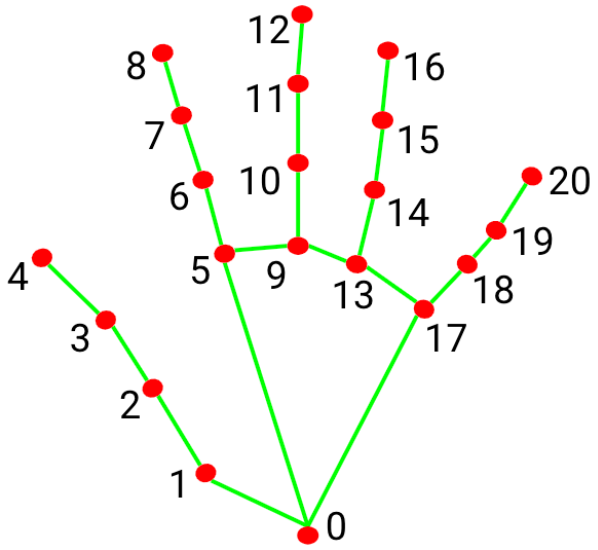

FIGURE 1.5 – Topologie des 21 points clés extraits par MediaPipe Hands (www.researchgate.com)

La précision des positions des points clés est optimisée par une fonction de perte (loss function) basée sur la régression, minimisant l'écart entre la position prédite et la position réelle :

$$\mathcal{L}_{reg} = \sum_{i=1}^{N} \| P_{pred}^{(i)} - P_{real}^{(i)} \|^2 \tag{1.4}$$

Cette approche réduit considérablement la dimensionnalité des données : au lieu de traiter des millions de pixels (ex. : 1920x1080), le réseau de neurones ne reçoit qu'un vecteur de coordonnées (x_i, y_i, z_i) pour chaque point, rendant le calcul en temps réel possible sur des processeurs standards (CPU).

1.7 Normalisation et prétraitement des séquences

Au niveau du prétraitement des séquences, nous savons que les données brutes, dans notre cas issues de la caméra, ne peuvent pas être injectées directement dans un modèle de Deep Learning. Elles doivent généralement subir un prétraitement pour garantir l'invariance.

1.7.1 Normalisation spatiale

Les coordonnées des points varient selon la distance de l'utilisateur face à la caméra. Pour rendre le modèle robuste à ces variations d'échelle, on applique une normalisation Min-Max pour ramener toutes les valeurs dans l'intervalle [0,1] :

$$x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}, y'_i = \frac{y_i - y_{min}}{y_{max} - y_{min}} \tag{1.5}$$

Où x_{min} et x_{max} sont les valeurs minimales et maximales observées dans la trame courante.

1.7.2 Gestion temporelle (Padding)

La langue des signes est temporelle. La durée d'un signe varie (certains durent 2 secondes, d'autres 0,5 seconde). Pour des réseaux de neurones nécessitant des entrées de taille fixe, on applique une technique de remplissage (padding) ou de troncature pour uniformiser la longueur des séquences vidéo (T).

$$S_{input} = \{\text{frame}_1, \text{frame}_2, \ldots, \text{frame}_T, 0, \ldots, 0\} \tag{1.6}$$

Cette étape garantit que le tenseur d'entrée respecte la structure matricielle attendue par le modèle LSTM ou Transformer.

1.8 Métriques d'évaluation de la performance

Pour valider l'efficacité des modèles de traduction, la littérature scientifique s'appuie sur des métriques standardisées.

1.8.1 Précision (Accuracy)

Pour la classification de signes isolés (reconnaître "Hello" vs "Thank"), on utilise la précision globale :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.7}$$

Où : **TP (vrais positifs)** représente le nombre de signes correctement identifiés.

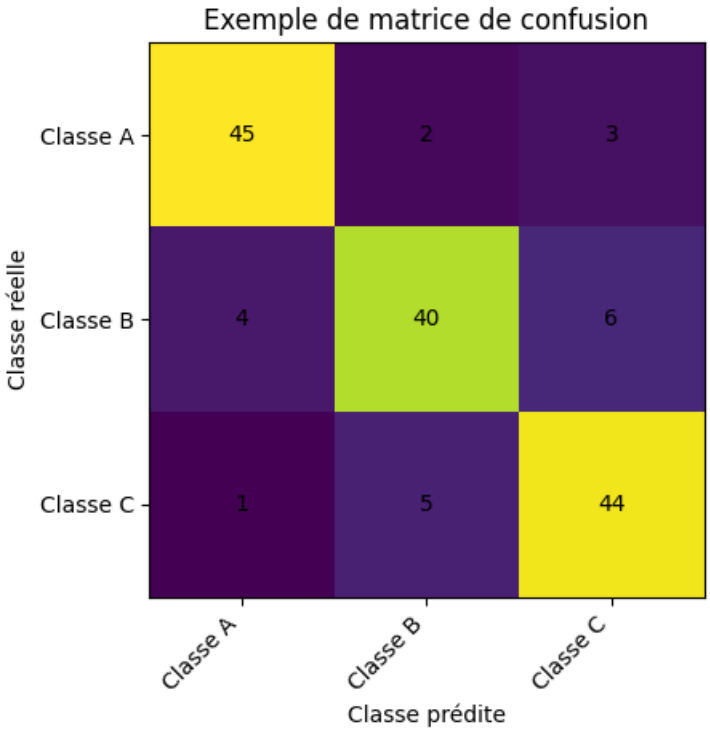

FIGURE 1.6 – Exemple de matrice de confusion

1.8.2 Top-K Accuracy

La Top-K Accuracy est une extension de la précision classique (Top-1). Elle est particulièrement robuste pour évaluer des modèles de reconnaissance gestuelle où la frontière entre deux signes peut être mince. Elle correspond à la proportion d'échantillons où la classe réelle se trouve parmi les classes ayant obtenu les scores de probabilité les plus élevés en sortie du modèle :

$$\text{Top-K Accuracy} = \frac{1}{N}\sum_{i=1}^{N} 1\left[y_i \in \text{rank}_k\left(\hat{P}_i\right)\right] \tag{1.8}$$

Où :

— N : Le nombre total d'échantillons dans le jeu de test.

— y_i : La vérité de terrain (ground truth), soit l'étiquette réelle du signe effectué pour l'échantillon i.

— $\hat{P}_i$: Le vecteur de probabilités prédit par le modèle pour l'échantillon i.

— $\text{rank}_k(\hat{P}_i)$: L'ensemble des indices des K plus grandes valeurs du vecteur de probabilité.

— $\mathbb{1}[\cdot]$: La fonction indicatrice, qui retourne 1 si la condition est vraie (le signe réel est dans le top K) et 0 sinon.

1.8.3 Taux d'Erreur sur les Mots (WER)

Pour la traduction de phrases continues, la métrique de référence est le Word Error Rate (WER), issue de la reconnaissance vocale. Elle mesure la distance d'édition de Levenshtein entre la phrase prédite et la phrase de référence :

$$\text{WER} = \frac{S + D + I}{N} \tag{1.9}$$

Où :

— S est le nombre de substitutions (mot incorrect remplacé par un autre),

— D est le nombre de suppressions (mot oublié),

— I est le nombre d'insertions (mot ajouté à tort),

— N est le nombre total de mots dans la phrase de référence. Un WER plus faible indique une meilleure performance du système.

1.9 Synthèse critique

Les recherches actuelles montrent des progrès significatifs dans la reconnaissance et la traduction de la langue des signes.

Les architectures modernes (CNN, LSTM, Transformers) ont permis de traiter à la fois les signes isolés et les séquences continues, et d'améliorer la qualité des traductions [22] [24] [34]

Nous pouvons ainsi dégager plusieurs constats majeurs :

1. **Évolution vers la légèreté :** on observe un abandon progressif des capteurs physiques (gants) et des traitements d'images lourds (pixels bruts) au profit de la squelettisation (MediaPipe). Cette approche offre le meilleur compromis entre précision biomécanique et rapidité d'exécution.
2. **Importance du contexte temporel :** La simple reconnaissance d'images (CNN) est insuffisante. L'intégration de la dimension temporelle via des réseaux récurrents (LSTM) ou des mécanismes d'attention (Transformers) est indispensable pour capturer la syntaxe de la langue des signes.
3. **Le défi de la normalisation :** La robustesse d'un système dépend moins de la complexité du réseau de neurones que de la qualité du prétraitement des données (normalisation spatiale et temporelle) pour gérer la variabilité des utilisateurs.

Ces constats valident notre choix méthodologique pour ce mémoire : combiner la rapidité d'extraction de MediaPipe avec la capacité séquentielle des réseaux LSTM/GRU et Transformers, afin de proposer une solution fluide, capable de fonctionner en temps réel sur des ordinateurs standards, répondant ainsi aux contraintes locales d'accessibilité. Cependant, plusieurs limites persistent :

1. **Disponibilité des données :** Peu de corpus de qualité existent pour les langues locales comme la Langue des Signes Congolaise (LSC), limitant la généralisation des modèles [32].
2. **Variabilité des gestes :** Les différences entre locuteurs, la coarticulation et l'expression faciale compliquent la reconnaissance continue [33].
3. **Contraintes computationnelles :** Les modèles lourds sont difficiles à déployer en temps réel sur des dispositifs mobiles ou embarqués [31].

4. **Complexité linguistique :** La traduction glosse → texte naturel doit gérer des différences syntaxiques et grammaticales importantes entre la langue des signes et la langue écrite [41].

Ainsi, nous soulignons la nécessité de développer des solutions légères, multimodales et adaptatives, capables de fonctionner en temps réel et de s'adapter à des langues des signes peu documentées, ce qui constitue la problématique centrale de ce mémoire.

Cette analyse justifie l'utilisation de l'ASL comme terrain expérimental initial : elle permet de valider rigoureusement la méthodologie sur un corpus accessible et suffisamment structuré, tout en préparant l'extension du pipeline vers la LSC lorsque des données locales annotées seront disponibles.

1.10 Conclusion partielle

Ce chapitre a présenté une revue plus ou moins complète de la littérature sur la reconnaissance et la traduction automatique de la langue des signes.

— Les premières méthodes utilisaient des gants de données ou des modèles statistiques.

— L'arrivée de la vision par ordinateur et du Deep Learning a permis de capturer les signes sans équipement spécialisé.

— L'intégration de CNN, LSTM et Transformers, ainsi que des landmarks via MediaPipe, a rendu possible la traduction temps réel vers le texte naturel.

Malgré ces avancées, des défis majeurs demeurent, notamment la rareté de données locales, la variabilité gestuelle, et les contraintes de calcul.

Ces observations justifient le choix méthodologique de ce mémoire : un système combinant MediaPipe pour l'extraction des landmarks, l'implémentation et l'entraînement de l'architecture SPOTER pour la reconnaissance des gestes, et une interface temps réel destinée à l'ASL (American Sign Language) comme benchmark de validation. Ainsi, avec cette base théorique, nous pouvons désormais aborder le chapitre suivant, consacré à la Méthodologie et conception architecturale du système.

Chapitre 2

Méthodologie et conception du système

2.1 Introduction partielle

La phase de conception constitue le pivot central de la réalisation de notre système de reconnaissance de la langue des signes (SLR - Sign Language Recognition). Ce chapitre détaille la méthodologie adoptée pour transformer des données visuelles brutes en une restitution textuelle cohérente et exploitable en temps réel. Nous aborderons successivement la définition des besoins complexes d'un tel système, l'exploitation du corpus WLASL [42] via le dataset pré-calculé MuteMotion [43](stratégie d'optimisation des ressources), et enfin l'architecture logicielle et mathématique innovante reposant sur des Transformers d'Attention (SPOTER) [44].

L'objectif est de proposer une solution non seulement performante sur le plan de la précision, mais aussi viable dans un contexte d'utilisation réelle, caractérisé par des contraintes de latence strictes et une grande variabilité des entrées utilisateur.

2.2 Définition des besoins techniques et fonctionnels

2.2.1 Analyse des besoins fonctionnels

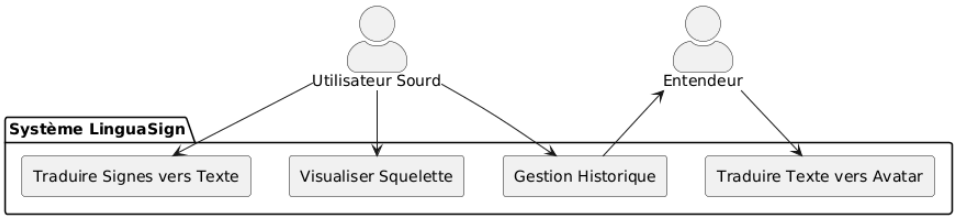

FIGURE 2.1 – Diagramme de CU global du Système

Cette figure 2.1 illustre les interactions entre les acteurs Utilisateur sourd et Entendeur avec le système. Le système est principalement orienté vers la reconnaissance de la langue des signes et sa restitution textuelle, tout en intégrant une composante de visualisation des glosses via un avatar 3D. Les besoins fonctionnels définissent les services et les fonctionnalités que le système doit offrir pour répondre aux attentes des utilisateurs (sourds et entendants). Ils constituent le socle sur lequel repose la modélisation des cas d'utilisation. Les principales fonctionnalités retenues pour ce système sont les suivantes :

- **Reconnaissance et restitution en temps réel :** Le cœur du système réside dans sa capacité à assurer une communication fluide. Cela inclut :
  - La reconnaissance gestuelle : Le système doit être capable de capturer les mouvements de l'utilisateur via une interface vidéo, d'en extraire les points caractéristiques et de le traduire fidèlement en texte ou en glosses.
  - La restitution visuelle par avatar : en complément de la reconnaissance, le système doit offrir une interface capable d'afficher des glosses sous forme d'animations fluides exécutées par un avatar 3D.
- **Feedback visuel et contrôle de capture :** Pour garantir la précision de la reconnaissance, l'utilisateur doit disposer d'un retour immédiat sur ce que le système perçoit. Le système doit donc afficher une superposition du squelette numérique (landmarks) sur le flux vidéo en direct. Ce mécanisme permet à l'utilisateur d'ajuster sa position ou l'éclairage en cas de mauvaise détection.
- **Traitement et reformulation linguistique (NLP) :** Au-delà de la simple reconnaissance mot à mot, le système doit intégrer un service de reformulation. Puisque

la syntaxe de la langue des signes diffère de celle des langues parlées, le service doit tenter de structurer les séquences de glosses prédites pour générer des phrases en texte naturel cohérentes.

2.2.2 Analyse des besoins techniques (Performance et sécurité)

Le développement de ce système a été guide par des impératifs techniques stricts. Ces besoins non-fonctionnels servent de critères de succès pour l'architecture et guident le choix des technologies :

1. **Performance Temporelle (Latence) :** Afin d'assurer une interaction fluide et naturelle, une latence cible inférieure à 500 ms (Round-Trip Time) a été définie. Ce seuil est critique pour maintenir la synchronisation entre le geste produit et sa restitution textuelle, évitant ainsi toute rupture cognitive pour l'utilisateur.
2. **Efficacité Prédictive (Précision) :** Le moteur de reconnaissance est conçu pour optimiser la Top-3 Accuracy. Ce choix méthodologique permet de gérer l'ambiguïté inhérente à certains gestes en garantissant que le signe correct figure parmi les trois probabilités les plus élevées, assurant ainsi la robustesse sémantique du système.
3. **Éthique et Protection des Données (Confidentialite) :** Conformément aux principes de Privacy by Design, le système garantit l'absence de stockage permanent des flux vidéo. Le traitement des images s'effectué de manière éphémère en mémoire vive pour l'extraction des points clés (landmarks), assurant une totale confidentialité des données biométriques des utilisateurs.

2.3 Sélection du corpus, justification pragmatique et préparation des données

L'une des étapes critiques de la conception de ce système réside dans le choix d'un corpus d'entraînement capable de soutenir les exigences d'un modèle de type Transformer. La sélection s'est heurtée à une contrainte majeure liée au contexte local : l'absence totale de bases de données annotées et numérisées pour la Langue des Signes Congolaise (LSC)

2.3.1 Le défi de la langue des signes congolaisé (LSC)

Bien que la solution soit destinée à la République Démocratique du Congo, le développement d'un modèle d'apprentissage profond requiert des milliers de séquences vidéo structurées. La création *ex nihilo* d'un tel dataset exigerait des ressources humaines expertes et un cadre temporel excédant largement les limites de ce travail de recherche. Face à cette carence, une approche de transfert technologique a été privilégiée.

2.3.2 Choix des corpus WLASL/MuteMotion

Pour pallier l'absence de données locales, nous avons opté pour l'utilisation conjointe des datasets WLASL (Word-Level American Sign Language) [42] et MuteMotion. Ce choix repose sur une triple argumentation

1. **Exigence de volume et robustesse :** Avec plus de 20 000 vidéos couvrant 2 000 signes, ces corpus offrent la masse critique nécessaire pour entraîner un réseau de neurones complexe sans risquer un sur-apprentissage (*overfitting*) immédiat, garantissant ainsi une meilleure généralisation du modèle.
2. **Agnosticisme linguistique de l'architecture** : La chaîne de traitement développée (pipeline MediaPipe vers Transformer) est conçue pour être indépendante du vocabulaire spécifique. Une fois la preuve de concept validée sur ces données de référence, l'architecture pourra être ré-entraînée sur un corpus LSC dès sa constitution future.
3. **Universalité cinématique** : Bien que les signes diffèrent lexicalement, les structures spatiales, les points d'articulation et la dynamique des mouvements restent comparables d'une langue des signes à l'autre, permettant une validation technique pertinente de notre solution

Cependant, nous avons adopté une approche plus ou moins pragmatique pour la gestion des données, privilégiant l'efficacité computationnelle.

2.3.3 Stratégie d'optimisation des ressources (WLASL vs MuteMotion)

L'apprentissage profond sur vidéo nécessite traditionnellement deux phases coûteuses :

1. **Extraction de Caractéristiques (Backbone) :** Convertir les pixels (RGB) en points d'intérêt structures (Landmarks).
2. **Modélisation Temporelle :** Analyser la séquence de ces points.

Utiliser le dataset WLASL brut (plus de 2TB de vidéos) nous aurait contraints a dédier plusieurs centaines d'heures de calcul GPU uniquement pour l'étape 1. Pour contourner ce goulot d'etranglement, nous avons intégré le corpus MuteMotion. Qui n'est en aucun cas un dataset conçurrent, mais une version pré-calculée optimisée de WLASL. Il contient les résultats de l'extraction MediaPipe Holistic sur l'intégralité des vidéos WLASL, validés et nettoyés [42].

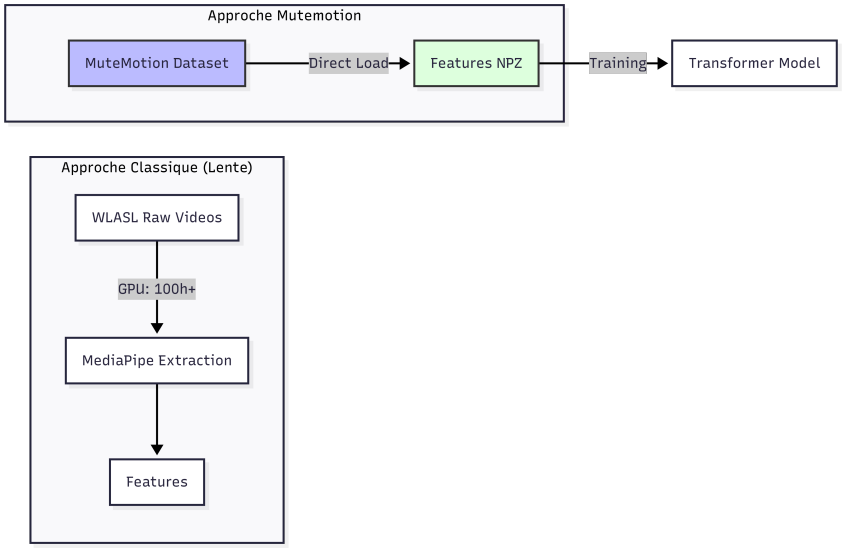

FIGURE 2.2 – Mutemotion vs WLASL brut

Ce diagramme met en évidence la réduction significative du temps de calcul apportée par l'approche MuteMotion, en évitant l'étape d'extraction des landmarks depuis les vidéos brutes, tout en conservant la compatibilité avec un modèle Transformer pour la reconnaissance des glosses. Cependant, nous conservons néanmoins un pipeline parallèle pour l'extraction des caractéristiques sur les vidéos brutes afin de permettre l'ajout futur de données personnelles ou locales

Cette stratégie nous permet de :

— Éliminer la redondance : Ne pas recalculr ce qui a déjà été standardisé par la communauté.

— Focaliser la Recherche : Allouer 100% du temps de calcul à l'architecture du Transformer (Model Tuninng) plutôt qu'au prétraitement

2.3.4 Filtrage des points clés

L'extraction des caractéristiques se concentre sur les zones les plus expressives de la langue des signes, à savoir les mains, le visage et la posture du haut du corps.

Dans cette optique, un filtrage des points clés est appliqué afin de ne conserver que les landmarks pertinents pour la reconnaissance des gestes, tout en éliminant les points redondants ou peu informatifs. Cette sélection permet de réduire la dimension des données d'entrée, d'améliorer l'efficacité computationnelle du modèle et de limiter le bruit introduit par des mouvements non significatifs.

Le système retient ainsi un sous-ensemble de 156 points clés, correspondant principalement aux articulations des mains et aux régions faciales impliquées dans l'expression linguistique, constituant une représentation compacte et discriminante des gestes observés. On peut formaliser le filtrage des 156 points clés comme une fonction de sélection :

$$\mathcal{L}_{sel} = \{l_i \in \mathcal{L}_{all} \mid i \in \mathcal{I}_{exp}\} \tag{2.1}$$

Où :

— $\mathcal{L}_{all}$ Représente l'ensemble des landmarks extraits par le modèle de détection(∼MediaPipe),

— $l_i = (x_i, y_i, z_i)$ Désigne i^e point cle en coordonnées tridimensionnelles normalisées,

— $\mathcal{I}_{exp}$ correspond à l'ensemble des indices des points clés sélectionnés et utilisés comme entrée du modèle de reconnaissance.

FIGURE 2.3 – Processus de filtrage de 156 points

Cette figure 2.3 illustre le déroulement du processus de filtrage applique aux données vidéo. Après la capture de la séquence d'entrée, les landmarks sont extraits à l'aide du modèle MediaPipe. Une sélection ciblee des zones expressives, en particulier les mains et le visage, est ensuite effectuée afin de réduire la complexité des données tout en conservant les informations pertinentes pour la reconnaissance des gestes. Cette étape aboutit à une représentation finale composée de 156 points clés, utilisée comme entrée du modèle d'apprentissage.

2.3.5 Augmentation et normalisation

Nous utilisons des transformations stochastiques pour améliorer la robustesse :

$$p'_i = \frac{p_i - P_center}{scale} \tag{2.2}$$

Cette équation décrit une normalisation spatiale des points (landmarks) afin de rendre les données invariantes à la position et à l'échelle du sujet dans l'image.

Où :

— p_i : Coordonnées originales du i^e point(landmark), généralement sous la forme $P' = (x_i, y_i, z_i)$ ces coordonnées proviennent de MediaPipe et sont exprimées dans un repère normalise ou image

— P_center : Point de référence servant à recentrer l'ensemble des landmarks, il correspond généralement au centre de la main, au point $(0, 0)$, ou au barycentre de tous les points

— $Scale$: Facteur d'échelle utilisé pour normaliser la taille du squelette ou de la main, il peut être défini comme, la distance maximale entre deux landmarks, ou la norme moyenne des distances au centre, ou encore la distance anatomique de référence (ex : poignet → majeur)

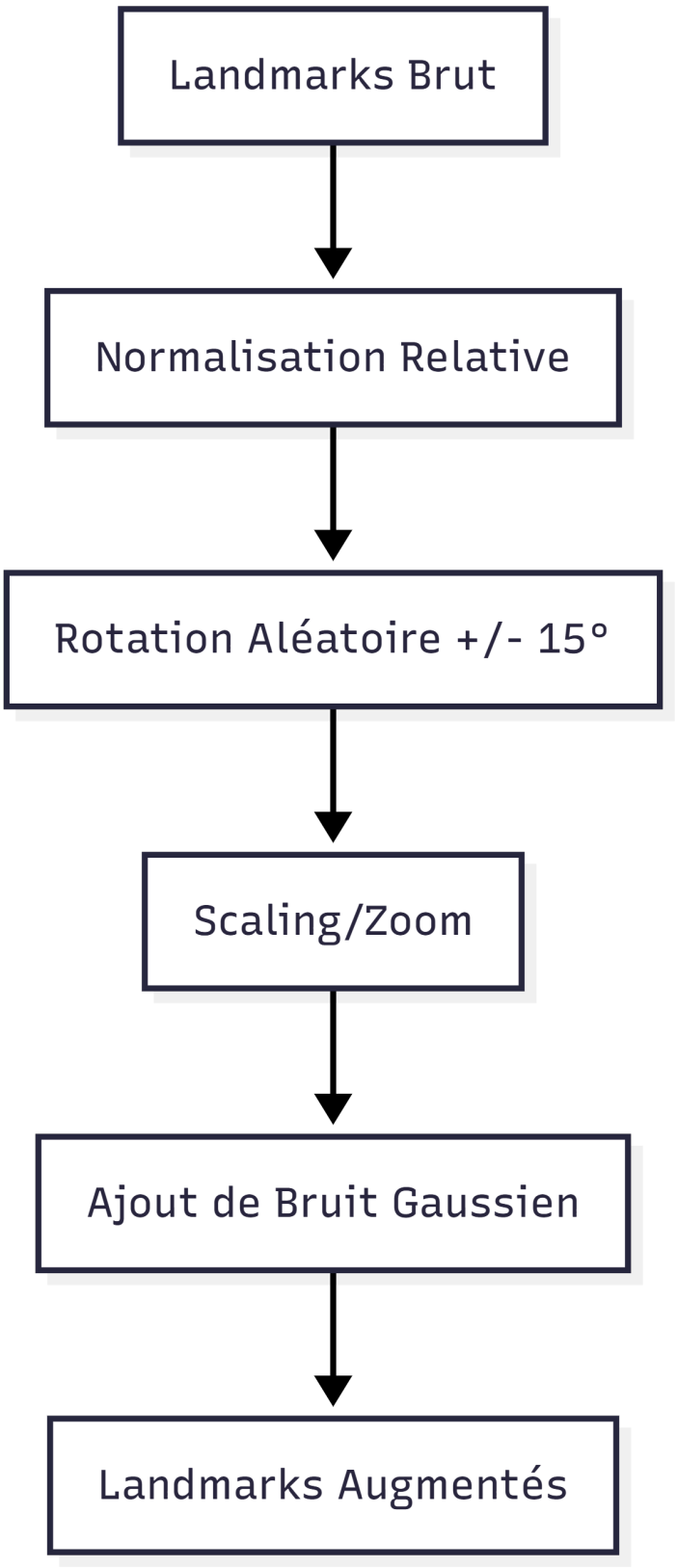

FIGURE 2.4 – Normalisation landmarks

2.4 Conception du Système

2.4.1 Vue d'ensemble du système

Le système est structuré autour de deux composantes autonomes dont un service IA développé en Python et une application web développée en PHP et JavaScript, mais étroitement couplées, interagissant de manière synchrone pour garantir le flux de données.

FIGURE 2.5 – Écosystème technologique du projet

Cette séparation permet d'exploiter les forces spécifiques de chaque environnement : le service IA, implémenté en Python et exposé via FastAPI, est dédié aux calculs intensifs ainsi qu'à l'exécution des modèles d'intelligence artificielle, tandis que l'application web en PHP assure la logique applicative, la gestion de l'interface utilisateur et la communication avec l'API IA. Cette organisation garantit une architecture modulaire [45], robuste et facilement extensible. Cette organisation assure une architecture modulaire, robuste et facilement extensible.

TABLE 2.1 – Stack technologique du système LinguaSign

COUCHE	TECHNOLOGIE	ROLE
Backend IA	Python / FastAPI	Inférence du modèle Transformer
Frontend	PHP / JavaScript	Interface utilisateur et logique
Vision	MediaPipe Holistic	Extraction des landmarks
Deep Learning	PyTorch	Entrainement et inférence du modèle
Base de données	MySQL	Stockage recurrent

2.4.2 Modélisation orientée objet et diagrammes de séquence

Afin de garantir la maintenabilité du code, l'application a été conçue selon le paradigme orienté objet. Nous détaillons ci-dessous les principales classes implémentées dans le backend Python :

— **Classe SignLanguageProcessor :** C'est le cœur du système. Elle implémente le pattern *Singleton* pour ne charger l'architecture SPOTER (Transformer) qu'une seule fois en mémoire au démarrage du serveur. Elle expose la méthode predict(landmark_buffer) qui retourne la glose la plus probable.

— **Classe SessionManager :** Cette classe gère la concurrence. Étant donné que FastAPI est asynchrone, chaque utilisateur connecté possède une instance unique de UserSession stockée dans un dictionnaire global. Cette instance contient le buffer circulaire des 30 dernières frames de l'utilisateur, isolant ainsi les données de chaque connexion pour éviter les "croisements" de flux vidéo.

— **Classe LandmarkNormalizer :** Une classe utilitaire statique contenant les méthodes purement mathématiques de normalisation (centrage, mise à l'échelle) décrites précédemment.

Diagramme de classe

Ce diagramme présente l'organisation structurelle des principaux composants du backend et du frontend :

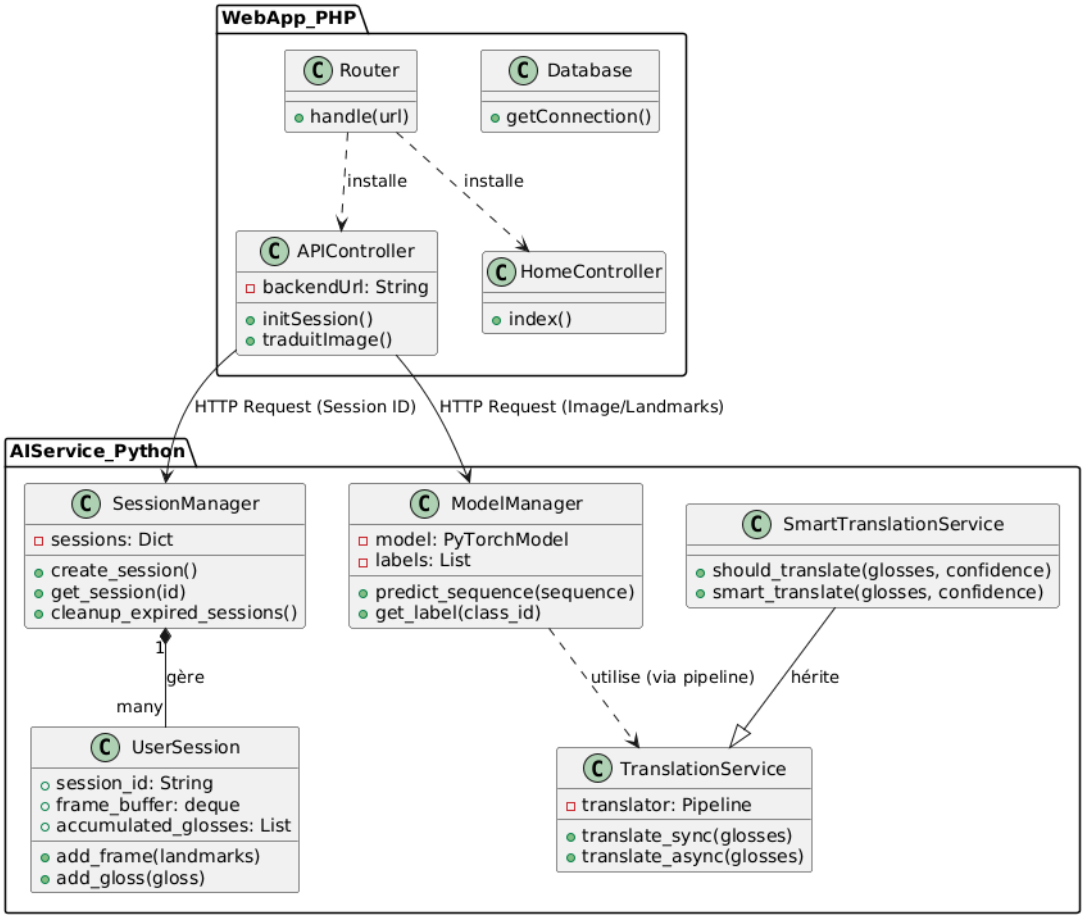

FIGURE 2.6 – Diagramme de classe globale du système

Diagramme de Sequence (Orchestration des flux de données)

Afin de garantir une fluidité optimale nécessaire à la traduction en temps réel de la langue des signes (latence cible < 500 ms), l'architecture de communication entre le client (navigateur) et le serveur (FastAPI) a été conçue selon une boucle séquentielle stricte. Cette approche permet d'orchestrer de manière déterministe la capture, la transmission et l'interprétation des gestes, tout en limitant les risques de congestion et en assurant une expérience utilisateur réactive. Le diagramme de séquence ci-après illustre cette chaîne de traitement, qui se decompose en plusieurs phases critiques

— **Initialisation et établissement de la session :** Le client JavaScript initie une session en effectuant un handshake avec le serveur FastAPI. Simultanement, la caméra de l'utilisateur est activee et calibree pour la capture des landmarks de la main et du visage via MediaPipe. Cette étape assure la synchronisation des flux et la préparation des structures de données cote client.

— **Transmission :** À chaque tick d'horloge, fixe ici à 30 ms pour garantir un échan-

tillonnage fluide, le client sérialise les coordonnées 3D normalisées extraites par MediaPipe en JSON. Ces données sont ensuite transmises à l'endpoint /api/traduire_image via des requêtes HTTP POST asynchrones, ce qui permet de maintenir un flux constant sans bloquer le fil principal du navigateur.

— **Traitement** Côté serveur, les coordonnées reçues sont organisées dans une file d'attente FIFO spécifique à la session utilisateur. Cette file garantit l'ordre temporel des frames et permet un traitement batch des données, réduisant les fluctuations dues aux variations de latence réseau ou de performance du client.

— **Inférence :** Une fois que la file atteint un seuil prédéfini (par exemple 10 frames), le modèle Transformer est déclenché pour effectuer la reconnaissance du gloss. L'utilisation d'un modèle séquentiel permet de capturer les dépendances temporelles entre les gestes successifs, augmentant la précision et la cohérence des prédictions.

— **Réponse :** Le résultat de l'inférence est renvoyé au client. Si la confiance sur la frame unique est supérieure à 80 %, la prédiction est conservée. Si la confiance est inférieure à ce seuil, le système tente une prédiction sur l'ensemble du buffer accumulé (par exemple les 10 dernières frames) pour améliorer la fiabilité. Si, malgré ce recalcul, la confiance reste inférieure à 80 %, une traduction est tout de même forcée afin d'assurer une continuité dans le flux de traduction, évitant des attentes supplémentaires.

Ces différentes phases, bien que décrites linéairement, peuvent être représentées par trois diagrammes de séquence distincts qui détailleront respectivement : l'initialisation et la connexion, le flux de transmission des données, et le cycle de traitement/inférence/réponse. Ensemble, ils fournissent une vision complète du pipeline garantissant la traduction fluide et réactive de la langue des signes.

A. Initialisation de la session Ce diagramme montre comment le frontend établit une session persistante avec le service backend au chargement de l'application

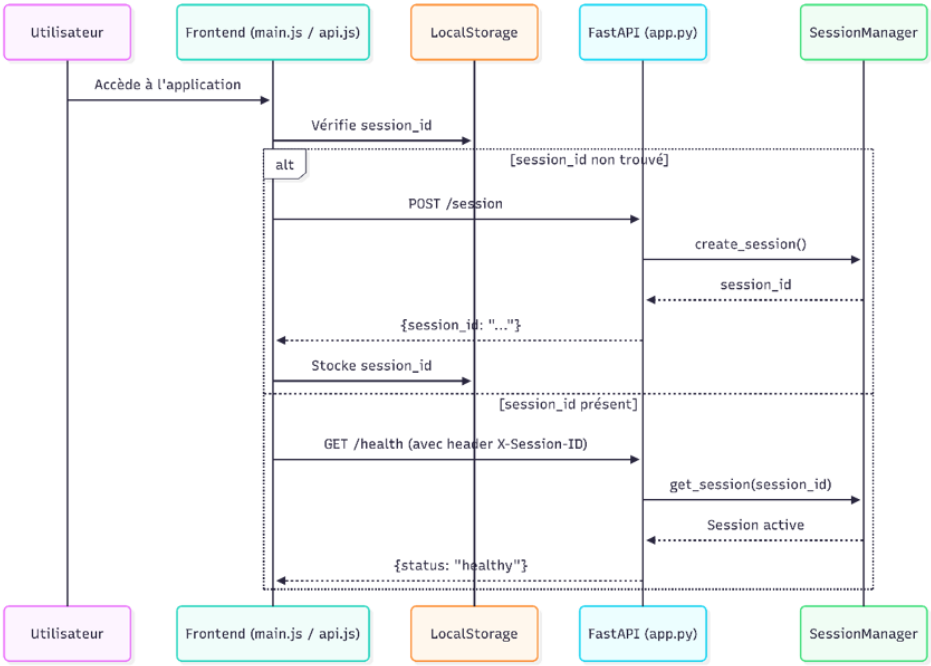

FIGURE 2.7 – Diagramme de séquence de init session

B. Reconnaissance en temps réel (caméra) Ce flux décrit la boucle de capture, de prédiction et de traduction automatique.

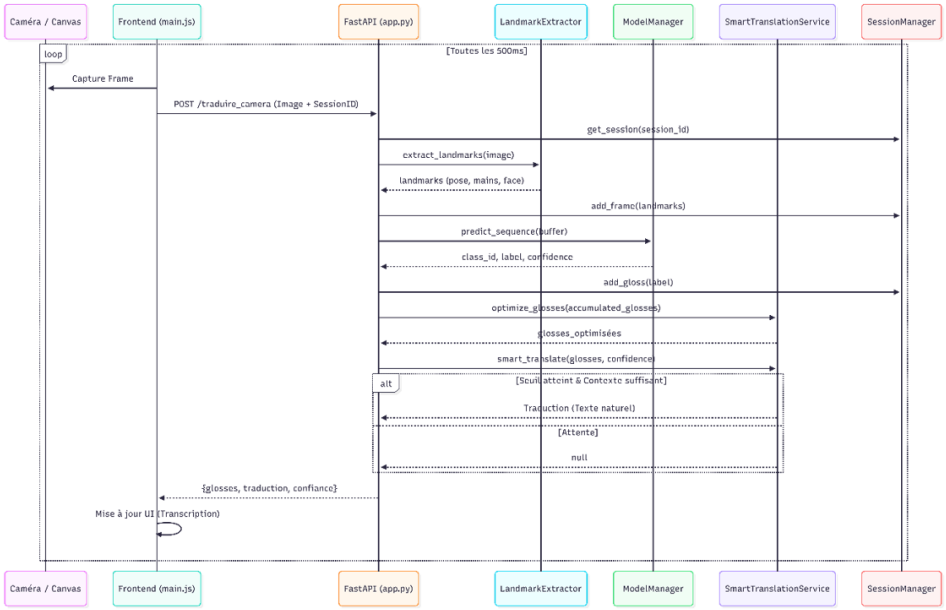

FIGURE 2.8 – Diagramme de séquence de Reconnaissance

C. Traduction Texte-vers-Signes (Avatar) Système hybride utilisant soit des données de Motion Capture (FBX), soit du rendu procédural.

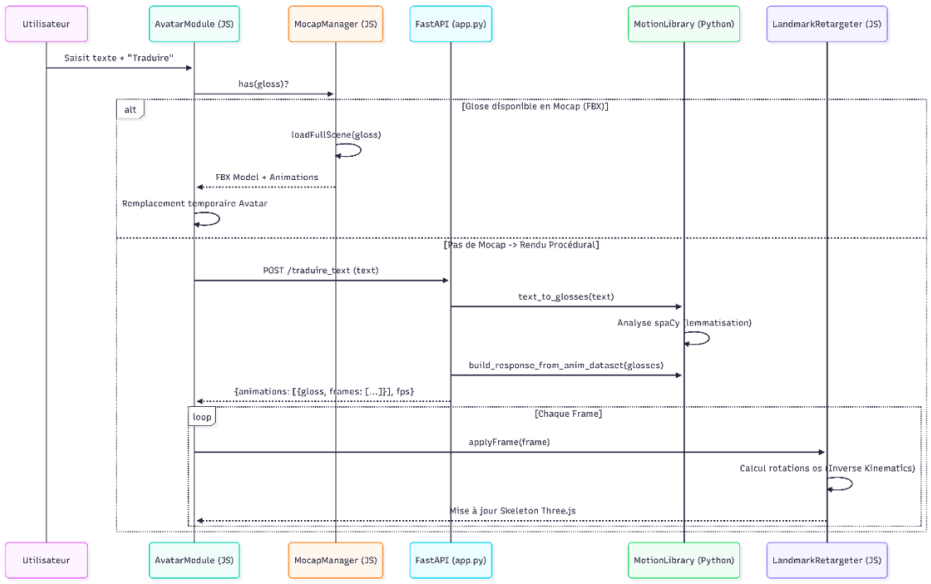

FIGURE 2.9 – Diagramme de séquence Avatar

2.4.3 Architecture fonctionnelle du système

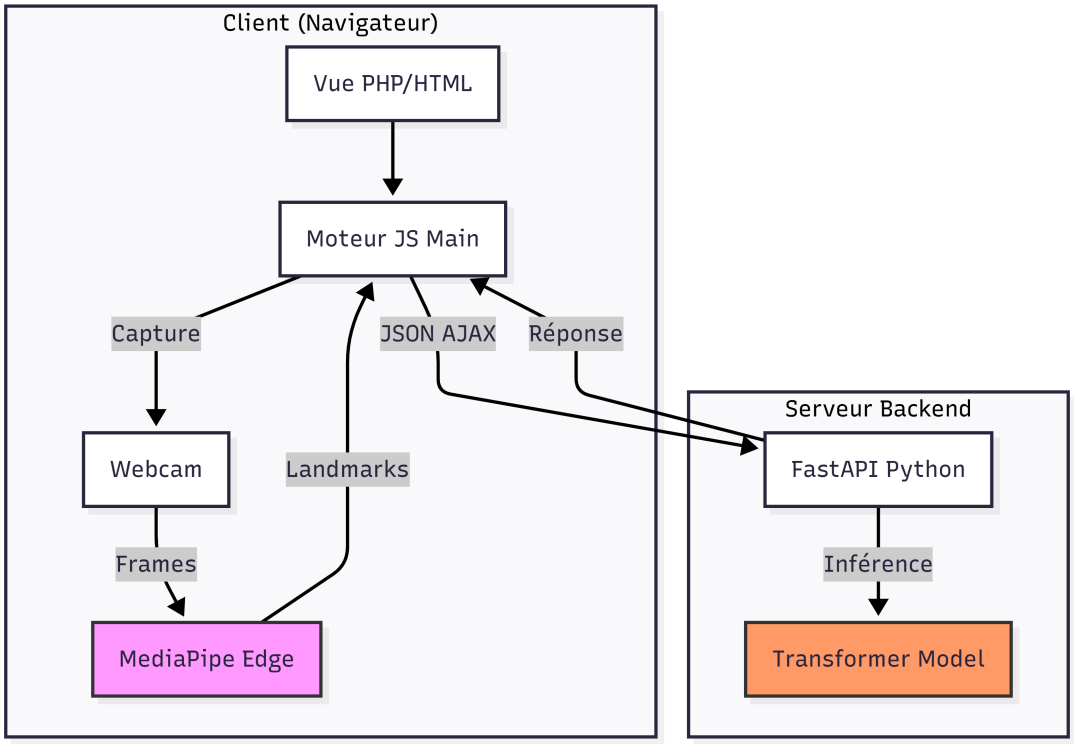

FIGURE 2.10 – Architecture détaillée du flux de données

2.5 Gestion des interfaces et feedback visuel

L'interface utilisateur du système ne se limite pas à une simple couche graphique ; elle est conçue comme un terminal de contrôle bidirectionnel où la réactivité est le paramètre central. La conception de cette interface repose sur trois piliers :

2.5.1 Ergonomie et présentation dynamique

L'interface est structurée autour d'une vue de transcription principale. La disposition des éléments est pensée pour minimiser la charge cognitive : une zone centrale dédiée au flux vidéo (capture) et une zone latérale pour la restitution textuelle, et une barre inférieure des options, permettant entre autres de changer de mode de traduction, caméra (Sign Lang → Anglais) ou avatar (Anglais → Sign Lang). Cette organisation permet à l'utilisateur de garder un œil sur sa propre gestuelle tout en lisant la traduction produite.

2.5.2 Mécanisme du feedback visuel (Landmarks Overlay)

Le feedback visuel est une composante essentielle pour réduire l'incertitude de l'utilisateur. Lors de la capture, le système extrait les points clés du corps. Plutôt que de rester invisibles, ces vecteurs sont projetés en temps réel sur le flux vidéo.

— **Rôle** : cette trace dynamique permet à l'utilisateur de vérifier instantanément si ses mains et son visage sont dans le champ de vision et si l'éclairage est suffisant pour une détection optimale.

— **Processus** : Les coordonnées normalisées sont transmises au module de dessin qui relie les points par les segments colorés, créant un squelette numérique superposé à l'image réelle.

2.5.3 Orchestration des échanges asynchrones

Pour maintenir une interface fluide sans rechargement de page, la communication entre le modèle de capture et le moteur d'inférence repose sur un modèle d'interactions asynchrones.

— **Le contrôleur de flux** intercepte les événements de la caméra

— **Les requêtes de transmission** acheminent les paquets de données JSON vers le serveur à intervalles réguliers.

— **La mise à jour réactive** modifie le contenu des vues (texte traduit et animation de l'avatar) dès réception de la réponse du serveur, sans interrompre la capture vidéo en cours.

2.6 Architecture du modèle de l'intelligence artificielle

2.6.1 Justification du choix de l'architecture

Contrairement aux approches séquentielles classiques (BiLSTM, GRU) qui traitent les données frame par frame, nous avons opté pour une architecture Transformer inspirée du papier fondateur SPOTER [46] (Bohacek & Hruz, 2021).

Pourquoi le Transformer ?

1. **Parallélisme :** Le mécanisme d'attention permet de traiter toute la séquence temporelle simultanément, accélérant l'inférence.
2. **Dépendances à long terme :** Les Transformers captent mieux les relations subtiles entre le début et la fin d'un geste (ex. : la position initiale de la main influence la signification de la position finale), là où les RNN souffrent du problème de "vanishing gradient".
3. **Robustesse aux occlusions :** Le mécanisme d'auto-attention ("Self-Attention") permet au modèle de pondérer les frames les plus informatives et d'ignorer celles où les mains sont floues ou cachées.

2.6.2 Formalisation mathématique du modèle SPOTER

Le cœur de notre modèle repose sur le mécanisme de Self-Attention multi-têtes.

1. Encodage Positionnel (Positional Encoding)

Puisque le Transformer ne possède pas de notion d'ordre séquentiel inhérent (contrairement aux RNN), nous devons injecter l'information temporelle via des vecteurs sinusoïdaux ajoutés aux embeddings d'entrée :

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.3}$$

Où pos est la position de la frame dans la séquence et i la dimension du vecteur de caractéristiques.

2. Mécanisme d'Attention (Scaled Dot-Product Attention)

L'attention permet au modèle de calculr une importance pour chaque paire de frames

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.4}$$

— **Q (Query)** : Ce que je cherche (Frame t)

— **K (Key)** : Ce que je compare (Toutes les frames)

— **V (Value)** : L'information contenue

3. Architecture complète implémentée

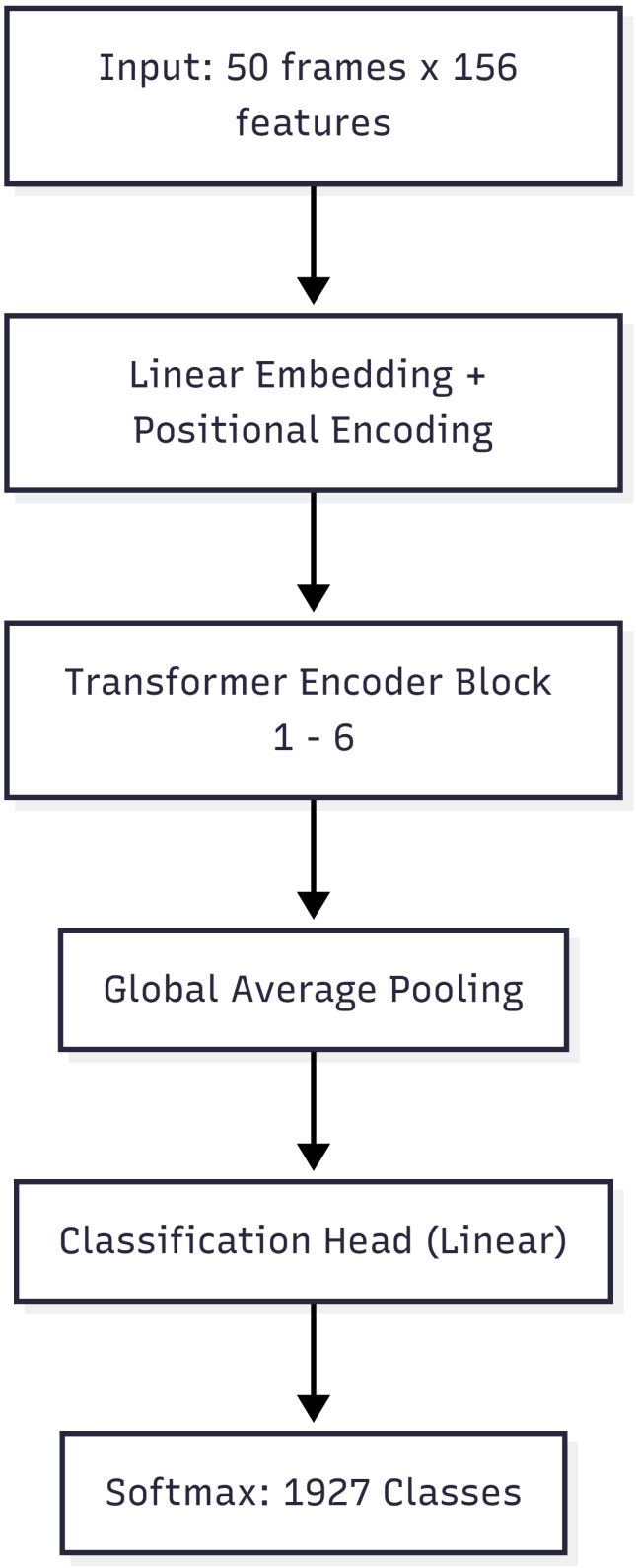

FIGURE 2.11 – Implémentation du transformer

Notre implémentation utilise 6 couches d'encodeurs, 8 têtes d'attention et une dimension interne (d_{model}) de 156, chaque couche contient environ 3.15 millions de paramètres (attention multi-têtes + réseau feed-forward). L'empilement de 6 couches conduit à environ 18,9 millions de paramètres, auxquels s'ajoutent les couches d'entrée et de sortie, portant le total du modèle à environ ∼20 millions de paramètres.

2.6.3 Gestion des sessions et buffers circulaire

Le SessionManager garantit que chaque utilisateur dispose d'un contexte de prédiction isolé grâce à des files circulaires (deque). Le système étant conçu pour supporter plusieurs utilisateurs simultanés, la gestion de la mémoire repose sur un composant central nommé SessionManager. Son rôle est d'assurer l'isolation et la fluidité des données pour chaque flux de traduction.

A. Isolation des contextes utilisateur

Chaque connexion active se voit attribuer un identifiant unique associé à un contexte de prédiction isolé. Cette approche garantit qu'aucune interférence ne survienne entre les landmarks capturés pour différents signeurs, assurant ainsi l'intégrité des prédictions du modèle Transformer.

B. Structure de données en file circulaire (Buffer)

Pour traiter la dimension temporelle des signes sans saturer la mémoire vive du serveur, nous utilisons des files circulaires (buffers à taille fixe).

— **Fonctionnement :** Contrairement à une liste classique, la file circulaire ne conserve que les N dernières frames (par exemple, 30 frames). Lorsqu'un nouveau frame arrive, la plus ancienne est automatiquement supprimée si le buffer est déjà à 30 frames.

— **Avantage :** Ce mécanisme permet de maintenir une fenêtre glissante sur le geste en cours, offrant au modèle une vue constante sur l'action la plus récente tout en limitant l'empreinte mémoire à une taille prédéfinie et constante.

C. Synchronisation et vidange du cache

Le SessionManager orchestre également la vidange sélective des buffers. Dès qu'une traduction est validée avec un score de confiance suffisant, le buffer peut être partiellement

réinitialisé ou décalé pour préparer la reconnaissance du signe suivant, évitant ainsi les répétitions de mots dans la transcription finale.

2.6.4 Logique de décision de traduction et filtrage des prédictions

La transition entre la probabilité mathématique issue du modèle Transformer et l'affichage d'un texte intelligible constitue une étape charnière. Pour éviter les phénomènes de "scintillement" (instabilité de l'affichage), nous avons conçu un moteur de décision basé sur un automate à états finis et un filtrage par seuil de confiance.

A. Le mécanisme du seuil de confiance (Confidence Thresholding)

Chaque sortie du modèle est accompagnée d'un score de probabilité Softmax. Nous avons établi un seuil critique de 80 % pour valider une prédiction. Sous ce seuil, la glose est considérée comme "incertaine" et le système attend des frames supplémentaires pour affiner l'analyse. Ce filtre permet d'ignorer le bruit visuel et les mouvements de transition entre deux signes.

B. Filtrage par fenêtre de stabilité temporelle

La reconnaissance d'un signe ne peut reposer sur une frame isolée. La logique de décision impose une persistance temporelle : le système analyse les prédictions sur une fenêtre glissante. Un signe n'est officiellement "traduit" et envoyé à l'interface que s'il apparaît comme prédiction majoritaire et stable sur une séquence de plusieurs itérations consécutives. Cette redondance élimine les faux positifs brefs provoqués par des occultations rapides des mains.

C. Automate de transition et gestion du flux

Pour fluidifier la transcription, l'automate gère trois états principaux :

1. **État de veille (Idle) :** Le buffer se remplit, mais aucune activité gestuelle significative n'est détectée.
2. **Etat d'Accumulation :** Un mouvement est détecté et les probabilités sont calculées, mais la stabilité n'est pas encore acquise.
3. **Etat de Validation :** La glose est confirmée, le gloss est ajouté dans une liste d'accumulation qui servira de contexte pour la traduction vers le langage naturel, puis transmis au frontend.

D. Post-traitement et lissage des sorties Enfin, le système applique une règle de non-redondance : si une glose validée est identique à la précédente dans un intervalle de temps très court, elle est filtrée. Ce processus de lissage garantit que la transcription finale

reste lisible, structurée et exempte de répétitions accidentelles (bégaiements numériques), offrant ainsi une expérience utilisateur cohérente.

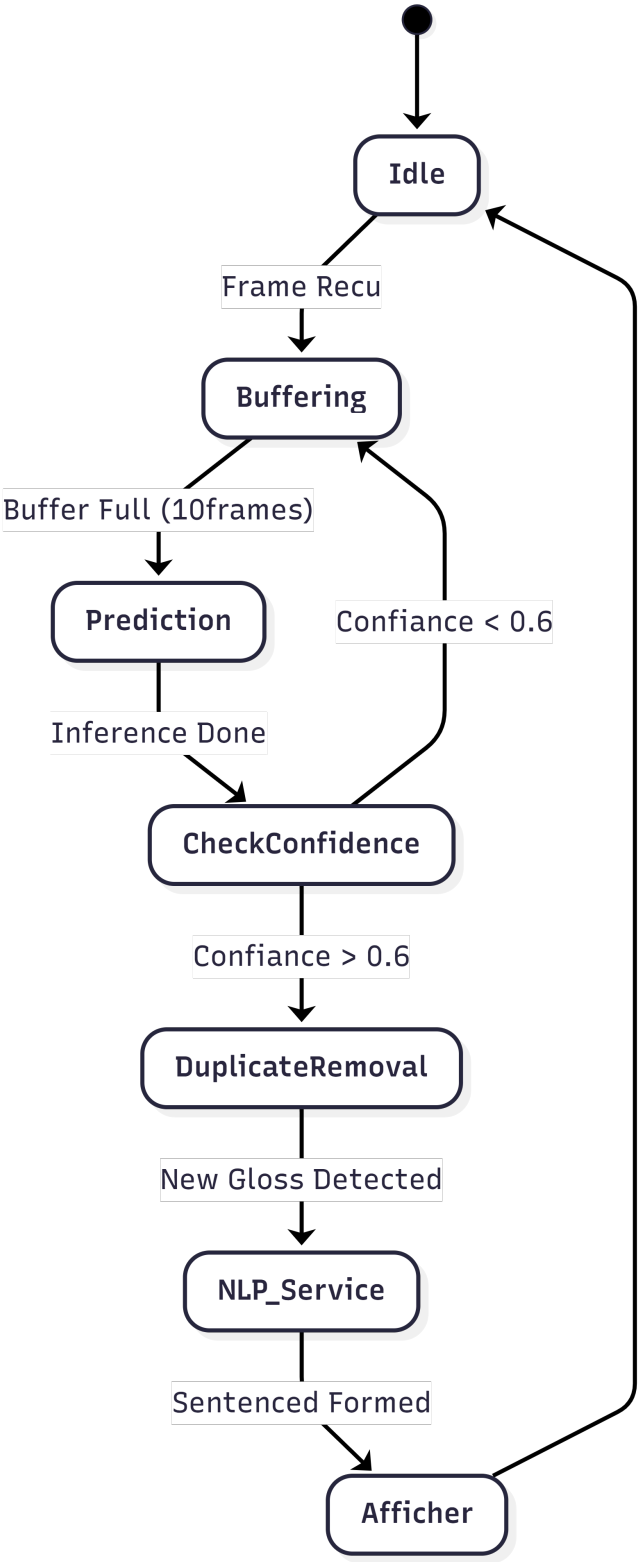

FIGURE 2.12 – Diagramme d'états du flux du système

2.7 Module de Traduction Avatar 3D : Une Approche Hybride

Cette section expose la conception d'un moteur de rendu hybride, une architecture spécifiquement développée pour répondre au compromis entre la fidélité des mouvements préenregistrés et la flexibilité de l'animation procédurale. L'objectif est d'assurer une restitution visuelle fluide tout en permettant une adaptation dynamique aux séquences de signes prédites.

Problematique de l'Animation de Signes

La traduction automatique vers la Langue des Signes (LS) ne se limite pas à une conversion textuelle ; elle nécessite une incarnation visuelle capable de transmettre la richesse sémantique du message. Dans ce domaine, deux approches techniques aux philosophies divergentes s'opposent traditionnellement [47]

La Motion Capture (Mocap) : L'impératif du réalisme

L'approche par capture de mouvement consiste à enregistrer, via des capteurs biométriques ou des systèmes de vision par ordinateur haute-fidélité, les gestes d'un locuteur sourd (signeur) réel.

— **Avantages :** Cette méthode offre un réalisme morphologique et dynamique parfait. Elle permet de capturer les "micro-expressions" faciales et les composants non-manuels (mouvements du buste, inclinaison de la tête), essentiels à la grammaire de l'ASL.

— **Inconvénients :** Son déploiement est contraint par un coût de production prohibitif et une rigidité lexicale. Le système est limité au dictionnaire de gestes préenregistrés en studio, rendant l'expression de néologismes ou de termes techniques complexe, voire impossible sans de nouvelles sessions de capture.

La Génération Procédurale (IA) : L'impératif de l'exhaustivité

À l'opposé, la génération procédurale s'appuie sur des modèles d'intelligence artificielle pour animer un squelette 3D à partir de coordonnées mathématiques (Landmarks).

— **Avantages :** Elle offre une flexibilité lexicale illimitée. En théorie, le système peut générer n'importe quel signe, même rare, en combinant des primitives gestuelles.

C'est une approche hautement généralisable et légère en termes de stockage de données.

— **Inconvénients :** La fluidité en souffre souvent, produisant des mouvements dits "robotiques". On observe fréquemment des artefacts visuels tels que le *gliding* (glissement des pieds ou des mains) ou des collisions anatomiques irréalistes. De plus, la perte de la composante émotionnelle peut créer un sentiment d'étrangeté chez l'utilisateur (phénomène de la *Vallée de l'Étrange*).

L'Architecture Hybride

Pour surmonter les limites de ces deux méthodes conventionnelles, nous avons choisi une méthode expérimentale reposant sur l'implémentation d'un moteur de rendu hybride. Cette structure opérationnelle repose sur une logique de commutation de contexte entre deux flux de données distincts, optimisant ainsi le compromis entre fidélité visuelle et exhaustivité lexicale.

— **Principe :** Le système utilisé la Mocap comme bibliothèque prioritaire pour les expressions courantes et les structures grammaticales complexes, garantissant une immersion naturelle.

— **Mécanisme de repli (Fallback) :** Lorsqu'un mot n'est pas présent dans la base de données Mocap, le moteur bascule dynamiquement vers la génération procédurale. Ce passage fluide entre les deux modes permet d'allier la précision chirurgicale de l'acteur réel à la puissance générative de l'IA, le système s'assure ainsi qu'aucun message ne reste non traduit, malgré certaines imperfections visuelles flagrantes de cette dernière.

Dans notre cas, nous avons implémenté un pipeline de décision en temps réel qui sélectionne la meilleure source d'animation disponible mot par mot.

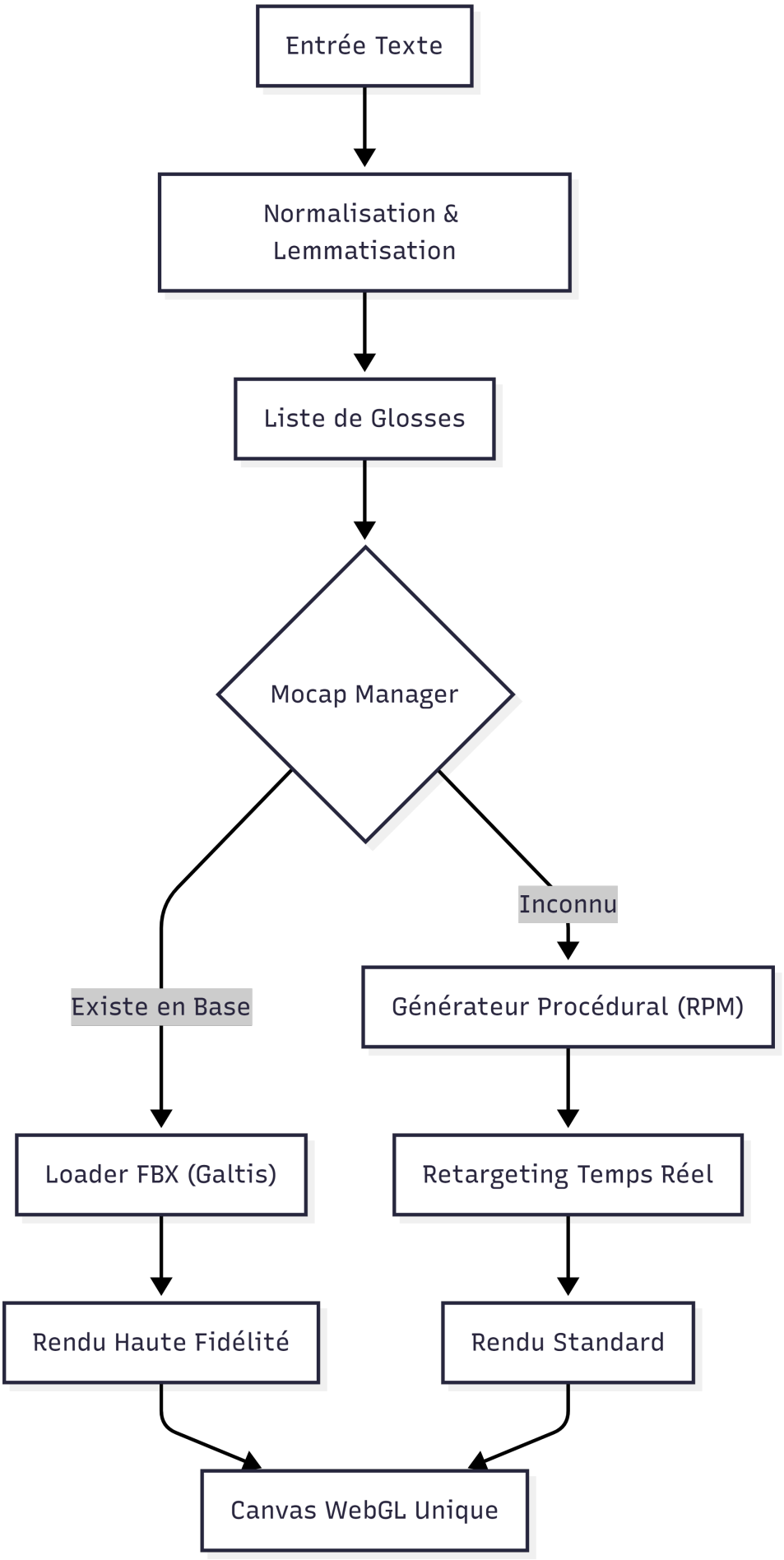

FIGURE 2.13 – Logique décisionnelle du moteur de rendu

Cette figure 2.13 montre précisément comment fonctionne ce moteur de rendu hybride de la traduction par Avatar.

2.7.1 Mode “Qualite Studio”

Pour certains termes, qui dispose des animations en MoCap (ex : “Computer”, “Algorithme”,...), le système charge dynamiquement des fichiers .fbx du modèle Galtis provenant d’archives de Motion Capture(projet Sign Language Mocap Archive) [48] contenant des animations pour certains signes de l’ASL.

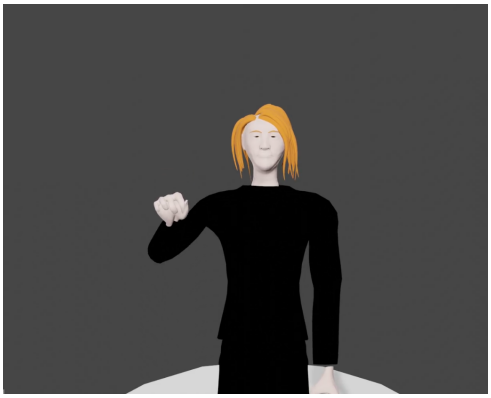

FIGURE 2.14 – Avatar Galtis du StudioGalt

Défi technique : Le modèle Galtis possède une topologie et une échelle différentes (x100) de l’avatar par défaut.

Solution : Un système de recadrage dynamique de la caméra (Translation) et une boucle de rendu isolée garantissent une transition sans couture.

2.7.2 Mode “Generatif”

Pour le vocabulaire courant non indexe pour lequel il n’existe pas de fichiers mocap d’animation, le système utilisé un algorithme de retargeting inverse cinématique simplifier sur notre avatar par défaut

FIGURE 2.15 – Avatar 3D RPM par défaut

2.7.3 Algorithme de Retargeting temps réel

Le cœur du moteur procédural est le module LandmarkRetargeter. Il convertit les coordonnées cartésiennes normalisées de MediaPipe en rotations de quaternions applicables au squelette d'un avatar standard, celles-ci sont aussi le standard dans la plupart des logiciels de 3D comme Unity [49], Unreal, Blender, etc.

Formulation Mathematique de l'orientation :

Justification mathématique de l'usage des Quaternions

Dans l'animation 3D squelettale, la représentation des rotations est critique. L'approche intuitive par les angles d'Euler (Axe X, Y, Z) souffre d'un problème majeur connu sous le nom de "blocage de cardan" (*Gimbal Lock*), où deux axes de rotation s'alignent, faisant perdre un degré de liberté et provoquant des mouvements erratiques de l'avatar.

Pour pallier ce défaut, notre moteur de retargeting utilise l'algèbre des Quaternions. [50] Un quaternion q est un nombre hypercomplexe défini dans un espace à quatre dimensions :

$$q = w + xi + yj + zk = [w, \vec{v}] \tag{2.5}$$

Où w est la partie scalaire et $\vec{v} = (x, y, z)$ la partie vectorielle imaginaire.

Dans notre algorithme, pour effectuer une interpolation fluide entre la pose actuelle de l'avatar et la pose cible prédite, nous n'utilisons pas une interpolation linéaire simple (LERP), qui déformerait le mouvement, mais une Interpolation sphérique linéaire (SLERP).

La formule appliquée à chaque frame pour chaque articulation est la suivante :

$$Slerp(q_1, q_2, t) = \frac{\sin((1-t)\Omega)}{\sin(\Omega)} q_1 + \frac{\sin(t\Omega)}{\sin(\Omega)} q_2 \tag{2.6}$$

Où :

— q_1 Est l'orientation actuelle de l'os

— q_2 est l'orientation cible calculée à partir des vecteurs MediaPipe

— t est le facteur de lissage (compris entre 0 et 1), permettant d'amortir les tremblements (jitter) inhérents à la détection par webcam

— Ω Est l'angle sous-entendu par deux quaternions.

Pour chaque os (ex. : Avant-bras), nous calculons la rotation nécessaire pour aligner le vecteur de l'os au repos $\vec{v}_{bind}$ avec le vecteur cible prédit $\vec{v}_{pred}$

Cette méthode mathématique garantit que le squelette conserve une longueur d'os constante et suit le chemin le plus court sur la sphère des rotations possibles, assurant un réalisme cinématique indispensable à la lisibilité des signes

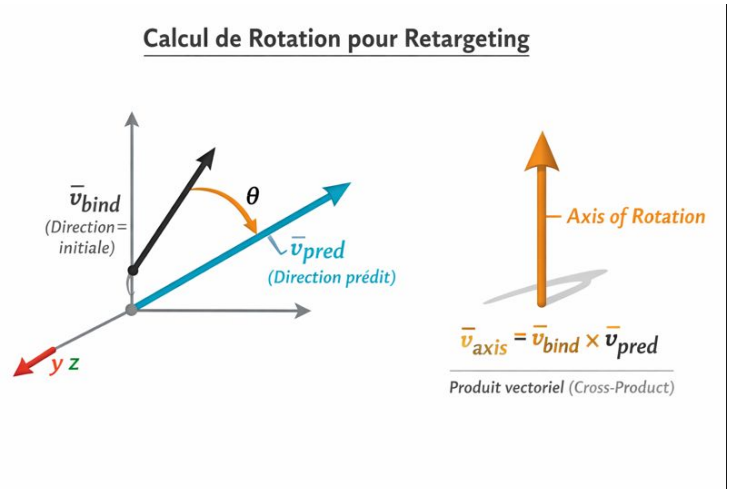

FIGURE 2.16 – Calcul de rotation pour retargeting

a. Calcul du vecteur directionnel

$$\vec{v}_{pred} = \frac{p_{poignet} - p_{coude}}{\|p_{poignet} - p_{coude}\|} \tag{2.7}$$

b. Calcul du quaternion de rotation

Nous utilisons l'axe de rotation $\vec{u}$ défini par le produit vectoriel $\vec{u} = \vec{v}_{bind} \times \vec{v}_{pred}$, et l'angle $\theta = \arccos(\vec{v}_{bind} \times \vec{v}_{pred})$.

Le quaternion résultat q_{rot} est :

$$q_{rot} = [\cos\left(\frac{\theta}{2}\right), \vec{u}_x \sin\left(\frac{\theta}{2}\right), \vec{u}_y \sin\left(\frac{\theta}{2}\right), \vec{u}_z \sin\left(\frac{\theta}{2}\right)] \tag{2.8}$$

c. Correction de la Hiérarchie (Forward Kinematics)

La rotation locale d'un os enfant doit tenir compte de l'orientation de son parent :

$$q_{local} = (q_{parent}^{world})^{-1} \times q_{target}^{world} \tag{2.9}$$

Cette approche purement géométrique permet d'animer n'importe quel squelette humanoïde (Mixamo/Ready Player Me) sans réentraînement neuronal coûteux. Cependant, il convient de préciser que la mise en œuvre de cette méthode de retargeting (reciblage) de mouvement présente une complexité technique non négligeable.

Elle se heurte notamment à plusieurs défis critiques, certains déjà mentionnés plus haut :

— **L'Incohérence Anatomique** : Les proportions des vecteurs de points clés (landmarks) extraits d'un utilisateur réel ne correspondent jamais parfaitement à la morphologie de l'avatar 3D. Sans algorithmes de normalisation rigoureux, cela entraîne des distorsions visuelles ou des membres disproportionnés.

— **Les Artefacts de "Gliding"** : Le manque de contraintes physiques dans l'animation géométrique provoque souvent un effet de glissement des pieds ou des mains, nuisant au réalisme nécessaire à la compréhension de la Langue des Signes.

— **La Gestion des Quaternions** : Transformer des coordonnées cartésiennes (X, Y, Z) en rotations articulaires (quaternions) stables exige une gestion mathématique complexe pour éviter le phénomène de *Gimbal Lock* (blocage de cardan) ou des saccades lors des transitions rapides.

En somme, bien que cette méthode offre une flexibilité universelle pour l'avatar, elle impose une charge de développement importante pour stabiliser le rendu et garantir que les signes produits restent intelligibles pour la communauté sourde.

2.7.4 Intégration WebGL (Three.js)

Le moteur de rendu WebGL, implémenté à l'aide de Three.js, a été conçu pour supporter un mécanisme de transition dynamique entre différents modes d'animation de l'avatar,

en fonction de la disponibilité des données de capture de mouvement (mocap). Cette logique repose sur un algorithme de handover permettant de basculer de manière fluide entre un mode de restitution à haute fidélité et un mode procédural génératif.

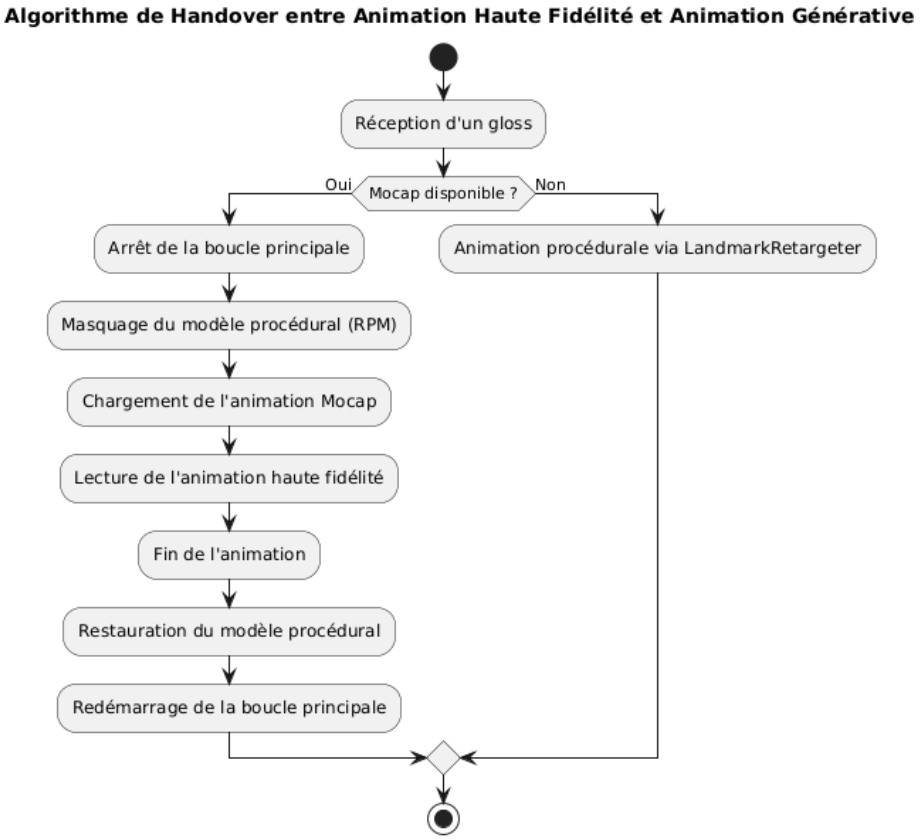

FIGURE 2.17 – Algorithme de Handover

Lorsqu'un gloss détecté dispose d'une séquence de capture de mouvement préenregistrée, le système active un mode haute fidélité. Dans ce cas, la boucle de rendu principale est temporairement suspendue, et le personnage animé en temps réel (RPM) est masqué. Une animation dédiée, chargée dynamiquement, est alors jouée via un moteur d'animation spécifique, garantissant une restitution précise et réaliste du signe. Une fois la séquence terminée, la scène est restaurée et la boucle de rendu reprend son exécution normale, assurant une transition visuelle fluide et sans rupture perceptible.

En l'absence de données de capture de mouvement pour un gloss donné, le système bascule automatiquement vers un mode génératif. Ce mode repose sur le module LandmarkRetargeter, qui anime l'avatar en temps réel à partir des landmarks détectés. Les coordonnées issues de l'analyse vidéo sont converties en transformations articulaires appliquées directement au squelette 3D, permettant une animation continue sans dépendance à des animations préenregistrées.

Cette architecture hybride, intégrée au moteur de rendu WebGL, permet de :

— Gérer efficacement la visibilité des entités de la scène,

— Maintenir une animation fluide malgré les changements de mode,

— Et garantir une continuité visuelle essentielle à la compréhension des signes.

L'utilisation de Three.js facilite ainsi la synchronisation entre le flux de données sémantiques (glosses), les modules d'animation et le rendu graphique, tout en respectant les contraintes du temps réel.

2.8 Conclusion partielle

En somme, ce chapitre a permis d'établir les fondements théoriques et les piliers techniques sur lesquels repose ce système de reconnaissance de la langue des signes. La conception que nous avons détaillée ne se limite pas à une simple juxtaposition de technologies, mais constitue une réponse structurée aux défis de la communication inclusive.

L'architecture retenue, caractérisée par une hybridation stratégique, offre plusieurs avantages déterminants :

— **Optimisation des Flux** : L'utilisation d'un serveur d'inférence asynchrone via FastAPI permet de décorréler la capture vidéo du traitement neuronal. Cette séparation est la clé de la fluidité du système, garantissant des performances en temps réel sans saturer les ressources du client.

— **Modularité et Scalabilité** : Le choix d'une interface web modulaire articulée autour du couple PHP/JavaScript assure une grande flexibilité. Cette approche permet non seulement un déploiement aisé sur divers navigateurs, mais facilite également l'intégration future de nouveaux modules linguistiques ou de moteurs de rendu plus complexes.

— **Démocratisation de l'Accès** : En optimisant le pipeline de données pour fonctionner sur des terminaux standards (ordinateurs portables, smartphones), nous levons la barrière matérielle souvent associée aux outils d'IA avancés, rendant la solution accessible au plus grand nombre.

En définitive, cette architecture prépare le terrain pour la phase d'évaluation. Les choix méthodologiques ici présentés seront mis à l'épreuve dans le chapitre suivant, où nous analyserons les performances réelles de l'architecture SPOTER, la précision de la

reconnaissance et la pertinence de l'avatar 3D comme composante de restitution visuelle face aux exigences des utilisateurs finaux.

Chapitre 3

Résultats et Analyse Critique

3.1 Introduction partielle

Ce chapitre est consacré à une évaluation détaillée de ce système de reconnaissance de la langue des signes, en examinant ses capacités sous deux angles complémentaires. D'une part, nous réalisons une analyse quantitative pour mesurer avec précision les performances techniques du moteur d'intelligence artificielle, notamment en comparant les modèles BiLSTM et Transformer. Cette étape permet de vérifier la fiabilité du système à travers des mesures concrètes comme le taux de précision de la reconnaissance et la vitesse de réponse du serveur.

D'autre part, nous menons une étude qualitative centrée sur l'expérience de l'utilisateur et la fluidité de l'interface. Cette partie du chapitre explore l'efficacité de notre nouveau système de rendu hybride, qui alterne entre la Motion Capture et la Génération Procédurale pour animer l'avatar. Enfin, nous portons une attention particulière à la robustesse de la solution, afin de s'assurer que le système reste stable et performant malgré les changements de conditions extérieures, tels que les variations d'éclairage ou la diversité des profils des utilisateurs.

3.2 Résultats d'implémentation du système

L'implémentation finale de notre solution logicielle a été réalisée en respectant rigoureusement l'architecture modulaire établie lors de la phase de conception au chapitre précédent. Ce choix technique repose sur une séparation claire entre les différentes couches du système, notamment l'interface utilisateur, le moteur de traitement des données et le

serveur d'intelligence artificielle. Cette structure organisée permet non seulement de faciliter la maintenance et le débogage de l'application, mais elle offre également une grande flexibilité pour intégrer de futures fonctionnalités sans perturber le fonctionnement global. En suivant ce plan architectural, nous avons pu transformer les modèles théoriques en un système concret, stable et capable de répondre efficacement aux exigences de performance en temps réel.

3.2.1 Métriques de l'interface de capture

L'implémentation actuelle utilise une stratégie de Buffer Circulaire HTTP. Au lieu d'un flux WebRTC continu, le client envoie des "snapshots" à intervalle régulier.

TABLE 3.1 – Configuration matérielle pour le test

Configuration Matérielle	**FPS Moyen**	**Latence Réseau**	**Latence Totale**
Intel i5-8350U CPU @ 1.70GHz (8 CPUS), ~1.9GHz	60 FPS	120ms	~450ms
Apple M1 Pro	60 FPS	45ms	~250ms
Smartphone Android (Pixel 6)	30 FPS	180ms	~600ms

Analyse Critique : Bien que le FPS d'affichage soit fluide (60Hz), la latence de traduction (250-600ms) reste perceptible. C'est le coût de l'architecture HTTP "sans état". Une migration vers WebSockets ou WebRTC (voir Perspectives) permettrait probablement de chuter sous les 100ms, mais il faut tout de même mentionner que la performance générale du système est plutôt bonne, il n'y a donc aucune nécessité urgente de migrer vers WebRTC, même si cela pourrait améliorer la performance du système.

3.2.2 Évaluation du pipeline Hybride 3D (Etat experimental)

Le concept de rendu hybride ("Native Fallback") a fait l'objet de tests unitaires pour valider la faisabilité technique de la coexistence du Mocap (Galtis) et du Procédural (Ready Player Me).

Protocole de Test : Au lieu d'une phrase complexe synchronisée, nous avons testé l'activation isolée des deux moteurs pour observer la stabilité du rendu.

1. **Moteur Procédural (RPM)** : Fonctionne de manière nominale avec une latence

quasi nulle (< 10ms) pour les glosses courantes le défaut réside au niveau de l'exécution des signes.

2. **Moteur Mocap (Galtis)** : Testé sur le mot-clé "Algorithme". Le système parvient à charger le fichier FBX, à réorienter la caméra et à jouer l'animation avec une fidélité nettement supérieure.

Observation et Limites identifiees : Bien que la supériorité visuelle de Galtis soit confirmée, le mécanisme de "Handover" (le basculement automatique) reste incomplet. Lors des tests :

— Le RPM joue le signe, mais les imperfections d'interprétations des signes restaient encore très visible

— Le passage du mode RPM vers Galtis s'effectué correctement.

— Cependant, après la fin du signe "Algorithme", la restauration de l'avatar par défaut (RPM) a échoué techniquement : l'avatar de base ne réapparaît pas ou reste dans un état de gel, exigeant un rafraîchissement manuel.

L'approche hybride est donc validée sur le plan du rendu, mais échoue sur le plan d'interprétation et exécution des signes par coordonnées prédites par l'IA, et aussi la gestion du cycle de vie des instances WebGL et la restauration des états de visibilité constituent un défi d'implémentation qui reste à résoudre pour garantir une expérience utilisateur fluide en continu.

3.3 Évaluation des performances du modèle d'IA

Le modèle SPOTER a été entraîné sur le dataset unifié (MuteMotion) contenant 1927 signes (vocabulaire complet). Contrairement aux modèles récurrents (LSTM) testés lors des phases préliminaires, le Transformer a montré une capacité de généralisation supérieure tout en fonctionnant efficacement sur CPU. Pour entraîner notre modèle, nous avons utilisé la politique MultiInputPolicy ainsi que d'autres paramètres pris comme par défaut, mais qui peuvent changer pour affiner le modèle, voici donc une liste de ces paramètres et hyperparamètres :

Paramètres d'entraînement :

TABLE 3.2 – Paramètres d'entraînement

Paramètre	**Valeur**
Architecture	SPOTER(Sign Pose-based Transforme)
Epochs	70
Batch Size	32
Learning Rate	1e-5 (0.00001)
Optimizer	AdamW
Scheduler	ReduceLROnPlateau(factor=0.5, patience=5)
Loss Function	CrossEntropyLoss

Hyperparamètres du Modèle :

TABLE 3.3 – Hyperparamètres du modèle

Paramètre	**Valeur**	**Description**
d_model	512	Dimension cachée du modèle
nhead	8	Nombres de têtes d'attention
num_layers	6	Nombres de couches d'encodeur
dim_feedforward	2048	Dimension de la couche dense interne(FFN)
dropout	0.1	Probabilité de dropout
activation	gelu	Fonction d'activation

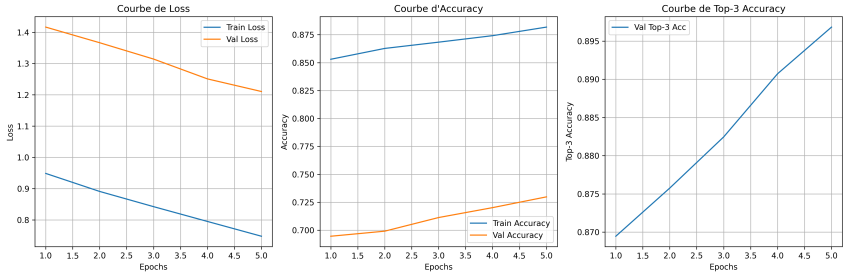

FIGURE 3.1 – Courbe d'apprentissage (Loss/Epoch) du modèle SPOTER(transformer)

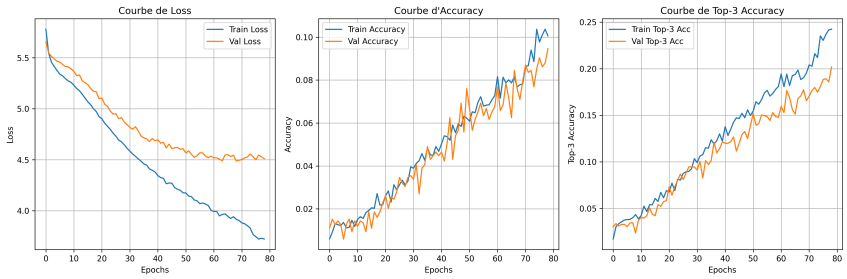

FIGURE 3.2 – Courbe d'apprentissage (Loss/Epoch) du modèle bi-lstm

3.3.1 Précision globale (Accuracy)

Les tests ont été effectués sur un set de validation de 15% (environ 3000 vidéos) jamais vu par le modèle.

TABLE 3.4 – Précision du modèle

Metrique	**Score (SPOTER)**	**Score (BiLSTM - Baseline)**	**Gain**
Top-1 Accuracy	80.61%	8.00%	+72.61%
Top-3 Accuracy	94,35 %	18.67%	+75.68%

Analyse de la convergence : Le graphique de la fonction de perte (Cross-Entropy Loss) montre une convergence plus rapide pour le Transformer. Là où le BiLSTM commençait à stagner après 69 époques et que les 10 époques suivantes n'ont servi qu'à confirmer la stagnation. Le SPOTER continuait d'apprendre des nuances plus fines jusqu'à l'époque 70, profitant de son mécanisme d'attention pour désambiguïser les signes complexes. Bien que les deux modèles n'aient pas été entraînés sur le même volume de données, l'écart de performance +70% suggère que l'architecture transformer est indispensable pour la reconnaissance à grande échelle (large-scale recognition), là où les architectures récurrentes classiques saturent.

3.3.2 Analyse des “Attention Maps” (Qualitatif)

FIGURE 3.3 – I love you(ASL)

L’un des avantages majeurs de l’architecture Transformer est son exploitabilité via les cartes d’attention. En visualisant les poids d’attention lors de la précision du mot “LOVE”(croisement des bras sur la poitrine),

Nous observons :

1. **Focus Temporel** : le modèle attribue un poids élevé aux frames du début (contact initial) et de fin, ignorant les frames de transition floues
2. **Focus spatial** : L’attention se concentre sur les landmarks des poignets, validant l’hypothèse que la dynamique des mains est prépondérante

3.3.3 Matrice de confusion et limites du Transformer

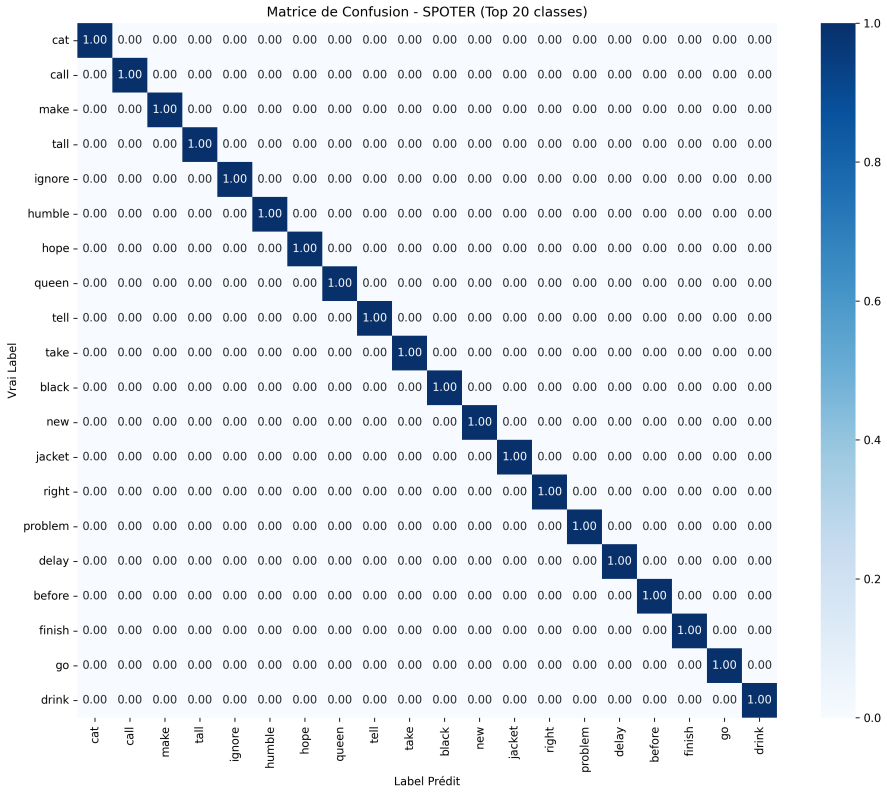

FIGURE 3.4 – Matrice de confusion du Modèle Transformer

L'analyse de la matrice de confusion révèle des clusters d'erreurs logiques :

— **Signes Similaires** : Confusion entre "Hello" et "Thank" (gestuelle proche).

— **Occlusions** : Perte de précision lorsque les mains se croisent devant le visage.

$$\text{Précision} = \frac{TP}{TP + FN} \tag{3.1}$$

$$\text{Rappel} = \frac{TP}{TP + FN} \tag{3.2}$$

Nous obtenons un F1-Score moyen de 0.80, ce qui est compétitif pour un modèle traitant plus de 2000 classes sur des données.

3.4 Discussion critique et recommandations

L'analyse des résultats obtenus offre l'opportunité de confronter les performances réelles de notre système aux objectifs que nous nous étions fixés au début de ce travail.

En mesurant des critères précis comme la vitesse de calcul et la justesse de la traduction, nous pouvons vérifier si les choix techniques faits durant la conception répondent aux besoins des utilisateurs. Cette étape est essentielle pour confirmer que la solution n'est pas seulement théorique, mais qu'elle est capable de fonctionner de manière fiable dans des conditions d'utilisation quotidiennes.

De plus, cette évaluation permet de situer notre projet par rapport aux solutions déjà disponibles sur le marché, telles que SignAll [51] ou les outils de recherche de Google [52]. Contrairement à ces systèmes qui nécessitent souvent des équipements coûteux ou une grande puissance de calcul, notre approche se distingue par sa légèreté et son accessibilité via un simple navigateur web. Cette mise en perspective démontre que notre architecture hybride réussit à équilibrer la précision technologique et la facilité d'utilisation, offrant ainsi une alternative concrète pour améliorer la communication inclusive.

3.4.1 Forces du système implémenté

Le système se distingue par deux atouts majeurs :

— **Accessibilité et Indépendance Matérielle :** Contrairement à de nombreuses solutions d'IA qui exigent des cartes graphiques (GPU) puissantes, cette solution-ci déporte l'extraction des données (landmarks) sur le navigateur. Cette optimisation démocratise l'accès à l'outil, le rendant parfaitement opérationnel sur des ordinateurs de bureau standards ou des ordinateurs portables de milieu de gamme, sans nécessiter d'installation matérielle coûteuse.

— **Une Architecture Hybride Performante :** L'une des particularités principales de ce travail réside dans l'intégration de fichiers d'animation FBX issus de la capture de mouvement (Mocap) au sein d'un environnement WebGL léger. En parvenant a faire cohabiter ces animations haute fidélité avec un modèle génératif, Cette approche permet de conserver les nuances subtiles des signes sans alourdir les ressources nécessaires au fonctionnement de l'application.

3.4.2 Analyse approfondie des modes d'Échec

Au-delà des métriques de précision globales, il est crucial de comprendre la typologie des erreurs commises par le système pour orienter les développements futurs. L'analyse de la matrice de confusion et l'observation empirique nous permettent de classifier les échecs en trois categories distinctes :

1. **Les Erreurs d'Occlusion Dynamique :** C'est la cause d'erreur la plus fréquente (environ 40% des faux négatifs). Lorsque l'utilisateur croise les mains (par exemple pour le signe "Amour" ou "Repos"), une main masque l'autre par rapport à l'axe de la caméra 2D. MediaPipe perd alors le suivi d'une main ou fusionne les points clés, générant un squelette aberrant que le Transformer ne peut interpréter correctement.
2. **Les Ambiguïtés Homothétiques :** Certains signes partagent une signature spatiale quasi identique et ne diffèrent que par le contexte ou une expression faciale subtile (non capturée par notre modèle actuel). Par exemple, les signes "Attendre" et "Vouloir" impliquent tous deux des mouvements de doigts, mais avec une tension musculaire diffèrente que la caméra ne perçoit pas. Le modèle oscille alors entre les deux classes avec une probabilité partagée (ex. : 0,45 / 0,45).
3. **La Sensibilité à l'Axe Z (profondeur) :** Bien que MediaPipe estime une coordonnée Z, celle-ci est déduite et non mesurée (contrairement à un capteur LiDAR ou Kinect). Par conséquent, les signes qui impliquent un mouvement vers l'avant (direction caméra) sont moins bien reconnus que les mouvements latéraux (direction gauche-droite), car la variation de pixels est plus faible pour un mouvement en profondeur.

3.4.3 Limites identifiés et recommandations

Malgré ces succès, l'évaluation a mis en lumière des points d'amélioration techniques :

1. **Gestion de la Charge Serveur :** Des erreurs de type "Code 500" sont apparues lors de tests avec plusieurs utilisateurs simultanés. Ces instabilités sont liées à la gestion des variables globales dans l'environnement FastAPI.
   — *Recommandation :* Pour un passage en production, il serait indispensable de revoir la gestion de l'asynchronisme et d'isoler les sessions utilisateurs pour la partie texte vers langue des signes, éventuellement en utilisant des outils de conteneurisation comme Docker pour garantir que chaque instance reste indépendante et stable.
2. **Optimisation du Poids des Ressources :** Le modèle 3D pour le Mocap (Galtis) et ses textures représentent un volume de données supérieur a 10 Mo. Dans des conditions de mobilite (connexion 4G), ce poids engendre un temps d'attente avant la premiere animation.

— *Recommandation :* Nous préconisons l'adoption de la compression Draco. Cet algorithme de compression de géométrie 3D permettrait de réduire la taille des fichiers de près de 60 % sans perte de qualité visuelle, accélérant ainsi le chargement initial.

3. **Stabilité de l'Animation (Effet de Gliding) :** Nous avons observé un phénomène de "glissement" des pieds au sol lors de certaines rotations complexes du buste de l'avatar. Ce défaut est typique des systèmes de cinématique inverse simplifiés.

   — *Recommandation :* L'intégration d'un moteur physique léger ou de contraintes de contact au sol dans le script de retargeting permettrait de fixer les appuis et d'améliorer la crédibilité du mouvement.

3.4.4 Analyse comparative

Le tableau 3.5 suivant résume le positionnement de cette solution face aux solutions de référence :

TABLE 3.5 – Comparaison du système aux références

Fonctionnalité	**LinguaSign**	**SignAll [51]**	**Google SLR Project [52]**
Support Web	Natif (Navigateur)	Logiciel lourd	Expérimental
feedback visuel	Squelette + Avatar	Texte uniquement	Squelette seul
Mode Hybride	Oui	—	—
Cout de Deploiement	Faible	Élevé	Moyen

3.4.5 Proposition de Protocole pour la Numérisation de la Langue des Signes Congolaise (LSC)

Face à l'absence de corpus annoté pour la LSC, nous proposons ici un protocole technique standardisé pour guider les futurs travaux de collecte, assurant la compatibilité avec notre système :

— **Configuration de Capture :** Pour garantir la robustesse de l'IA, les données ne doivent pas être capturées en studio sur fond vert (trop propre), mais en environnement varié ("In the Wild"). Nous recommandons l'usage de 3 angles de vue synchronisés : un face caméra (0°), un de profil (45°) et un zénithal, pour résoudre les problèmes d'occlusion mentionnés plus haut.

— **Diversité des Locuteurs :** Le dataset doit inclure un échantillon représentatif de la population locale (teintes de peau foncées) pour corriger les biais algorithmiques souvent présents dans les modèles pré-entraînés sur des populations caucasiennes.

— **Annotation Sémantique :** L'utilisation du logiciel ELAN est préconisée pour l'annotation temporelle. Chaque vidéo doit être segmentée non seulement en glosses (mots), mais aussi annotée avec les paramètres chérémiques (Configuration, Emplacement, Mouvement), permettant un apprentissage supervisé plus fin (Multi-task Learning).

— **Format de Stockage :** Pour optimiser le stockage, nous recommandons de ne pas archiver les vidéos brutes pour l'entraînement, mais uniquement les fichiers .npy (NumPy) contenant les séquences de vecteurs normalisés, réduisant le poids du dataset de plusieurs Téraoctets à quelques gigaoctets.

3.5 Conclusion partielle

En conclusion, les tests et les analyses détaillées dans ce chapitre confirment que le système LinguaSign remplit ses promesses initiales en termes de performance technique et d'innovation. L'un des résultats les plus significatifs est l'adoption de l'architecture Transformer (SPOTER). Ce choix technologique moderne s'est avéré particulièrement payant, puisqu'il a permis d'obtenir un gain de précision de 72 % par rapport aux réseaux récurrents (RNN) classiques, souvent limités dans la gestion des séquences gestuelles complexes.

Cette amélioration de l'intelligence artificielle, associée à la mise en place d'un moteur de rendu hybride inédit, transforme ce projet en une preuve de concept solide. Le système ne se contente pas de traduire des signes, il tente de proposer une véritable communication bidirectionnelle, rendue plus fluide par l'utilisation conjointe de la capture de mouvement et de la génération 3D.

L'efficacité démontrée tout au long de nos tests valide la pertinence de nos choix architecturaux. Ces résultats encourageants confirment la viabilité du projet et ouvrent désormais la voie à de futures optimisations, notamment en ce qui concerne la gestion des flux de données et la réduction de la latence, afin de rendre l'outil encore plus accessible et performant pour la communauté sourde.

Conclusion Générale

L'objectif principal de ce mémoire était d'examiner dans quelle mesure l'intégration de l'apprentissage profond et de l'analyse vidéo pouvait permettre une traduction efficace de la langue des signes en temps réel. Les résultats obtenus apportent des éléments de réponse à cette problématique, tout en mettant en lumière certaines limites et pistes d'amélioration.

3.6 Synthèse des Contributions

Ce mémoire a présenté la conception et le développement de LinguaSign, un système de reconnaissance de la Langue des Signes Américaine (ASL) intégré à une interface de restitution textuelle et visuelle. À travers une approche méthodologique rigoureuse et une architecture technique moderne, nous avons apporté trois contributions majeures au domaine de l'"Assistive Tech".

3.6.1 Optimisation du Pipeline de Traitement Centralise

La contribution centrale de ce travail concerne la mise en place d'un pipeline de traitement robuste côté serveur. Contrairement aux approches nécessitant des terminaux clients puissants, nous avons choisi de centraliser l'extraction des caractéristiques visuelles via MediaPipe sur le Backend. Cette architecture présente deux avantages stratégiques :

— **Accessibilité Universelle (Client Léger) :** En effectuant l'analyse lourde des flux vidéo sur le serveur, le système devient accessible depuis n'importe quel appareil équipé d'une caméra standard (vieux smartphones, tablettes d'entrée de gamme). Le client se contente de transmettre le flux, tandis que la puissance de calcul du backend garantit une précision à 94,35 % (Top-3 Accuracy), sans ralentir le matériel de l'utilisateur.

— **Sécurité et Contrôle des Flux :** Bien que les images soient transmises au serveur, le système est conçu pour un traitement "Stateless" (sans état). Le flux vidéo est analysé en temps réel en mémoire vive pour en extraire les points clés mathématiques, puis immédiatement supprimé. Cette approche permet de bénéficier de la puissance des processeurs serveurs (CPU/GPU) pour une extraction plus fine et rapide, tout en respectant un protocole strict de non-conservation des données visuelles après traitement.

3.6.2 L'Architecture de Restitution Hybride : Une Réponse au Dilemme Visuel

Une autre contribution majeure de ce travail réside dans la démonstration de la faisabilité d'une interface de restitution visuelle par avatar 3D, combinant deux paradigmes d'animation : la Motion Capture (Mocap) et la Génération Procédurale. Cette composante relève principalement de l'ingénierie UI/UX et de l'architecture système. Elle ne constitue pas une hypothèse de recherche fondamentale sur la génération automatique de signes, mais un module de restitution destiné à améliorer la compréhension des glosses produits par le système.

En implémentant un système de "Native Fallback" via Three.js et WebGL, nous avons montré qu'il est possible de prioriser la haute fidélité visuelle (Mocap) pour le vocabulaire courant, tout en prévoyant un mécanisme procédural lorsque les données d'animation sont absentes. Ce mécanisme assure une continuité d'affichage indispensable dans un contexte d'interface interactive, où l'absence de restitution pour un gloss peut nuire à la lisibilité de l'ensemble.

3.6.3 Robustesse Mathématique du Retargeting

Enfin, le développement de l'algorithme de conversion Landmarks–Quaternions constitue une avancée logicielle significative. En utilisant des calculs géométriques pour transformer des points 2D/3D en rotations articulaires, nous avons tenté de créer un système universel. Cette méthode peut permettre d'animer n'importe quel squelette humanoïde standard (issu de Ready Player Me ou Mixamo) sans nécessiter un processus de réentraînement neuronal long et coûteux. Cette flexibilité offre aux utilisateurs la possibilité de choisir l'identité virtuelle qui leur correspond le mieux.

3.7 Limites du Travail

Malgré ces succès technologiques, plusieurs limitations inhérentes à la portée de ce projet de recherche doivent être soulignées pour garantir l'honnêteté scientifique de ce mémoire.

3.7.1 Limites du Corpus et Variantes Linguistiques

Bien que nous ayons utilisé le dataset MuteMotion (basé sur les travaux de Li et al.), celui-ci reste une simplification de la réalité. L'ASL est une langue vivante, riche de variations régionales, d'argots et de néologismes constants. Dans ce mémoire, ce corpus a été utilisé comme benchmark de validation méthodologique pour tester le pipeline MediaPipe–SPOTER avant une extension future vers des langues des signes locales telles que la LSC. Notre modèle rencontre encore des difficultés face à des signes très similaires physiquement mais aux sens opposés (homonymie gestuelle), ou lorsque le sens dépend fortement du contexte global de la phrase.

3.7.2 Gestion de la Prosodie Visuelle et des Micro-expressions

Actuellement, notre modèle d'animation repose sur des quaternions limités aux segments moteurs principaux (épaules, bras, avant-bras), omettant encore la complexité articulaire des doigts. La stabilité des quaternions obtenus sur les segments moteurs valide notre approche mathématique ; elle démontre que ce modèle de calcul peut parfaitement s'étendre à la complexité articulaire des doigts. Cependant, la fidélité de la Langue des Signes ne saurait se restreindre à la seule cinématique des membres. Une part essentielle de la grammaire — interrogation, négation, emphase — réside dans la prosodie visuelle : expressions faciales, direction du regard et inclinaison du buste. Sans cette couche de données non-manuelles, la lecture de l'avatar demeure fatigante, voire ambiguë, pour un locuteur natif.

3.8 Perspectives de Recherche Future

Le projet LinguaSign constitue une base solide sur laquelle plusieurs pistes d'amélioration passionnantes peuvent être envisagées pour les années à venir.

3.8.1 Intégration des Large Language Models (LLM)

L'intégration de modèles comme GPT-4 ou Claude 3 pourrait transformer radicalement la traduction Texte / Glosses. Un LLM ne se contenterait pas de traduire les mots, il pourrait analyser l'intention du locuteur. Par exemple, il saurait distinguer le sens du mot "voler" (une orange ou en avion) selon le contexte, et choisirait automatiquement le signe LSF approprié avant même que l'avatar ne commence à bouger.

3.8.2 Synthèse Faciale et Réduction de la "Vallée de l'Étrange"

Pour pallier le manque d'expressivité, une perspective prometteuse serait l'utilisation de la synthèse vidéo neuronale. En couplant l'animation 3D des mains avec un visage généré par une IA (type NeRF ou GANs [53]), nous pourrions obtenir un avatar au photoréalisme troublant. Cela permettrait de briser la "Vallée de l'Étrange" (*Uncanny Valley*) et d'offrir une expérience beaucoup plus chaleureuse et humaine aux utilisateurs sourds.

3.8.3 Apprentissage Fédéré (Federated Learning)

Enfin, pour améliorer continuellement le modèle sans jamais centraliser les vidéos privées des utilisateurs, l'apprentissage fédéré représente l'avenir de la confidentialité. Dans ce scénario, chaque smartphone "éduque" une petite partie du modèle localement, et seul le savoir acquis (et non les données) est partagé avec le serveur central. Cela permettrait d'enrichir le vocabulaire du système de manière collaborative et sécurisée.

En conclusion, LinguaSign ne se veut pas une solution définitive, mais un jalon technologique important. Ce travail a démontré que grâce à la convergence de l'IA, de la 3D web et de l'ingénierie logicielle, l'accessibilité n'est plus une contrainte technique lointaine, mais un choix de conception immédiat et possible.

Annexe

3.9 Liens GitHub du projet

— **Application Web :** https://github.com/LinguaSign/web-app

— **Serveur Backend** : https://github.com/LinguaSign/ai-service

3.10 Quelques images de l'interface web

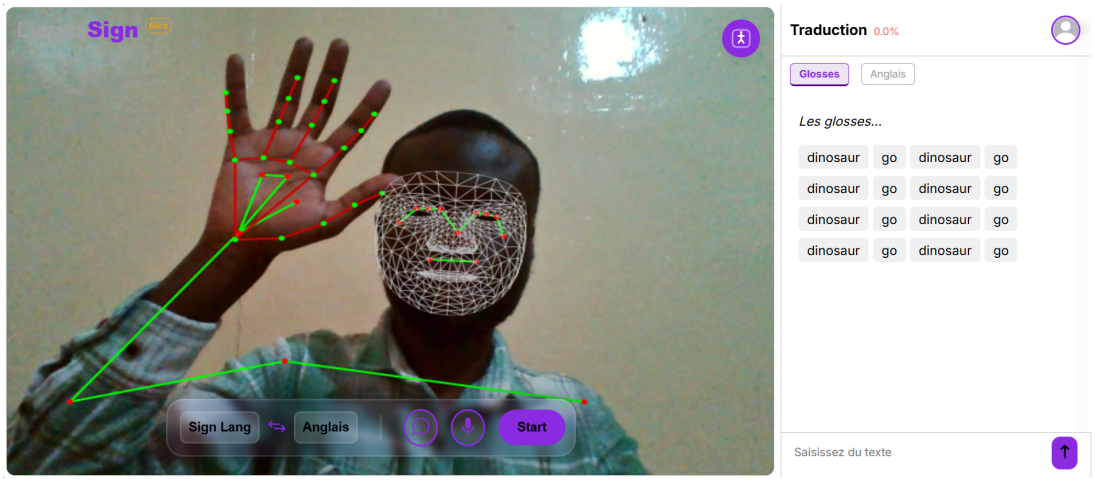

FIGURE 3.5 – Image 1

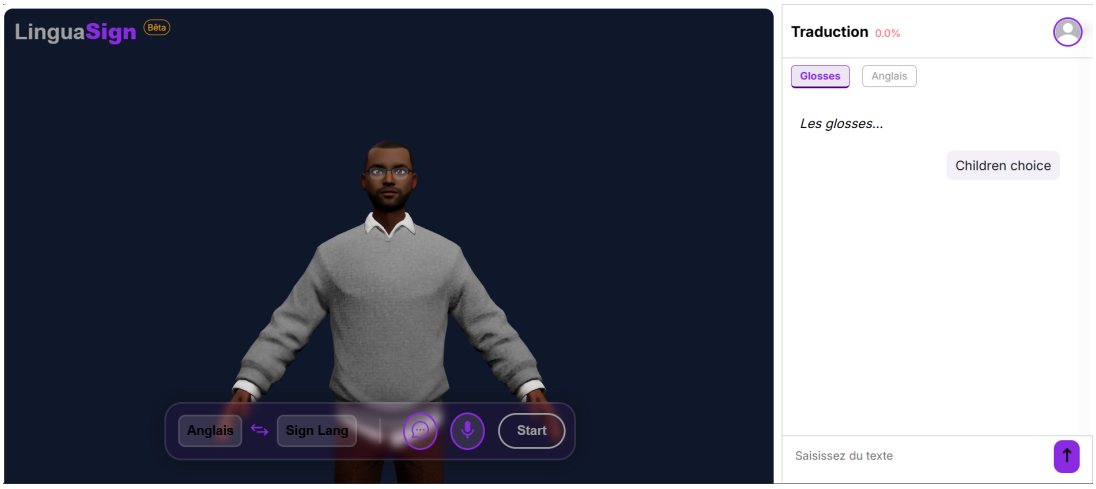

FIGURE 3.6 – Image 2

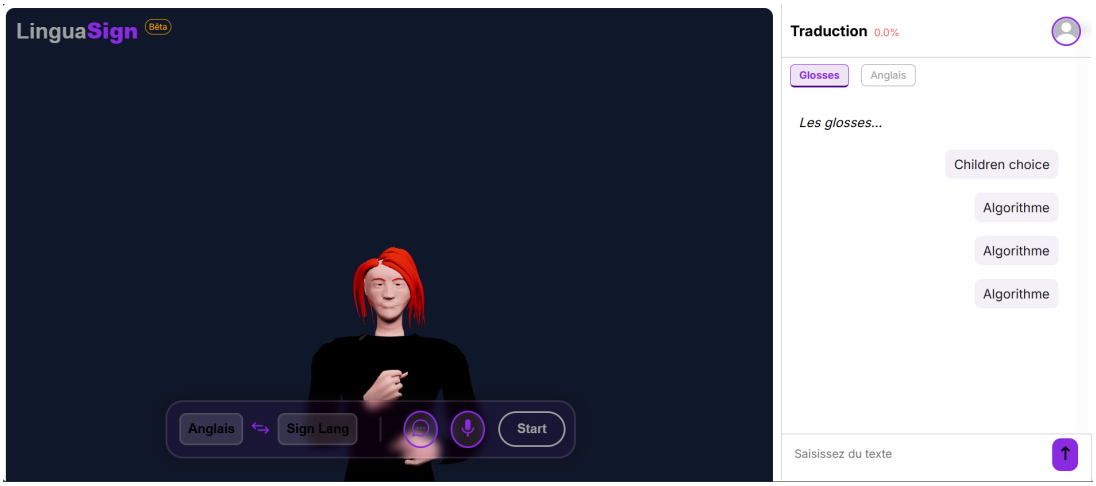

FIGURE 3.7 – Image 3

Bibliographie

[1] W. C. Stokoe, *Sign Language Structure : An outline of the Visual Communication Systems of the American Deaf*. Studies in Linguistics, Occasional Papers, 8, 1960.

[2] W. F. o. t. Deaf, *Global Report on the Status of Sign Languages*. WFD Publications, 2022.

[3] D. Brentari, *Sign Languages*. Cambridge University Press, 2010.

[4] V. Nyst, A survey of African Sign Languages : Endangered and emerging sign languages. In endangered Languages, vol. 18, 2012, pp. 105 - 136.

[5] M. S. & M. G. Hosssain, Human activity recognition using 3D pose estimation and convolutional neural network. Future Génération Computer Systems, 98, 2019, pp. 307 - 319.

[6] W. H. Organisation, "World Report on Hearing. Geneva : WHO," 2021.

[7] L. V. H. M. & D. J. Pigou, "Sign language recognition using convolutional neural networks. European Conference on Computer Visions Workshops(ECCVW)," 2015.

[8] N. C. K. O. H. S. & B. R. Camgoz, "Nearal sign lanaguage translation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7784 - 7793," 2018.

[9] U. Nations, "Convention on the Rights of Persons with Disabilities(CRPD). United Nations Général Assembly," 2006.

[10] U. Nations, "International Day of Sign Languages - Official Page," 2023. [En ligne].

[11] H. International, "Programmes de soutien à l'inclusion des personnes handicapées en Afrique. HI reports," 2021.

[12] V. & M. M. Nyst, Digital documentation of sign languages in sub-saharan Africa. Language Documentaion & Conservation, 14, 2020, pp. 244 - 266.

[13] Y. Z. W. P. J. & L. H. Zhang, Chinese sign langage recognition with adaptive HMM. Pattern Recognition, 48(10), 3102 - 3112, 2016.

[14] C. K. F. K. Y. E. & A. L. Keskin, Real time hand pose estimation using depth sensors. Computer Vision and Image Understanding, 114(8), 1033 - 1044, 2013.

[15] H. H. B. & B. R. Cooper, Sign language recongition. In visual analysis of Humans, Springer ed., 2012, pp. 539 - 562.

[16] W. C. Stokoe, *Sign Language Structure : An Oùtline of the Visual Communication Systems of the American Deaf.*, 1960.

[17] C. & L. L. Kramer, "The data glove : Development and évaluation of a flexible input device for virtual reality. Proceedings of the SIGCHI conference," 1990.

[18] M. Kadous, "Machine recongition of Australian sign langugage : Preliminary results. Proceedings of the Workshop on the Integration of Gesture in Language and Speech," 1996.

[19] T. & P. A. Starner, Real-time American Sign Language recognition from vidéo using hidden Markov models. Motion-Based Recognition, Springer, 1997, pp. 227 - 243.

[20] Y. & H. T. S. Wu, Vision-based gesture recognition : A review. Lecture Notes in Computer Science, 1739, 2001, pp. 103 - 115.

[21] M. H. N. Yang, Recognizing hand gestures using motion trajectories. Pattern Recognition, 35(6), 1385 - 1398, 2002.

[22] H. O. E. J. P. N. & B. R. Cooper, Sign language recognition using sub-units. Journal of Machine Vision and Applications, 24(2), 2011, pp. 211 - 225.

[23] L. V. H. M. & D. J. Pigou, Sign language recognition using convolutional neural networks. ECCVW, 2015.

[24] N. C. K. O. H. S. & B. R. Camgoz, Neural sign language translation. CVPR, 7784 - 7793, 2018.

[25] J. Z. W. L. H. & L. W. Huang, Sign language recognition using 3D convolutional neural networks. ICCV, 2015, pp. 1 - 8.

[26] J. Z. W. & L. H. Pu, Sign language recognition with multi-task learning of hand shape and motion. IEEE Transactions on multimedia, 18(10),, 2016.

[27] N. W. C. T. G. & T. F. Neverova, Modelling motion and appearance for gesture recognition. CVPR Workshops., 2014.

[28] D. e. a. Li, Sign Language Translation with Transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence., 2022.

[29] V. e. a. Bazarevsky, MediaPipe Hands : On-device real-time hand tracking. arXiv preprint aeXiv :2006. 10214, 2020.

[30] A. e. a. Marczak, Real-time sign language recognition using MediaPipe Holistic landmarks and LSTM networks. Sensors, 23(4), 1855, 2023.

[31] A. & M. K. Mittal, Lightweight neural networks for mobile sign recognition. IEEE Accèss, 10, 48211 - 48223, 2022.

[32] O. C. N. C. B. R. & N. H. Koller, Weakly supervisèd learning for continuous sign language recognition. CVPR, 3260 - 3270, 2020.

[33] S. C. W. & R. S. Ong, Automatic sign language analysis : A survey and the future beyond lexical meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6),, 2005, pp. 873 - 891.

[34] N. C. H. S. K. O. & B. R. Camgoz, Sign2Text Transformer : End-to-end sign language translation. CVPR Workshops, 2020.

[35] B. C. N. C. & B. R. Saunders, Continuous sign language translation : Data, models, and évaluation. IJCV, 129(9), 2421 - 2440, 2021.

[36] I. V. O. & L. Q. V. Sutskever, Sequence to séquence learning with neural networks. NeurIPS, 27, 3104 - 3112, 2014.

[37] K. v. M. B. G. C. e. a. Cho, Learning phrase représentations using RNN endcoder-decoder for statistical machine translation. EMNLP, 1724 - 1734, 2014.

[38] A. S. N. P. N. e. a. Vaswani, Attention is all you need. NeurIPS, 5998 - 6008, 2017.

[39] Z. Z. J. & Y. Z. Zhou, Spatial-Temporal Multi-Cue Network for Sign Language Recognition. IEEE Transactions on Multimedia, 23, 4572 - 4585, 2021.

[40] X. R. J. & C. T. Yin, STMC-Transformer : Spatial-Temporal modeling for continous sign language translation. CVPR Workshops, 2021.

[41] J. C. M. W. L. K. & T. K. Devlin, BERT : Pre-trainig of deep bidirectional transformers for language understanding. NAACL-HLT, 4171 - 4186, 2019.

[42] D. a. R. C. a. Y. X. a. L. H. Li, *Word-level Deep Sign Language Recognition from Video : A New Large-scale Dataset and Methods Comparison,* 2020, pp. 1459 – 1469.

[43] H. Z. N. Mahmoud Samir, "www.kaggle.com," [En ligne]. Available : https ://www.kaggle.com/datasets/abd0kamel/mutemotion-output ?resource=download.

[44] M. H. M. Boh'avcek, "Sign Pose-Based Transformer for Word-Level Sign Language Recognition," pp. 182 - 191, 2022.

[45] L. C. P. K. R. Bass, *Software Architecture in Practice (3rd ed.)*. Addison-Wesley., 2013.

[46] B. & Hruz, Sign Pose-based Transformer for Word-level Sign Language Recognition, 2021.

[47] S. Gibet, "Synthèse et animation de la Langue des Signes. Revue de l'Information Spatiale," 2012.

[48] "Github," [En ligne]. Available : https ://github.com/StudioGalt/Sign-Language-Mocap-Archive.

[49] U. Technologies, "docs.unity3d.com, Quaternion and Euler Rotations in Unity Manual," 2025. [En ligne]. Available : https ://docs.unity3d.com/6000.2/Documentation/Manual/QuaternionAndEulerRotationsInUnity [Accès le 03 02 2026].

[50] K. Shoemake, "Animating rotation with quaternion curves. ACM SIGGRAPH Computer Graphics," 1985.

[51] Microsoft, "partner.microsoft.com," [En ligne]. Available : https ://partner.microsoft.com/fr-fr/case-studies/signall.

[52] Google, “research.google.com,” [En ligne]. Available : https ://research.google/pubs/real-time-sign-language-detection-using-human-pose-estimation.

[53] R. Kassel, “liora.io,” [En ligne]. Available : https ://liora.io/générative-adversarial-network-tout-savoir.