الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبو بكر بلقايد - تلمسان
Université Abou Bekr Belkaid – Tlemcen

كلية العلوم - قسم الاعلام الالي
Faculté des sciences - Département d'Informatique

Mémoire

Dans le cadre du **décret 1275**
Présenté pour l'obtention du diplôme de **Master** en **Informatique**
Spécialité : **Intelligence Artificielle**

Sujet du Mémoire

Lunettes intelligentes pour la reconnaissance de la langue des signes

Présenté par :

- Khiat Siham
- Mahi Aya

Soutenu le 02 / 07 / 2025 devant le jury composé de :

- M. SMAHI Mohamed Ismail	Président
- Mme. AMRAOUI Asma	Encadrante
- M. SAIDI Abdessamad	Examinateur
- Mme. SELADJI Yassamine	Experte I2E

Année Universitaire : 2024 / 2025

Remerciements

Les plus sincères remerciements vont à **Madame Amraoui Asma**, encadrante de ce mémoire, pour sa disponibilité constante, sa bienveillance, et la pertinence de ses conseils. Toujours à l'écoute, elle a su guider chaque étape du projet avec rigueur et générosité. Son accompagnement attentif a grandement contribué à la qualité et à l'aboutissement de ce travail.

Nous remercions sincèrement **Monsieur Smahi Mohamed Ismail, Monsieur Saidi Abdessamad et Madame Seladji Yassamine** d'avoir accepté d'examiner notre travail et pour le temps qu'ils consacreront à l'étude de ce mémoire.

Nous adressons également nos remerciements à **l'ensemble des enseignants**, de l'école primaire jusqu'à l'université, pour leur accompagnement tout au long de notre parcours. Leur patience, leurs efforts et leur transmission du savoir ont joué un rôle essentiel dans l'accomplissement de ce travail.

Un remerciement chaleureux à **Monsieur Tedlaoui Mohammed**, pour son soutien précieux. Il nous a assurés qu'il mettrait à notre disposition les équipements nécessaires à la réalisation de notre projet. Son encouragement et son retour positif à propos de notre idée nous ont motivés et renforcés dans notre démarche.

Une profonde reconnaissance va à **l'équipe pédagogique de l'École pour Enfants Handicapés Auditifs Slimani Bouhafce**, pour l'accueil chaleureux et bienveillant réservé durant la période de stage. Cette immersion sur le terrain a permis de renforcer la dimension humaine et professionnelle du projet.

Une mention toute particulière à **Madame Souhila Ben Ahmed**, pour sa disponibilité, son accompagnement attentif et ses conseils éclairés tout au long de l'expérience au sein de l'établissement.

Enfin, nos remerciements vont à **la directrice de la DASS** pour son accueil, son autorisation et son soutien durant le stage, ainsi qu'à **Monsieur Melki** pour son accompagnement.

Dédicaces

À mon père et ma mère,
Merci d'avoir cru en moi, même dans les moments où moi, je doutais. Vos sacrifices, vos prières silencieuses et votre amour constant sont les fondations de tout ce que je suis aujourd'hui.

À mes deux sœurs Bahidja , Zahera et mon frère Mounir,
Toujours là, simplement. Dans les rires comme dans les silences, dans les regards et les petits gestes qui en disent long. Vous êtes une part essentielle de ma force.

À ma chère Adida,
Ta tendresse, ton attention et ta présence m'ont enveloppée de douceur tout au long de ce parcours. Ta place dans mon cœur est unique.

À mes précieuses amies : Amel, Hind, Rania, Amina, Nihel,
Votre amitié a été un vrai rayon de soleil. Vos mots, votre énergie et vos éclats de rire ont rendu ce chemin plus lumineux.

À Rayen, Yousra, Islem, Feryel, Youcef, Fatéma et Aya,
Vos encouragements et votre bienveillance m'ont apporté un vrai soutien.

Et enfin, à mon binôme Siham,
Merci pour la collaboration, la patience et l'engagement partagé. Cette aventure n'aurait pas eu la même saveur sans toi.

Aya Mahi

Dédicaces

À ma mère,
pilier de mon parcours et source de ma force, dont les prières m'ont accompagnée à chaque étape.

À mon père,
qui m'a soutenue, encouragée et fait confiance.

À mon frère Mohammed,
qui a toujours pris le temps de m'écouter et m'a offert un soutien moral précieux.

À mon frère Said et sa petite famille,
qui m'ont offert des instants de bonheur et ont illuminé mes journées.

À mes sœurs Samira, Atika et Hassiba,
qui, par leur amour, m'ont toujours porté vers le succès et le bonheur.

À tous mes neveux et nièces,
Chaque instant passé en votre compagnie est gravé dans mon cœur.

À mon binôme Aya,
partenaire de route, merci pour ta collaboration sincère et ta belle énergie.

Je suis infiniment reconnaissante de vous avoir dans ma vie.

C'est avec tout mon amour et ma gratitude que je consacre ce mémoire à vous.

Siham Khiat

Résumé

SignLens est un système innovant de lunettes intelligentes conçu pour la reconnaissance et la traduction en temps réel de la langue des signes, grâce à un modèle d'apprentissage profond embarqué. Équipé d'une caméra et d'un microcontrôleur Raspberry Pi, SignLens capture et analyse les gestes des mains à l'aide d'un modèle d'intelligence artificielle afin de les convertir en texte lisible ou en parole synthétisée. Cette solution portable et économique vise à faciliter l'accès à la communication des personnes sourdes et muettes dans les milieux éducatifs, médicaux et professionnels, favorisant ainsi l'inclusion sociale et l'autonomie.

Mots-clés : Reconnaissance de langue des signes, Traduction de la langue des signes, lunettes intelligentes, technologie d'assistance, apprentissage profond, réseau neuronal convolutionnel, Raspberry Pi, EfficientNet B0, EfficientNet V2S, MobileNet V2, Perceptron Multicouche, MLP
.

Abstract

SignLens is an innovative smart glasses system designed to enable real-time recognition and translation of sign language using embedded deep learning models. Integrating a camera and a Raspberry Pi microcontroller, SignLens captures and analyzes hand gestures through an artificial intelligence model to convert them into readable text or synthesized speech. This portable and cost-effective solution aims to improve communication accessibility for deaf and mute individuals across educational, medical, and professional environments, promoting social inclusion and autonomy.

Keywords : Sign language recognition,Sign language translation, smart glasses, assistive technology, deep learning, convolutional neural network ,Raspberry Pi, EfficientNet B0, EfficientNet V2S, MobileNet V2, Multilayer Perceptron , MLP .

الملخص

نظام SignLens هو نظارة ذكية مبتكرة تهدف إلى تمكين التعرف والترجمة الفورية للغة الإشارة باستخدام نماذج التعلم العميق المدمجة. يعتمد النظام على كاميرا متصلة بوحدة تحكم صغيرة من نوع Raspberry Pi ، حيث يلتقط الإيماءات اليدوية ويحللها عبر نموذج ذكاء اصطناعي لتحويلها إلى نص مقروء أو كلام مسموع. وقد صُمم هذا الحل المحمول ومنخفض التكلفة بهدف تحسين سُبل التواصل للأشخاص الصمّ و البكم في مختلف البيئات التعليمية والطبية والمهنية، مما يساهم في تعزيز اندماجهم الاجتماعي واستقلاليتهم.

الكلمات المفتاحية : التعرف على لغة الإشارة، ترجمة لغة الإشارة ، النظارات الذكية، التكنولوجيا المساعدة ، التعلم العميق، الشبكات العصبية، Raspberry Pi ، MobileNet V2S ، EffecientNet V2S ، EfficientNet B0 ، MLP ، Multilayer Perceptron .

Table des matières

Liste des figures

Liste des tableaux

Liste des équations

Liste des abréviations

1D	Une dimension
2D	Deux dimensions
3D	Trois dimensions (avec profondeur)
4K	Résolution d'image de 3840 × 2160 pixels
AMD	Advanced Micro Devices
ANN	Artificial Neural Network
AR	Augmented Reality
ASL	American Sign Language
AutoCAD	Automatic Computer-Aided Design
B2B	Business to Business
BGR	Blue Green Red
BOVW	Bag of Visual Words
BSL	British Sign Language
CAO	Conception Assistée par Ordinateur
CATIA	Computer-Aided Three-dimensional Interactive Application
CD-ROM	Compact Disc - Read Only Memory
CNN	Convolutional Neural Network
CSI	Camera Serial Interface
DA	Dinar Algérien
DL	Deep Learning
DPI	Dots Per Inch
DS-1	Dataset 1
DS-2	Dataset 2
DSI	Display Serial Interface

EOH	Edge Orientation Histogram
FCC	Federal Communications Commission
FPC	Flexible Printed Circuit
FLPOs	Floating Point Operations
fps	Frames Per Second
GAN	Generative Adversarial Network
GELAN	Generalized Efficient Layer Aggregation Network
GHz	Gigahertz
Go	Gigaoctet
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HDMI	High-Definition Multimedia Interface
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HUD	Head-Up Display
I2C	Inter-Integrated Circuit
IA	Intelligence Artificielle
IDE	Integrated Development Environment
IoT	Internet of Things
IR	Infrarouge
ISL	Indian Sign Language
JPS	JPEG Stereo
KSL	Kurdish Sign Language
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LSTM	Long Short-Term Memory
LPDDR4	Low Power Double Data Rate version 4
MBConv	Mobile Inverted Bottleneck Convolution

microSD	Micro Secure Digital
MIPI	Mobile Industry Processor Interface
MnasNet	Mobile Neural Architecture Search Network
MLP	Multi-Layer Perceptron
MP	Mégapixels
NLP	Natural Language Processing
ONG	Organisation Non Gouvernementale.
PDAF	Phase Detection Auto Focus
PGI	Programmable Gradient Information
RAM	Random Access Memory
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
ResNet	Residual Network
RGB	Red Green Blue
RoHS	Restriction of Hazardous Substances
R&D	Recherche et Développement
RV	Réalité virtuelle
SSD	Solid State Drive
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
TL	Transfer Learning
USB	Universal Serial Bus
VGG	Visual Geometry Group
VS Code	Visual Studio Code
VNC	Virtual Network Computing
Wi-Fi	Wireless Fidelity
YOLO	You Only Look Once
°C	Degré Celsius

Introduction générale

Dans un monde où la communication est au cœur de chaque interaction sociale, il reste malheureusement encore des barrières importantes pour certaines communautés, notamment les personnes sourdes et muettes. Ces dernières rencontrent quotidiennement des difficultés à se faire comprendre et à interagir efficacement avec leur entourage, en raison du manque d'outils technologiques accessibles et adaptés à leur langage principal : la langue des signes.

Malgré les avancées dans le domaine de l'intelligence artificielle, de la vision par ordinateur et des objets connectés, peu de solutions complètes et abordables ont vu le jour pour traduire en temps réel la langue des signes en texte ou en voix. Les dispositifs existants sont souvent complexes, coûteux, ou encore difficilement transportables, rendant leur usage limité dans la vie quotidienne. Ce constat met en évidence un besoin urgent d'outils intelligents, intuitifs et pratiques, capables d'agir comme passerelles entre les mondes auditif et visuel.

C'est dans ce contexte qu'intervient notre projet : **SignLens**, une paire de lunettes intelligentes conçue pour reconnaître la langue des signes à l'aide d'un système embarqué reposant sur un modèle d'apprentissage profond. Équipées d'une caméra et d'un microcontrôleur tel que le *Raspberry Pi*, ces lunettes permettent de capturer les mouvements des mains et de les interpréter en temps réel grâce à un modèle d'intelligence artificielle. Le but étant de convertir ces gestes en texte, voire en son, facilitant ainsi la communication avec des personnes ne maîtrisant pas la langue des signes.

Le projet **SignLens** ambitionne de proposer une solution portable, économique et inclusive, adaptée aux contextes éducatifs, médicaux et professionnels, tout en valorisant l'autonomie des utilisateurs sourds ou muets. Il s'adresse également aux institutions désireuses d'intégrer davantage de technologies inclusives dans leurs services.

Ce mémoire s'articule autour de quatre chapitres principaux. Le premier chapitre est dédié à l'étude de l'existant, incluant les dispositifs similaires et les recherches scientifiques liées à la reconnaissance de la langue des signes. Le deuxième chapitre présente les concepts théoriques fondamentaux, ainsi que les outils matériels et logiciels utilisés dans le développement de notre prototype. Le troisième chapitre expose notre contribution personnelle, les étapes de mise en œuvre du système, les tests réalisés et les résultats obtenus. Enfin, le quatrième chapitre est consacré à l'expérimentation du prototype sur le terrain, plus précisément à l'École pour Enfants Handicapés Auditifs, afin d'évaluer son efficacité en conditions réelles.

Chapitre 1

Étude de l’existant

1.1 Introduction

Dans ce chapitre, on a fait une étude de l'existant, qui constitue une phase fondamentale dans la réalisation d'un projet de recherche ou de développement technologique. Elle a permis d'identifier des dispositifs, ainsi que des approches scientifiques décrites dans la littérature, afin d'examiner les performances et de mieux cerner les attentes des utilisateurs. Cette approche a offert la possibilité de vérifier la présence de technologies similaires sur le marché, d'examiner leurs avantages et inconvénients, et d'identifier les fonctionnalités principales de ces dispositifs. Dans le contexte des lunettes intelligentes dédiées à la traduction de la langue des signes, cette étude s'est révélée particulièrement importante pour comparer les solutions disponibles et juger de leur efficacité. Elle a ainsi servi à valider la pertinence du projet en s'appuyant sur une analyse objective de l'existant et des références concrètes.

1.2 Dispositifs existants

Avant de se concentrer sur les lunettes intelligentes spécifiquement destinées à la traduction de la langue des signes, il est important d'abord de comprendre les lunettes intelligentes en général afin de mieux comprendre leur utilisation et les applications dans divers domaines.

1.2.1 Lunettes intelligentes

Les lunettes intelligentes sont des accessoires technologiques qui intègrent des fonctionnalités avancées pour enrichir l'expérience utilisateur. Elles embarquent divers composants en fonction des besoins et des usages auxquels elles sont destinées. Au fil des années, plusieurs entreprises ont tenté de concevoir et de commercialiser ces dispositifs, avec des degrés de succès variables. Certaines innovations n'ont pas su convaincre le grand public, en raison de leur coût élevé, de leurs performances limitées ou des inquiétudes liées à la protection des données personnelles. Tandis que certaines marques ont abandonné leurs projets après des échecs, d'autres ont su innover et améliorer progressivement leurs modèles. Voici quelques modèles qui ont réussi à se faire une place sur le marché :

1.2.1.1 Ray-Ban Meta

Les Ray-Ban Meta[1], lancées le 17 octobre 2023[2], constituent la seconde génération de lunettes intelligentes de Meta. Succédant aux Ray-Ban Stories de 2021, elles résultent d'une collaboration entre Meta (anciennement Facebook) et Ray-Ban, avec un prix de départ fixé à 329 euros. Comme le montre la figure 1.1, ces lunettes conservent l'apparence classique de Ray-Ban tout en intégrant des technologies avancées, notamment une caméra de 12 MP, des haut-parleurs, un assistant vocal activé par la commande "Hey Meta", ainsi que des commandes tactiles. Elles disposent également d'un stockage interne, d'une connectivité Bluetooth et Wi-Fi, et permettent de diffuser des vidéos en direct. Grâce à ces fonctionnalités, elles offrent diverses utilisations, telles que la capture de photos et de vidéos en temps réel, ainsi que la possibilité de passer des appels sans nécessiter de téléphone.

Fig. 1.1 : Ray-Ban Meta [1]

1.2.1.2 Vuzix Blade 2

Les Vuzix Blade ont été conçues par la société américaine Vuzix, spécialisée dans les technologies de vision assistée et de réalité augmentée. Reconnue pour ses avancées dans le domaine des lunettes intelligentes et des dispositifs portables, l'entreprise a continuellement amélioré ses produits afin d'enrichir leurs fonctionnalités et leurs performances.

La version la plus récente, le Vuzix Blade 2[3], a été lancée en 2022 et est actuellement commercialisée au prix de 1 380 euros (hors taxes)[4]. Cette nouvelle édition illustrée à la figure 1.2, apporte plusieurs améliorations par rapport à son prédécesseur, notamment une connectivité optimisée grâce à la prise en charge du Wi-Fi et du Bluetooth, facilitant ainsi l'interaction avec d'autres appareils et assurant un accès rapide aux réseaux. Par ailleurs, la capacité de stockage a été portée à 40 Go, permettant une utilisation plus étendue pour des applications aussi bien professionnelles que personnelles. L'affichage a été amélioré et la caméra de 8 MP offre une meilleure résolution, optimisant ainsi les performances en réalité augmentée. De plus, l'appareil est doté d'un processeur plus performant et fonctionne sous Android 11, assurant ainsi une compatibilité avec un large choix d'applications.

Fig. 1.2 : Vuzix Blade 2 [3]

1.2.2 Dispositifs de traduction de la langue des signes

1.2.2.1 Gant BrightSign

Le BrightSign[5], présenté à la figure 1.3, est un gant intelligent conçu pour faciliter la communication des personnes sourdes ou non verbales. Équipé de capteurs avancés, il capte les

mouvements des mains et des doigts afin de les convertir en texte ou en parole grâce à une application mobile dédiée. Offrant un large éventail de personnalisation, il prend en charge plus de 90 langues et propose plus de 900 voix et accents différents. De plus, l'utilisateur peut entraîner le dispositif à reconnaître de nouveaux signes, garantissant ainsi une grande adaptabilité. Disponible en plusieurs tailles (Large et Small/Medium) pour les droitiers comme pour les gauchers, il est fourni avec un kit de chargement USB. Son prix est fixé à 3100 dollars [6].

Fig. 1.3 : Gant BrightSign [5]

1.2.2.2 MyVoice

"MyVoice"[7] est un prototype qui a été baptisé en 2012 par des étudiants de l'Université de Houston pour interpréter la langue des signes en paroles audibles et vice versa. Ce dispositif portable est équipé d'un microphone, d'un haut-parleur, d'une caméra et d'un écran,comme illustré à la figure 1.4 . La caméra enregistre les gestes des mains de l'utilisateur, permettant ainsi la conversion des signes en son. À l'inverse, l'appareil peut également transcrire des paroles en langage des signes, qui s'affichent sur son écran.

Fig. 1.4 : dispositif MyVoice [7]

1.2.2.3 Traducteur de langue des signes basé sur Kinect

En 2013, Microsoft Research Asia, en collaboration avec l'Académie chinoise des sciences et l'Université de Pékin, a développé le prototype Kinect Sign Language Translator[8], présenté à la figure 1.5 . Ce système utilise la caméra Kinect de la Xbox pour capturer et analyser les gestes

des mains et du corps d'un utilisateur s'exprimant en langue des signes. Grâce à des algorithmes avancés de reconnaissance gestuelle, il traduit ces mouvements en texte ou en parole en temps réel. De plus, il est capable de convertir des paroles en représentations visuelles de la langue des signes, facilitant ainsi l'échange entre personnes sourdes et entendantes.

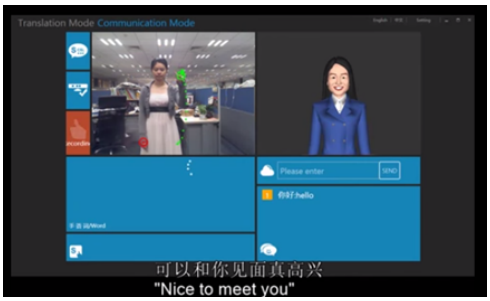

Fig. 1.5 : Traducteur de langue des signes basé sur Kinect [8]

1.2.2.4 Gant intelligent qui traduit la langue des signes en mots écrits et parlés

Une équipe de jeunes universitaires de Constantine a développé un gant intelligent [9] capable de traduire la langue des signes en mots écrits et parlés. Ce dispositif utilise des capteurs pour détecter les mouvements des mains et des doigts, puis les transforme en texte affiché sur un écran ou en voix synthétique, grâce à l'intelligence artificielle. Ce projet, conçu dans un cadre universitaire, vise à faciliter la communication des personnes sourdes et muettes avec leur entourage et à favoriser leur inclusion sociale.

1.2.2.5 Lunettes capables de traduire la langue des signes

D'après PatentScope [10], ce sont des lunettes conçues pour convertir la langue des signes en paroles audibles. Elles intègrent des caméras, des modules de reconnaissance gestuelle, des systèmes de synthèse vocale et une batterie lithium assurant leur alimentation. Les caméras qui sont placées près des lentilles, capturent les mouvements des mains et envoient ces informations aux modules de traduction. Ceux-ci convertissent les gestes en parole, qui est ensuite diffusée par les haut-parleurs (voir la figure 1.6).

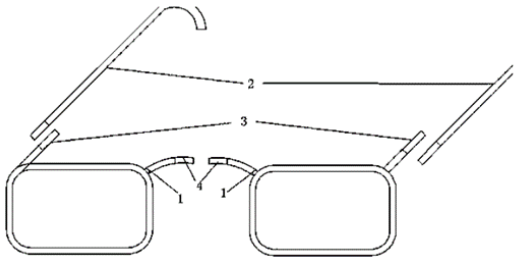

Fig. 1.6 : Lunettes capables de traduire la langue des signes [10]

1.2.2.6 Dispositif de lunettes de reconnaissance de la langue des signes

Un autre brevet a été obtenu pour ces lunettes intelligentes de reconnaissance de la langue des signes [11], dont une illustration est présentée à la figure 1.7. Elles intègrent des fonctionnalités

avancées pour une interaction optimale. Un collecteur de mouvements capte les gestes effectués, lesquels sont transformés en texte et en messages audio via un module vocal. Elles proposent aussi une interaction vocale permettant à l'utilisateur de sélectionner des options à l'aide de commandes vocales quand elles ne sont pas utilisées, par exemple l'option de projeter des vidéos ou écouter de la musique sur les lentilles HUD grâce au contrôle vocal. Alimentées par une batterie lithium intégrée, elles garantissent une portabilité accrue.

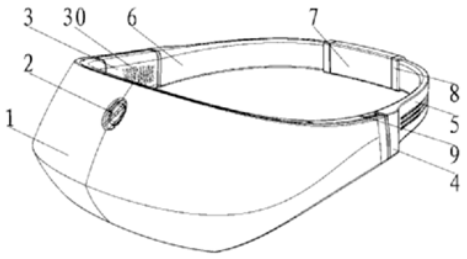

Fig. 1.7 : Dispositif de lunettes de reconnaissance de la langue des signes [11]

1.2.2.7 Lunettes de communication spéciales pour sourds-muets

Ces lunettes [12], représentées à la figure 1.8, sont conçues pour faciliter la communication entre les personnes sourdes-muettes et celles ne maîtrisant pas la langue des signes. Dotées d'un écran transparent, elles affichent des traductions en langue des signes ainsi que textuelles, rendant la communication plus claire. Lorsqu'une personne entendante parle, un module capte la voix, qui est ensuite analysée par un système de reconnaissance vocale avant d'être transformée en signes et en texte sur l'écran. Lorsqu'une personne sourde utilise la langue des signes, un enregistreur vidéo capture ces gestes, les interprète, puis les convertit en texte et en audio diffusés par un haut-parleur. L'alimentation du dispositif est assurée par un panneau photovoltaïque, garantissant une autonomie optimale, notamment en extérieur. En résumé, cette technologie permet une interaction fluide et bidirectionnelle entre les deux groupes, tout en exploitant une source d'énergie renouvelable .

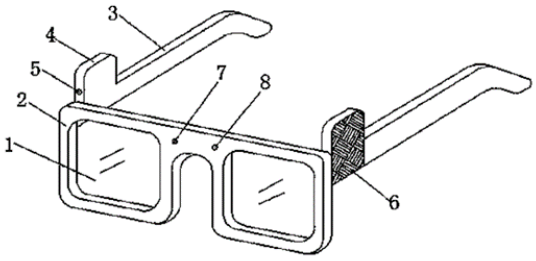

Fig. 1.8 : Lunettes de communication spéciales pour sourds-muets [12]

1.2.2.8 Machine de traduction portable

La figure 1.9 présente une machine de traduction portable brevetée en 2019 [13], sous la forme de lunettes intelligentes conçues pour interpréter les gestes de la langue des signes en texte ou en parole. Équipées de caméras de détection de profondeur 3D, elles intègrent également divers composants comme des boutons de volume et un port de charge. Elles utilisent des

technologies avancées, notamment l'imagerie laser et la mesure de lumière structurée. Une application mobile associée permet une sortie vocale en temps réel, facilitant la communication entre les personnes sourdes ou muettes et celles ne maîtrisant pas la langue des signes. L'objectif principal est d'assurer une traduction fluide et en temps réel.

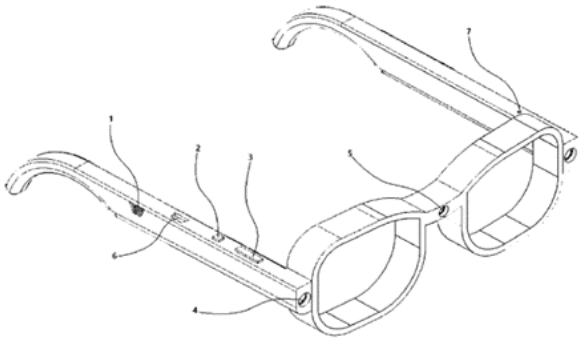

Fig. 1.9 : Machine de traduction portable [13]

1.2.2.9 Dispositif et méthode d'interprétation de la langue des signes en temps réel utilisant des lunettes AR

La figure 1.10 met en évidence le dispositif breveté en 2023, basé sur des lunettes de réalité augmentée (AR) permettant l'interprétation de la langue des signes en temps réel [14]. Il est capable d'analyser à la fois les gestes des mains et les expressions faciales de l'utilisateur afin d'assurer une meilleure traduction. Il est composé de plusieurs modules : un pour la reconnaissance des mouvements des mains et un autre pour l'analyse des expressions du visage, un module dédié à la traduction des gestes, ainsi qu'une unité de sortie audio permettant de restituer la traduction sous forme vocale. Par ailleurs, le système peut aussi identifier les gestes et expressions d'un interlocuteur et adapter la traduction en fonction de caractéristiques comme l'âge ou le sexe des personnes impliquées. En complément, il intègre une fonctionnalité de conversion vocale : il capte la voix de l'interlocuteur, la transforme en texte, puis la traduit en langue des signes. Son objectif principal est de faciliter les échanges entre les personnes qui maîtrisent et utilisent la langue des signes et celles qui ne la maîtrisent pas.

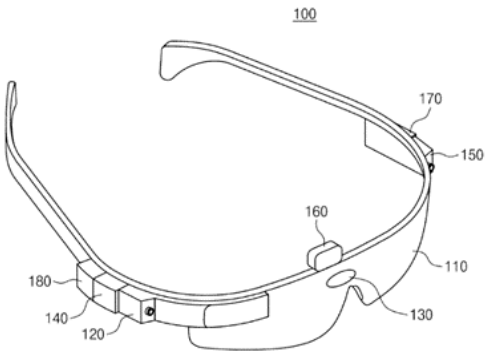

Fig. 1.10 : Dispositif et méthode d'interprétation de la langue des signes en temps réel utilisant des lunettes AR [14]

1.2.2.10 Système de traduction de la langue des signes

Ce système de traduction de la langue des signes [15] repose sur l'utilisation d'un bracelet intelligent porté par un premier utilisateur et d'un casque porté par un second utilisateur. Le bracelet est équipé de plusieurs composants intégrés dans son corps, notamment un capteur électromyographique placé sur sa face interne, un accéléromètre, un gyroscope, un module de communication réseau, un processeur principal ainsi qu'un module d'alimentation fournissant l'énergie nécessaire à son fonctionnement. De son côté, le casque embarque un module réseau, un processeur secondaire, un haut-parleur, un microphone et une source d'alimentation dédiée.

Conçu pour être simple d'utilisation, ce système garantit des traductions précises et en temps réel. Grâce à la connectivité Internet, il permet d'associer plusieurs bracelets et casques, assurant ainsi une transmission instantanée des gestes en langage des signes. Cette technologie favorise la communication fluide et simultanée entre plusieurs utilisateurs, rendant les échanges plus accessibles aux personnes sourdes et malentendantes.

1.2.2.11 Dispositif de reconnaissance de la langue des signes

Un système de reconnaissance de la langue des signes [16] est équipé d'un processeur et d'une mémoire a aussi obtenu un brevet en 2020. La mémoire contient des instructions permettant au processeur d'analyser les gestes en détectant la position, la forme et l'orientation des doigts à l'aide d'une caméra. En complément, le dispositif tient compte de la couleur du gant utilisé et exploite ces données visuelles, ainsi que les informations fournies par un capteur de flexion, pour identifier et interpréter les signes avec précision. La figure 1.11 donne un aperçu visuel du dispositif :

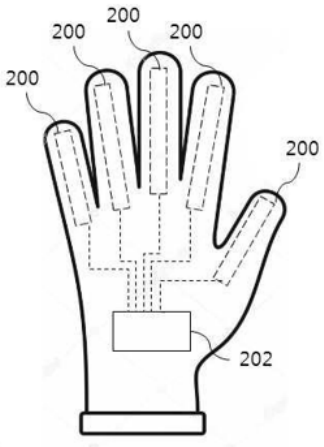

Fig. 1.11 : Dispositif de reconnaissance de la langue des signes [16]

1.2.2.12 Lunettes de détection intelligente

Ce dispositif portable [17], présenté à la figure 1.12, se compose principalement d'un écran fixé sur la tête, d'un système de caméra orienté vers l'extérieur et d'un processeur intégré. Il est conçu pour analyser et traduire les gestes de la langue des signes en temps réel. Les caméras grand-angle capturent l'environnement et détectent les mouvements des mains, tandis que le processeur exploite des réseaux neuronaux pour identifier et interpréter les gestes enregistrés.

Une fois les signes reconnus, le système les convertit en une autre langue, qu'elle soit parlée ou signée. La traduction peut être affichée sous forme de texte, diffusée sous forme audio ou représentée visuellement pour une autre langue des signes, selon les besoins de l'utilisateur. De plus, le système ajuste automatiquement la langue de traduction en fonction du contexte, comme la localisation ou d'autres paramètres personnalisés, afin d'optimiser l'expérience utilisateur

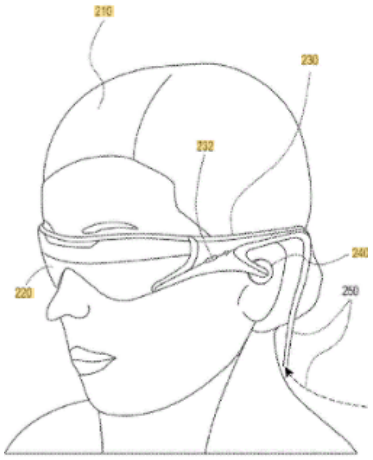

Fig. 1.12 : Lunettes de détection intelligente [17]

1.2.2.13 Appareil d'interprétation des mouvements de la main et de communication

Ce dispositif [18], montré à la figure 1.13, permet de convertir en temps réel les gestes de la langue des signes en mots et phrases, rendant la communication plus accessible. Il utilise huit capteurs flexibles placés sur les doigts, ainsi qu'un accéléromètre et une unité de mesure inertielle pour analyser les mouvements. Ces capteurs génèrent des signaux électroniques transmis à une unité de traitement chargée de les interpréter.

Les gestes sont ensuite comparés à une base de données interne afin d'identifier leur signification. Une fois reconnus, ils peuvent être affichés sous forme de texte ou restitués sous forme audio grâce à un synthétiseur vocal. Léger et ergonomique, cet appareil s'adapte aux mouvements naturels de la main, du poignet et des doigts, offrant une grande liberté d'utilisation.

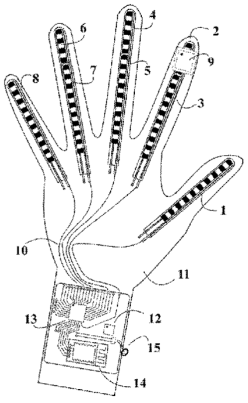

Fig. 1.13 : Appareil d'interprétation des mouvements de la main et de communication [18]

1.2.3 Tableau récapitulatif

Nom	Caractéristiques principales	Points forts	Points faibles
Ray-Ban Meta	Caméra 12 MP, haut-parleurs, assistant vocal, Bluetooth/Wi-Fi.	Design discret, commandes tactiles intuitives.	Qualité sonore limitée, dépendance au smartphone, confidentialité.
Vuzix Blade 2	Android 11, caméra 8 MP, écran Waveguide, 40 Go de stockage, Bluetooth/Wi-Fi.	Compatible avec Teams, bonne visibilité en plein soleil.	Prix élevé, compatibilité limitée avec certaines applications.
Gant BrightSign	Gant avec capteurs, traduit en texte/parole, 90 langues, 900 voix/accents.	Traduction multilingue, adaptable aux besoins.	Coût élevé, dépendance au smartphone, peut être inconfortable.
MyVoice	Prototype portable, microphone, haut-parleur, caméra, écran.	Communication bidirectionnelle, facile à transporter.	Détection gestuelle limitée.
Traducteur de langue des signes basé sur Kinect	Caméra, traduction en temps réel, reconnaissance gestuelle.	Bidirectionnelle, sans capteur physique sur l'utilisateur, traduction en texte et parole.	Dépend de l'éclairage, limitation des langues supportées.
Gant intelligent qui traduit la langue des signes en mots écrits et parlés	Gant avec capteurs de mouvement, traduction en texte ou voix synthétique.	Projet universitaire inclusif, facilite la communication et l'intégration sociale.	Prototype académique, précision et robustesse non encore validées à grande échelle.
Lunettes capables de traduire la langue des signes	Caméras, module de traduction, module vocal, batterie lithium.	Design portable, améliore la communication.	Pas d'affichage textuel, limitation de discrétion.

Tab. 1.1 : Tableau récapitulatif des dispositifs de reconnaissance et traduction de la langue des signes

Nom	**Caractéristiques principales**	**Points forts**	**Points faibles**
Dispositif de lunettes de reconnaissance de la langue des signes	Capture et conversion des gestes en texte/voix, HUD interactif.	Traduction en texte et parole, interaction vocale.	Inconfort avec utilisation prolongée de HUD, fonctionnalités complexes.
Lunettes de communication spéciales pour sourds-muets	Affichage transparent, reconnaissance vocale, alimentation par énergie solaire.	Communication fluide, autonomie énergétique.	Problèmes de confidentialité avec l'enregistrement vidéo.
Machine de traduction portable	Caméras 3D, imagerie laser, traduction texte/parole.	Traduction rapide et fluide.	Dépendance au smartphone.
Dispositif et méthode d'interprétation de la langue des signes en temps réel utilisant des lunettes AR	Reconnaissance des mains et expressions faciales, traduction vocale et gestuelle.	Précision améliorée, traduction bidirectionnelle, adaptation de voix selon plusieurs caractéristiques.	Sensible à l'éclairage et aux mouvements rapides, difficulté de traduction des gestes et expressions.
Système de traduction de la langue des signes	Bracelet intelligent + casque connecté, traduction en temps réel.	Traduction fluide et bidirectionnelle, connexion multi-utilisateurs.	Dépendance à Internet et au smartphone.
Dispositif de reconnaissance de la langue des signes	Caméra + capteurs de flexion, analyse des gestes et couleurs des doigts.	Précision améliorée grâce aux couleurs distinctes et capteurs.	Dépend de l'angle de la caméra et de la visibilité des doigts.
Lunettes de détection intelligente	Affichage tête haute, caméra grand angle, IA pour la reconnaissance gestuelle.	Traduction textuelle, audio et graphique, détection de langue cible selon des contextes.	Limites dans les environnements multilingues, limitation des langues supportées.
Appareil d'interprétation des mouvements de la main et de communication	Appareil avec capteurs flexibles et accéléromètre, synthèse vocale.	Filtrage avancé des gestes, portable et sans fil, traduction en texte ou son.	Dépendance aux capteurs, latence possible.

Tab. 1.1 : Tableau récapitulatif des dispositifs de reconnaissance et traduction de la langue des signes

1.3 Etat de l'art

Cette section présente un état de l'art des travaux de recherche existants dans le domaine de la reconnaissance de la langue des signes, en mettant en lumière les différentes approches, techniques et architectures utilisées dans la littérature scientifique.

Orovwode et al.[19] ont proposé un système de reconnaissance de la langue des signes en temps réel, basé sur un réseau de neurones convolutifs (CNN). Son architecture comprend trois couches convolutionnelles suivies de Max Pooling pour réduire la dimension des données, puis d'un passage par des couches entièrement connectées. La classification est assurée par une couche Softmax de 24 neurones. Le modèle utilise l'optimisation Adam et l'entropie croisée catégorielle comme fonction de perte.

Le jeu de données utilisé est constitué de 44 654 images représentant l'alphabet en langue des signes américaine, à l'exception des lettres J et Z. Ces images ont été capturées via une webcam et traitées avec le module `HandDetector` de `CvZone`. Les images, redimensionnées en 224×224 pixels, ont été converties en vecteur one-hot via `to_categorical`, normalisées, puis réparties en trois sous-ensembles : 20 772 images (70 %) ont été dédiées à l'entraînement, 8 903 à la validation et le reste pour le test.

Après cinq époques d'apprentissage avec des lots de 64 images, le modèle a atteint une précision de 99,86 % en entraînement, 99,94 % en validation et 94,68 % en test, démontrant son efficacité dans la reconnaissance des signes.

Selon **Karim et al.**[20] , la reconnaissance de la langue des signes kurde (KSL) repose sur l'utilisation de réseaux de neurones convolutifs (CNN), intégrant des outils avancés tels que `MediaPipe` et `TensorFlow` afin d'améliorer la précision de la détection.

Le jeu de données initial comprenait 66 000 images représentant 43 classes, les 33 lettres de l'alphabet kurde et les chiffres de 0 à 9, le nombre d'images par classe varie entre 700 et 1 500. Pour renforcer la diversité des exemples et améliorer la robustesse du modèle, des techniques d'augmentation de données telles que la rotation ont été appliquées pour avoir un nombre total d'images égal à 132 000. Chaque image a été redimensionnée à 256×256 pixels.

Le jeu de données a été divisé en deux ensembles : 105 600 images ont été utilisées pour l'entraînement du modèle, tandis que 26 400 images ont servi à l'évaluation.

Le modèle repose sur une architecture CNN intégrant plusieurs couches de convolution, des couches de pooling et des couches entièrement connectées. L'entraînement est optimisé grâce à l'algorithme `Adam`, avec une fonction de perte basée sur l'entropie croisée catégorielle.

Après plusieurs ajustements des hyperparamètres, une taille de lot de 32 a été sélectionnée pour optimiser à la fois l'utilisation des ressources mémoire et la performance de l'apprentissage. Le modèle a été entraîné sur 50 époques, en appliquant une stratégie d'arrêt automatique (*early stopping*) pour éviter que le modèle ne sur-apprenne. Le modèle a atteint une précision de 99,87 %, démontrant ainsi son efficacité dans la reconnaissance des signes de la langue kurde.

Selon **Paul et al.** [21], un système de reconnaissance de la langue des signes en temps réel combine `CNNet` `LSTM` pour interpréter et traduire les signes en texte ou en parole. Il utilise `MediaPipe` pour extraire les points clés des mains, du visage et de la posture, et les images capturées via `DroidCam` sont redimensionnées (50×50 pixels) et segmentées pour l'entraînement et

le test. LSTM et GRU analysent les gestes dynamiques sous forme de tableaux 2D, optimisés avec Adam et Dropout (0.5), tandis que CNN basé sur ResNet50 est utilisé pour reconnaître les lettres. Deux ensembles de données sont exploités : DS-1 (gestes : "hello", "thanks", "iamhungry") et DS-2 (28 600 images des 26 lettres de l'alphabet).

Les performances montrent que LSTM est le plus efficace avec une *accuracy* de 94.3 % pour les gestes dynamiques, suivi de CNN avec 89.07 % pour les lettres, tandis que GRU est moins performant avec 79.3 %. LSTM offre un meilleur *recall* et un score *F1* plus élevé, garantissant une meilleure fiabilité. L'optimisation avec Adam permet de réduire les erreurs et de stabiliser le modèle, assurant une reconnaissance fluide et précise en temps réel.

Katoch et al.[22], ont proposé une approche pour la reconnaissance de la langue des signes indienne (ISL) intégrant la vision par ordinateur et l'apprentissage automatique. Le système repose sur l'extraction des caractéristiques des images de signes en utilisant l'algorithme SURF (Speeded-Up Robust Features), qui permet d'identifier des points clés robustes aux variations de rotation et d'éclairage. Ces caractéristiques sont ensuite transformées en un *Bag of Visual Words* (BOVW), où les points d'intérêt sont regroupés en clusters à l'aide de Mini-Batch K-Means.

Pour la classification, deux modèles ont été évalués : SVM (Support Vector Machine), utilisant un noyau linéaire pour discriminer les signes à partir des histogrammes BOVW, et CNN (Convolutional Neural Network), exploitant une architecture de plusieurs couches convolutionnelles suivies de couches de pooling et de *fully connected*.

L'expérimentation a été réalisée sur un dataset personnalisé comprenant 36 000 images couvrant les signes des lettres (A-Z) et des chiffres (0-9). Après entraînement, le modèle CNN a obtenu une précision de 99,64 %, surpassant légèrement le SVM qui a atteint 99,17 %. Le système développé permet non seulement de reconnaître les signes en temps réel à partir d'une webcam, mais aussi de convertir ces gestes en texte et en parole grâce à des modules de synthèse vocale.

Rokade et Jadav[23] ont proposé un système reposant sur l'acquisition et la segmentation d'images de la main, suivies d'une extraction de caractéristiques et d'une classification des gestes pour la reconnaissance des signes de la langue des signes indienne (ISL). L'ensemble de données utilisé contient 17 signes de l'ISL (A, B, D, E, F, G, H, J, K, O, P, Q, S, T, X, Y, Z), avec des images de résolution 320×240 pixels. La segmentation est effectuée à l'aide d'un modèle probabiliste de détection de la peau, permettant de convertir les images en format binaire.

Pour caractériser la forme des signes, plusieurs techniques sont appliquées, notamment la transformation de distance euclidienne, la projection des distances en lignes et colonnes, ainsi que l'extraction de descripteurs via les moments centraux et les moments de Hu.

Le processus de classification s'appuie sur deux modèles : un réseau de neurones artificiels (ANN) et une machine à vecteurs de support (SVM) avec noyau polynomial. L'ANN est entraîné avec 548 images et testé sur 300 images, atteignant une précision de 94,37 % lorsque 13 caractéristiques sont utilisées. De son côté, le modèle SVM, entraîné avec 479 images et testé sur 369, obtient une précision maximale de 92,12 %.

L'amélioration de la précision est directement liée à l'augmentation du nombre de caractéristiques utilisées, démontrant l'efficacité des moments de Hu et de la transformation de distance pour différencier les signes. En comparaison, l'ANN s'est avéré plus performant que le SVM dans cette tâche, ce qui suggère que les réseaux neuronaux sont mieux adaptés à la reconnaissance des formes complexes de la langue des signes.

L'étude menée par **Quinn et Olszewska** [24] a permis de développer une application mobile atteignant une précision de 99 % dans la reconnaissance des gestes de l'alphabet de la langue des signes britannique (BSL). Cette approche repose sur l'extraction de caractéristiques avec HOG (Histogram of Oriented Gradients) et une classification par un SVM avec un noyau RBF.

Entraîné sur un ensemble de 520 images et testé sur 13 066 échantillons en conditions réelles, le modèle offre un temps de traitement moyen de 170 ms par image, le rendant adapté aux applications en temps réel. Comparée aux méthodes utilisant EOH ou HMM, cette solution se distingue par sa meilleure précision et son efficacité computationnelle.

Selon l'étude de **Nagarajan et Subashini**[25], cette approche repose sur la reconnaissance des gestes statiques de l'ASL en exploitant l'Edge Oriented Histogram (EOH) pour extraire les caractéristiques visuelles et un SVM pour effectuer une classification en plusieurs catégories.

L'ensemble de données comprend un total de 720 images, organisées en 24 classes distinctes, chacune représentée par 30 échantillons. Les images sont acquises via une webcam Zebronics Clarion avec une résolution de 320×240 pixels. Un prétraitement est appliqué aux images : elles sont redimensionnées à 200×200 pixels, converties en niveaux de gris et segmentées afin de séparer la main de l'arrière-plan. Des opérations morphologiques et un filtrage médian sont appliqués pour améliorer la qualité des images, suivis d'une détection des contours avec l'algorithme de Canny.

L'EOH est ensuite utilisé pour extraire des caractéristiques en générant un histogramme des contours avec différentes granularités. L'apprentissage du modèle SVM repose sur un ensemble de 480 images (20 par classe), tandis que 240 images (10 par classe) sont utilisées pour les tests.

Les tests montrent qu'avec 64 bins, le modèle atteint une précision de 93,75 %, bien que des erreurs de classification puissent survenir entre des lettres visuellement proches comme A, E et S.

D'après les travaux **d'Abu-Jamie et Abu-Naser** [26], la reconnaissance de la langue des signes a été réalisée à l'aide d'un réseau de neurones convolutifs (CNN) basé sur l'architecture VGG-16 .

L'étude s'appuie sur un ensemble de données comprenant 43 500 images réparties en 29 catégories (lettres de l'alphabet, espace, suppression et absence de signe), issues d'une base de données publique. Chaque image a été redimensionnée en 64×64 pixels avant l'entraînement du modèle. L'ensemble des données a été divisé en trois parties : 70 % pour l'apprentissage, 15 % pour la validation et 15 % pour le test.

L'efficacité du modèle a été évaluée grâce à la validation croisée k-fold, et les résultats obtenus indiquent un taux de précision de 100 % après 20 époques, mettant ainsi en avant la robustesse de cette approche pour la classification des signes.

D'après **Rathi D.**[27] , cette recherche propose une approche optimisée pour la reconnaissance de la langue des signes américaine (ASL) sur une plateforme mobile. Elle utilise l'apprentissage par transfert, appliqué à deux modèles : MobileNet et Inception V3, réentraînés sur un ensemble de 27 455 images correspondant aux lettres de l'ASL (sauf "J" et "Z").

L'entraînement a été réalisé avec une division des données en 80 % pour l'apprentissage, 10 % pour la validation et 10 % pour les tests. Les modèles ont subi un entraînement sur 5 000 itérations avec un lot de 100 images. MobileNet a obtenu une précision d'entraînement de 97,24

% et une précision de validation de 95,06 %, tandis qu'Inception V3 a atteint une précision d'entraînement de 98,01 % et une précision de validation de 93,36 %.

Dans leur étude *DeepVision Transformer for Sign Language Recognition*, **Kothadiya et al.** [28] explorent l'utilisation des Transformers pour améliorer la reconnaissance des signes en langue des signes indienne (ISL). Contrairement aux méthodes classiques reposant sur des réseaux de neurones convolutionnels (CNN), ce modèle exploite un Vision Transformer (ViT) pour segmenter chaque image en plusieurs parties et analyser leurs relations spatiales à l'aide d'un mécanisme d'attention multi-têtes.

Avant d'être traitées par le Transformer Encoder, les images subissent un prétraitement comprenant diverses transformations comme le redimensionnement et la normalisation. Le modèle est structuré en six couches d'auto-attention et intègre un réseau de neurones MLP pour finaliser la classification des signes.

Les tests réalisés sur un ensemble de 36 classes de signes (lettres et chiffres), avec plus de 1000 images par classe, ont montré une précision de 99,29 % après seulement cinq cycles d'apprentissage. En comparaison avec des approches plus conventionnelles comme CNN et ResNet, cette méthode se distingue par sa robustesse face aux variations d'arrière-plan et de luminosité, garantissant ainsi une reconnaissance fiable dans différents environnements.

Imran et al. [29] présentent une approche avancée pour la reconnaissance en temps réel de la langue des signes américaine (ASL) en utilisant YOLO-v9, une évolution récente de l'algorithme YOLO. Afin d'optimiser la précision et la rapidité du modèle, deux méthodes clés sont mises en place : Programmable Gradient Information (PGI) et Generalized Efficient Layer Aggregation Network (GELAN). PGI utilise une branche auxiliaire réversible pour limiter la perte d'informations et assurer un apprentissage plus efficace grâce à des gradients améliorés. De son côté, GELAN facilite l'agrégation des caractéristiques visuelles, permettant une détection plus performante des gestes sur différentes échelles.

L'architecture du modèle repose sur plusieurs composants : un Backbone dédié à l'extraction des caractéristiques, un Neck pour leur fusion, un Head chargé de la prédiction des signes, ainsi qu'un module auxiliaire qui optimise l'apprentissage. Entraîné sur un ensemble de données couvrant les 26 lettres de l'ASL, YOLO-v9 affiche une précision de 96,83 %, confirmant son efficacité pour l'identification rapide et précise des gestes en langue des signes.

• Tableau récapitulatif

Auteur	Nombre de Classes	Technique utilisée	Résultats (Accuracy)
Orovwode et al. [19]	24	CNN	99,86%
Karim et al. [20]	43	CNN+Mediapipe	99,87%
Paul et al. [21]	26	ResNet50	89,07%
Paul et al.[21]	3	LSTM	94,3%
Paul et al. [21]	3	GRU	79,3%
Katoch et al. [22]	36	CNN + SURF	99,64%
Katoch et al. [22]	36	SVM + SURF	99,17%
Rokade et Jadav [23]	17	ANN	94,37%
Rokade et Jadav[23]	17	SVM	92,12%
Quinn et Olszewska [24]	26	HOG + SVM	99%
Nagarajan et Subashini [25]	24	EOH + SVM	93,75%
Abu-Jamie et Abu-Naser [26]	29	VGG16	100%
Rathi D. [27]	24	MobileNet	97,24%
Rathi D. [27]	24	Inception V3	98,01%
Kothadiya et al. [28]	36	Transformers	99,29%
Imran et al. [29]	26	YOLO-v9	96,83%

Tab. 1.2 : Tableau récapitulatif de la reconnaissance des signes selon différentes méthodes.

1.4 Conclusion

Cette étude de l'existant a permis d'identifier les principales technologies de traduction de la langue des signes ainsi que les méthodes les plus efficaces en reconnaissance gestuelle. Malgré les avancées, les solutions actuelles présentent plusieurs limites en termes de coût, d'ergonomie, d'accessibilité ou encore de performances en temps réel.

Ce constat confirme la pertinence de développer un dispositif innovant, autonome et accessible, s'appuyant sur l'intelligence artificielle pour améliorer l'inclusion des personnes sourdes et muettes dans divers contextes de la vie quotidienne.

En conclusion, ce chapitre a posé les bases nécessaires pour orienter la conception de notre solution, en identifiant les leviers d’innovation technologique à exploiter dans la suite du projet.

Chapitre 2

Fondements Théoriques et Technologiques

2.1 Introduction

Dans le cadre de ce mémoire, il est essentiel de définir un certain nombre de concepts et d'outils sur lesquels repose notre projet. Ce chapitre rassemble les principales définitions nécessaires à la compréhension du système développé, en abordant certaines notions théoriques, architectures de modèles, ainsi qu'une sélection de matériels et de bibliothèques utilisés.

2.2 Concepts de Base

2.2.1 Une image numérique

Une image numérique [30], est une représentation visuelle codée sous forme binaire et exploitée par des dispositifs informatiques. Elle peut être créée directement à l'aide de logiciels utilisant des outils comme la souris, les tablettes graphiques ou la modélisation 3D, produisant ainsi des images de synthèse. Elle peut également être obtenue par conversion d'une source analogique en données numériques, grâce à des équipements tels que les scanners, les appareils photo numériques ou les cartes d'acquisition vidéo. Une fois générée, une image numérique peut être modifiée avec des logiciels de traitement graphique permettant d'en ajuster la taille, les couleurs, ou encore d'y ajouter et supprimer des éléments. Enfin, ces images sont stockées sur divers supports informatiques comme les disques durs, les SSD, les clés USB ou les CD-ROM afin d'être conservées et réutilisées selon les besoins .

Il existe différents types d'images :

a) **Une image matricielle** [30], ou bitmap, est une représentation visuelle composée de pixels en 2D ou de voxels en 3D. Chaque point est défini par sa position et sa couleur. La résolution affecte la qualité d'affichage, variant entre écrans (72-96 dpi) et imprimantes (600 dpi ou plus). En 3D, les pixels deviennent des voxels, et une séquence d'images sur une échelle temporelle forme une animation. Les images stéréoscopiques créent un effet de relief en affichant deux perspectives différentes pour chaque œil, souvent stockées au format ”jps” .

b) **Les images vectorielles** [30], sont représentées par des formules géométriques plutôt que par une grille de pixels. Elles enregistrent les instructions de dessin, comme tracer une ligne entre deux points ou dessiner un cercle de rayon donné. Cela leur permet d'être agrandies sans perte de qualité et d'occuper peu d'espace de stockage. Elles sont particulièrement utilisées pour les schémas techniques dans les logiciels de CAO comme AutoCAD ou CATIA, ainsi que pour les animations web en Flash. Cependant, comme les écrans affichent principalement des images matricielles, les fichiers vectoriels doivent être convertis avant d'être visualisés .

2.2.2 Apprentissage automatique

L'apprentissage automatique, connu sous le nom de **Machine Learning** est une subdivision de l'intelligence artificielle. Elle donne la possibilité aux ordinateurs d'optimiser leurs performances grâce à l'examen des données et à l'expérience acquise[31], sans qu'une programmation explicite soit nécessaire [32]. Tom Mitchell (1997) le définit ainsi : "Un programme informatique apprend de l'expérience E avec une tâche T et une mesure de performance P, si sa performance sur T, mesurée par P, s'améliore avec l'expérience E" [33].

Les principaux types d'apprentissage incluent :

a) **Apprentissage supervisé** : le modèle s'entraîne sur des données étiquetées pour effectuer des tâches comme la classification ou la régression [34].

b) **Apprentissage non supervisé** : il analyse des données non étiquetées pour identifier des tendances ou des regroupements[34].

c) **Apprentissage par renforcement** : L'algorithme adapte ses choix en fonction des récompenses reçues, optimisant ainsi son apprentissage[34].

2.2.3 Apprentissage profond

L'apprentissage profond, également connu sous son appellation anglaise **deep learning** (DL), est une branche avancée de l'intelligence artificielle (IA) et de l'apprentissage automatique (AA). Il repose sur des réseaux neuronaux artificiels, conçus pour imiter la structure et le fonctionnement des neurones biologiques du cerveau humain [35]. Grâce à ces modèles inspirés du cerveau, les ordinateurs peuvent apprendre par l'observation et traiter de vastes quantités de données de manière autonome [36].

Au cœur de cette technologie se trouvent les réseaux neuronaux artificiels (RNA), composés de plusieurs couches interconnectées. Ces réseaux fonctionnent comme un organigramme, débutant par une couche d'entrée qui reçoit les données brutes (images, textes, etc.), suivie de couches cachées où les informations sont progressivement transformées pour en extraire des caractéristiques abstraites. Enfin, la couche de sortie fournit un résultat final, permettant la classification, la reconnaissance ou la prédiction des données traitées [35].

L'apprentissage profond se distingue par sa capacité à traiter des tâches complexes, autrefois réservées à l'intelligence humaine. Il est à la base de nombreuses applications modernes telles que la reconnaissance d'image, la traduction automatique, l'assistance vocale et même la conduite autonome [35]. Différentes architectures de réseaux sont utilisées selon les besoins :

a) **Les réseaux neuronaux convolutifs (CNN)** : spécialisés dans l'analyse des images et la reconnaissance de formes [35].

b) **Les réseaux neuronaux récurrents (RNN)** : adaptés à l'analyse des séquences, comme le texte et la voix [35].

c) **Les réseaux antagonistes génératifs (GAN)** : capables de créer du contenu artificiel, comme des images réalistes [35].

L'efficacité du *deep learning* repose sur un entraînement basé sur de vastes ensembles de données. Durant cette phase, le modèle ajuste continuellement ses paramètres pour améliorer la précision de ses prédictions. Plus le réseau contient de couches, plus il peut capturer des relations complexes et affiner son analyse [36].

Le **deep learning** est une percée révolutionnaire en intelligence artificielle, permettant aux systèmes d'analyser, de s'améliorer et de s'adapter de façon autonome en fonction des données qu'ils traitent [35].

2.3 Architectures de Modèles d'Apprentissage Profond

2.3.1 Perceptron multicouche

Le Perceptron Multicouche (MLP)[34] est un modèle de réseau de neurones artificiel appartenant à la catégorie des réseaux à propagation avant. Il est couramment utilisé pour des applications telles que la classification et l'identification de motifs.

La figure 2.1 présente l'architecture générale d'un Perceptron Multicouche (MLP), il est constitué de plusieurs niveaux de neurones interconnectés :

a) Un premier niveau, appelé **couche d'entrée**, reçoit les données sous forme de vecteurs et les envoie au reste du réseau.

b) **Les couches intermédiaires**, dites **cachées** , traitent les informations en appliquant des pondérations aux entrées et en passant les résultats dans une fonction d'activation non linéaire, comme la tangente hyperbolique ou la sigmoïde. Cependant, l'auteur ne mentionne pas l'utilisation de ReLU dans les couches cachées des MLP, alors qu'elle est largement adoptée aujourd'hui. Elle est donc pertinente dans notre définition.

c) Le dernier niveau, appelé **couche de sortie**. Elle est responsable de la génération du résultat final adapté au problème traité :

- Dans le cas d'une classification binaire, un seul neurone de sortie doté d'une activation sigmoïde permet d'estimer la probabilité d'appartenance à une catégorie.
- Lorsqu'il s'agit de classer des données en plusieurs catégories, la couche de sortie comprend plusieurs neurones et applique une activation softmax pour convertir les scores en probabilités, la somme de ces derniers est égale à 1 .
- Dans un problème de régression un unique neurone en sortie utilise une activation linéaire afin de générer une valeur numérique continue.

Le MLP utilise la rétropropagation de l'erreur pour apprendre, en modifiant progressivement les poids des connexions afin de minimiser l'écart entre les résultats obtenus et ceux attendus.

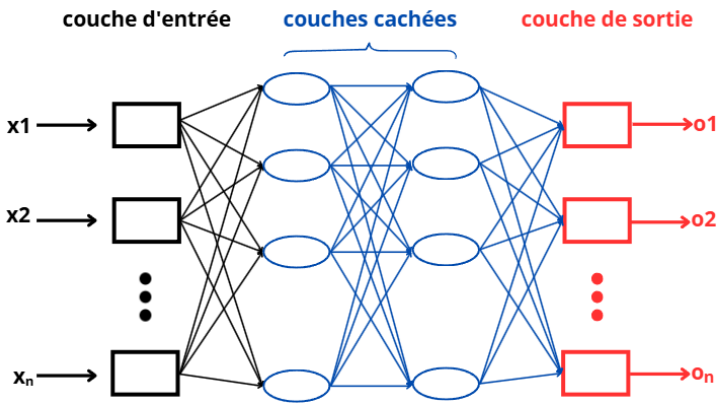

Fig. 2.1 : Exemple d'architecture d'un MLP avec 2 couches cachées

2.3.2 Réseaux de Neurones Convolutifs

Les réseaux de neurones convolutifs (CNN) sont un type particulier de réseau de neurones artificiels, spécifiquement conçus pour traiter des données structurées en grille, telles que les images (grilles 2D de pixels) ou les séries temporelles (grilles 1D) [37]. Inspirés par le fonctionnement du cortex visuel du cerveau humain [38], les CNN ont révolutionné le traitement automatique des images en permettant l'extraction hiérarchique de caractéristiques à différents niveaux de complexité [41]. Contrairement aux réseaux neuronaux traditionnels, ils reposent sur une opération mathématique appelée convolution, qui remplace les multiplications matricielles classiques dans certaines couches, et permet ainsi d'identifier automatiquement des motifs locaux dans les données d'entrée [37]. L'architecture d'un CNN repose sur une succession de couches spécifiques — notamment les couches de convolution, de regroupement (pooling) et les couches entièrement connectées — qui contribuent à réduire la taille des données tout en conservant les informations les plus pertinentes [41]. Par ailleurs, cette approche traite les images comme des matrices de pixels, en tenant compte de leurs dimensions et du nombre de canaux. Par exemple, une image de 32×32 pixels avec trois canaux (RGB) est représentée sous la forme $(32 \times 32 \times 3)$[38].

2.3.2.1 Couche de Convolution

Dans un réseau de neurones convolutifs (CNN), la couche convolutionnelle[38] joue un rôle fondamental en détectant les motifs locaux d'une image. Contrairement aux couches classiques entièrement connectées, elle ne traite pas l'image entière d'un coup mais analyse des sous-régions spécifiques. En appliquant des filtres (ou noyaux) glissants sur l'image d'entrée, elle extrait progressivement des caractéristiques simples, comme les contours ou les textures. Ces informations sont ensuite combinées dans les couches suivantes pour identifier des structures plus complexes, permettant ainsi une reconnaissance efficace des objets.

2.3.2.2 Couche de Pooling

Pour optimiser le traitement et éviter un trop grand nombre de paramètres, les CNN utilisent des couches de pooling[38]. Celles-ci ont pour objectif de réduire la taille des données tout en conservant les caractéristiques essentielles. L'une des méthodes les plus courantes est le max-pooling, qui sélectionne uniquement la valeur la plus élevée dans une région donnée. Contrairement aux couches de convolution, cette couche ne nécessite pas de poids à apprendre ; elle applique simplement une fonction d'agrégation sur les valeurs d'entrée .

La figure 2.2 montre comment fonctionne le max-pooling :

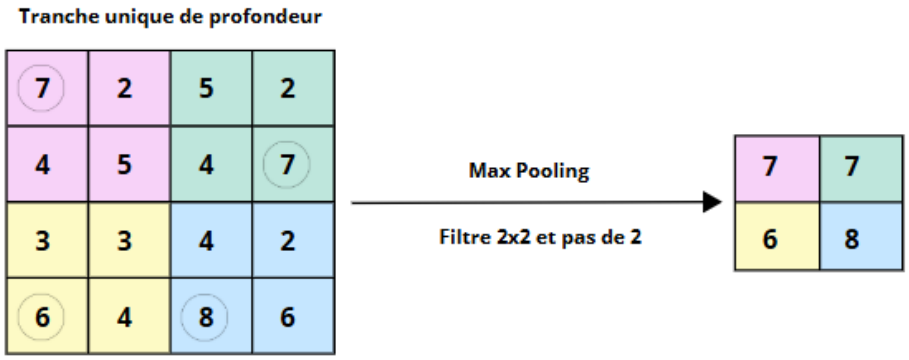

Fig. 2.2 : Exemple de Max pooling

2.3.2.3 Couche Entièrement Connectée (Dense Layer)

Une couche dense se réfère à une structure de neurones complètement interconnectée dans un réseau de neurones artificiels, où chaque neurone est connecté à tous les neurones de la précédente couche[39]. Chaque neurone traite les données reçues en effectuant une somme des entrées pondérées par des coefficients ajustables, y ajoute un biais, puis applique une fonction d'activation afin de modéliser des relations non linéaires .Ce type de couche est généralement utilisé à la fin d'un CNN (réseau de neurones convolutif) pour effectuer la classification finale, mais il est également présent dans d'autres types de réseaux de neurones, comme les réseaux entièrement connectés (MLP)[40].

Les CNN ont démontré une efficacité remarquable dans de nombreux domaines d'application, allant de la reconnaissance d'images et d'objets à l'analyse sémantique de contenu visuel, en passant par la reconnaissance vocale et le traitement du langage naturel (NLP) [38]. Leur capacité à construire des représentations hiérarchiques les rend robustes face aux variations des données d'entrée, ce qui les rend particulièrement utiles dans des contextes complexes comme les véhicules autonomes ou le diagnostic médical assisté par l'image [41].

2.4 Apprentissage par transfert

Le transfert d'apprentissage (TL)[42], est une technique en machine learning où un modèle déjà entraîné sur une tâche est ajusté pour en accomplir une autre similaire. Plutôt que de développer un nouveau modèle à partir de zéro, ce qui demande beaucoup de données, de calcul et de temps, cette approche permet de tirer parti des connaissances acquises par un modèle existant. Par exemple, un modèle ayant appris à reconnaître une catégorie d'images peut être affiné pour identifier une autre catégorie avec moins de données et d'entraînement .

2.4.1 Étapes du Transfer Learning

L'apprentissage par transfert suit plusieurs étapes essentielles pour adapter un modèle préentraîné à une nouvelle tâche :

a) **Choisir un modèle préentraîné :** Sélectionner un modèle ayant déjà appris des caractéristiques pertinentes sur une tâche proche de celle que l'on veut réaliser.

b) **Adapter la structure du modèle :** Pour adapter un modèle pré-entraîné à une nouvelle tâche, on peut utiliser plusieurs stratégies. D'abord, il est possible de geler certaines couches, notamment les premières, afin de conserver les connaissances acquises sur des caractéristiques générales comme les bords et les textures. Ensuite, on peut modifier ou supprimer des couches, en ajustant les dernières couches du réseau pour mieux correspondre aux nouvelles classes à reconnaître. Enfin, il est parfois nécessaire d'ajouter de nouvelles couches, spécifiquement conçues pour capturer les particularités de la nouvelle tâche, améliorant ainsi la capacité du modèle à s'adapter sans perdre les informations essentielles apprises précédemment.

c) **Réentraîner le modèle sur de nouvelles données :** Affiner les paramètres en utilisant un ensemble de données adaptées à la tâche cible. Pendant cet entraînement, il est possible d'ajuster des hyperparamètres (comme le taux d'apprentissage) afin d'améliorer la précision du modèle.

2.4.2 Stratégies d'apprentissage par transfert

Selon la nature de la tâche et les données disponibles, plusieurs stratégies d'apprentissage par transfert peuvent être appliquées. Ces stratégies influencent la manière dont le modèle est ajusté et réentraîné afin de mieux répondre aux exigences de la nouvelle tâche.

a) **Apprentissage par transfert transductif :** Cette approche consiste à adapter un modèle entraîné sur un domaine à un autre domaine similaire mais distinct. Elle est particulièrement utile lorsque les données étiquetées sont limitées dans le domaine cible. Par exemple, un modèle d'analyse des sentiments formé sur des critiques de produits peut être utilisé pour analyser des critiques de films, car les structures linguistiques restent comparables.

b) **Apprentissage par transfert inductif :** Ici, le domaine source et le domaine cible sont identiques, mais les tâches diffèrent. Un modèle préentraîné sur un large corpus de textes, par exemple, peut être affiné pour des tâches spécifiques comme la classification de sentiments ou la reconnaissance d'entités. De même, en vision par ordinateur, un modèle généraliste peut être ajusté pour la détection d'objets précis.

c) **Apprentissage par transfert non supervisé :** Cette méthode est utilisée lorsque ni le domaine source ni le domaine cible ne disposent de données étiquetées. Le modèle apprend à identifier des motifs communs dans les données non étiquetées avant d'être affiné pour une tâche spécifique. Par exemple, un modèle entraîné sur un grand ensemble d'images de véhicules peut être affiné pour mieux reconnaître les motos sans avoir besoin d'annotations détaillées au départ.

Chaque stratégie permet d'adapter un modèle en fonction des besoins spécifiques et des ressources disponibles .

2.5 Modèles Préentraînés Utilisés

2.5.1 MobileNetV2

MobileNetV2[43] est une architecture de réseau neuronal conçue spécialement pour les circonstances à ressources restreintes, tels que les dispositifs mobiles. Elle introduit une innovation en architecture nommée inverted residual block avec linear bottleneck montrée à la figure 2.3, qui est le cœur de son efficacité. Ce bloc reçoit une représentation condensée, la projette dans un espace de dimension supérieure grâce à une couche d'expansion, effectue par la suite des convolutions depthwise pour capter les caractéristiques, et finalement réduit de nouveau la dimensionnalité par une projection linéaire. Ce processus permet de maintenir une grande capacité de représentation tout en diminuant de manière importante le coût computationnel et l'utilisation de mémoire.

Dans cette structure, un choix clé est l'élimination intentionnelle des non-linéarités dans les couches de goulot d'étranglement (bottlenecks), une approche conçue pour prévenir la perte d'information lorsque les représentations sont fortement compressées. Cette approche optimise les performances générales du modèle, contrairement aux architectures traditionnelles qui utilisent des activations telles que ReLU partout dans le réseau.

Par ailleurs, MobileNetV2 se distingue par l'utilisation de shortcuts qui relient directement les couches étroites au sein des blocs inversés, contrairement aux residual connections traditionnelles qui relient des couches larges. Ce renversement structurel permet non seulement une meilleure circulation du gradient, mais améliore aussi l'efficacité mémoire qui est un avantage majeur pour les déploiements sur des dispositifs embarqués.

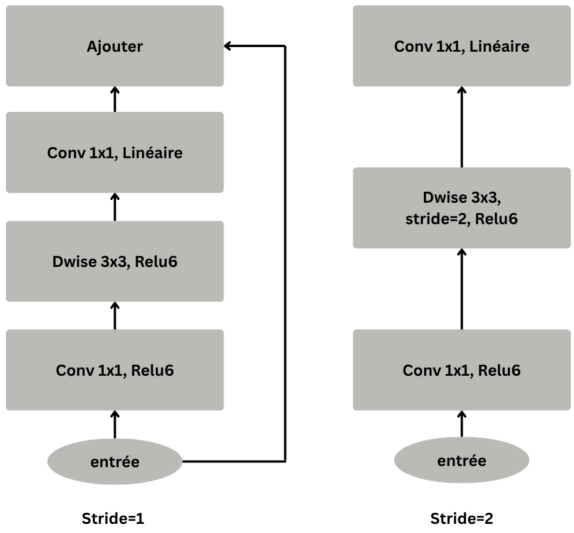

Fig. 2.3 : Blocs fondamentaux de l'architecture MobileNetV2

Il est important de noter que des bibliothèques telles que Keras proposent MobileNetV2 en tant que modèle déjà entrainé. Pour adapter les données d'entrée à ce modèle, il faut faire appel à la fonction tf.keras.applications.mobilenet_v2.preprocess_input. Conformément à la documentation officielle, cette fonctionnalité normalise les pixels en les réduisant à une échelle de valeurs allant de -1 à 1. [44]

2.5.2 EfficientNet-B0

EfficientNet-B0[45] marque le point de départ d'une série de réseaux de neurones convolutifs modernes, dont l'objectif est de combiner précision en classification et sobriété computationnelle. Ce modèle, fondé sur MnasNet, a été généré à l'aide d'une méthode automatisée de conception d'architecture (Neural Architecture Search) optimisant simultanément la performance sur ImageNet et la réduction du coût calculatoire mesuré en FLOPs.

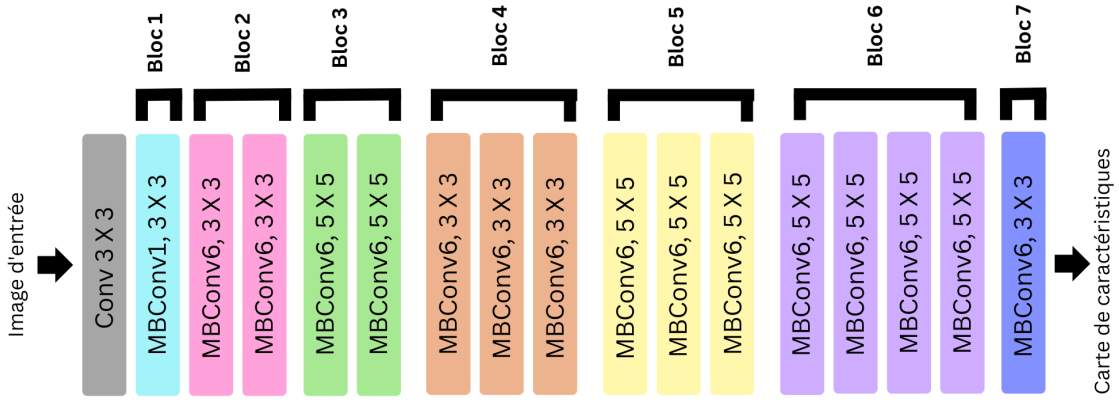

Fig. 2.4 : Architecture de EfficientNet-B0

L'ossature du réseau repose principalement sur les blocs MBConv, comme illustré à la figure 2.4, enrichis par des modules squeeze-and-excitation (SE) qui réajustent dynamiquement l'importance des canaux de caractéristiques. Par ailleurs, la fonction d'activation Swish, choisie pour son efficacité en apprentissage profond, y est systématiquement utilisée.

En termes de performances, EfficientNet-B0 se distingue par une précision top-1 de 77,1% sur ImageNet, tout en demeurant léger avec seulement 5,3 millions de paramètres et une complexité de 0,39 milliard de FLOPs. Ces performances surpassent nettement celles de modèles standards tels que ResNet-50, avec un coût computationnel nettement inférieur.

Enfin, ce modèle sert de base à une montée en puissance progressive vers des variantes plus élaborées (EfficientNet-B1 à B7), obtenues grâce à une stratégie d'élargissement dite compound scaling, qui ajuste simultanément la profondeur du modèle, la largeur des couches et la résolution des images, assurant ainsi une échelle de croissance équilibrée et cohérente.

2.5.3 EfficientNetV2-S

EfficientNetV2-S[46] constitue une architecture de réseau de neurones convolutifs conçue spécifiquement pour concilier efficacité d'entraînement et compacité du modèle. Élaboré à l'aide d'une méthode de recherche d'architecture automatisée sensible aux contraintes de temps d'apprentissage, ce modèle intègre des choix structurels optimisés pour les accélérateurs matériels modernes.

Sa conception repose sur une combinaison équilibrée de blocs MBConv et Fused-MBConv (voir la figure 2.5), utilisés de manière sélective selon les étapes du réseau, dans le but d'améliorer les performances tout en maîtrisant la complexité computationnelle. L'architecture adopte une disposition non uniforme des couches, accordant une plus grande profondeur aux parties intermédiaires du réseau, tout en limitant la taille des noyaux de convolution et le taux d'expansion afin de réduire l'accès mémoire et les temps de calcul.

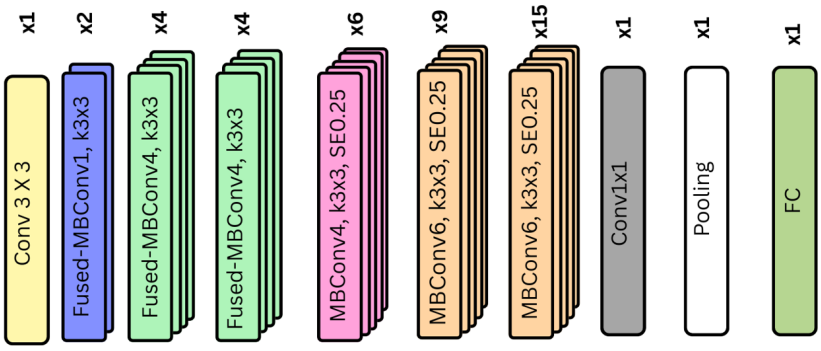

Fig. 2.5 : Architecture de EfficientNetV2-S

Cette approche architecturale est couplée à un mécanisme d'apprentissage progressif, qui ajuste dynamiquement la taille des images d'entrée et l'intensité des techniques de régularisation au fil des étapes de l'entraînement. Ce procédé permet au modèle de maintenir un bon équilibre entre vitesse de convergence et capacité de généralisation.

Au regard des résultats expérimentaux rapportés, EfficientNetV2-S atteint une précision Top-1 de 83,9 % sur ImageNet, tout en ne mobilisant que 22 millions de paramètres et 8,8 milliards d'opérations, illustrant ainsi son efficacité remarquable tant en termes de performance que de ressources requises.

2.6 Outils et Bibliothèques

2.6.1 Mediapipe

Développé par Google, MediaPipe[47] est un outil conçu pour analyser des flux d'images et de vidéos en temps réel à l'aide de modèles d'inférence. Il repose sur une structure modulaire où chaque composant traite une tâche spécifique, comme la détection de visages, la localisation d'objets et le suivi d'objets. Son architecture optimisée permet une exécution fluide sur différents supports, comme des smartphones ou ordinateurs. Il est conçu pour les applications de vision par ordinateur et de machine learning.

2.6.1.1 Mediapipe Hands

MediaPipe Hands[48] est une technologie innovante permettant de suivre et d'analyser les mouvements des mains en temps réel à l'aide d'une simple caméra RGB, sans nécessiter de capteurs spécialisés comme les capteurs de profondeur. Conçue pour fonctionner sur divers appareils, y compris les smartphones et ordinateurs, cette solution facilite l'intégration de la reconnaissance gestuelle dans des environnements interactifs tels que la réalité augmentée (AR) et la réalité virtuelle (VR).

Le système repose sur un processus en deux étapes :

a) tout d'abord, un détecteur identifie la paume de la main afin d'assurer un suivi stable, même en cas d'occlusion ou de gestes complexes.

b) Ensuite, un second modèle affine cette détection en repérant précisément 21 points clés correspondant aux articulations et aux doigts, visibles à la figure 2.6, permettant ainsi de reconstruire la position de la main en trois dimensions.

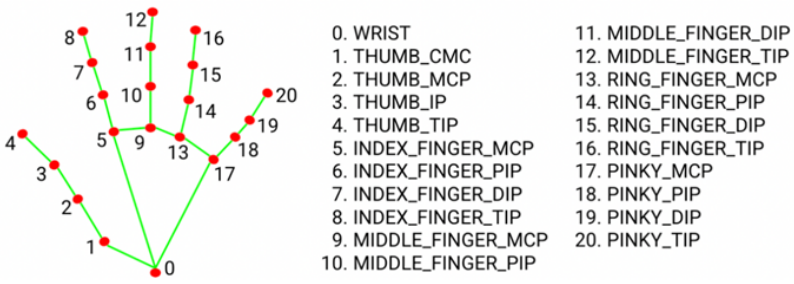

Fig. 2.6 : Les 21 points clés détectés par MediaPipe Hands [49]

De plus, ce modèle est capable de distinguer une main gauche d'une main droite et d'évaluer la fiabilité de ses prédictions cela veut dire que le modèle attribue un niveau de confiance à chaque résultat qu'il produit. En d'autres termes, il estime à quel point il est sûr de la position des points clés détectés sur la main.

Grâce à sa rapidité et sa précision, cette technologie favorise une interaction fluide avec les interfaces numériques, rendant possible le contrôle gestuel sans nécessiter d'équipement supplémentaire.

2.6.2 TensorFlow Lite

TensorFlow Lite[50] est une version optimisée de TensorFlow, conçue pour permettre l'exécution de modèles d'apprentissage automatique sur des appareils aux ressources limitées, tels que les dispositifs mobiles, les systèmes embarqués et les microcontrôleurs, y compris des plateformes comme le Raspberry Pi. Elle permet de faire fonctionner des modèles d'apprentissage automatique directement sur ces dispositifs, tout en étant optimisée pour les environnements à faibles ressources en calcul et en mémoire.

La création d'un modèle TensorFlow Lite peut s'effectuer de différentes manières : il est possible de partir d'un modèle existant, d'en entraîner un nouveau à l'aide de l'outil Model Maker adapté à des jeux de données personnalisés, ou encore de transformer un modèle TensorFlow classique à l'aide d'un convertisseur dédié.

2.7 Concepts Techniques Connexes

2.7.1 Couche dropout

Dropout[51] est une technique destinée à optimiser la capacité de généralisation des réseaux de neurones en atténuant le phénomène d'overfitting. Au cours de l'entraînement, certaines unités et leurs liens sont désactivés au hasard, ce qui évite une forte interdépendance entre les neurones et encourage un apprentissage plus autonome et résilient. Ce principe est illustré à la figure 2.7, qui montre un réseau avant et après l'application du Dropout.

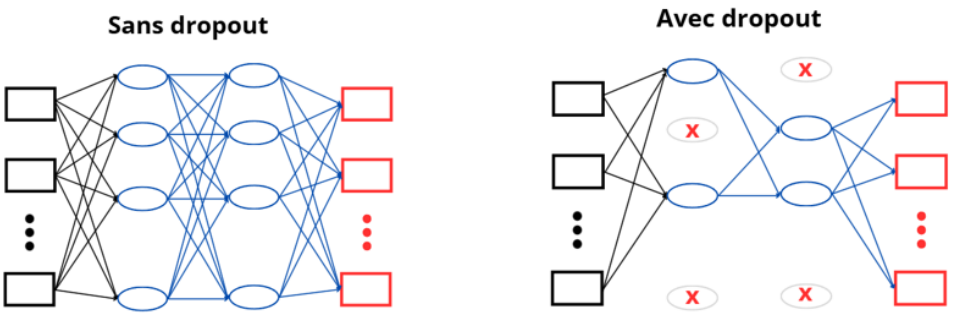

Fig. 2.7 : Avant et Après Application du Dropout dans un Réseau de Neurones

2.7.2 Batch Normalization

La batch normalization[52] , est une méthode utilisée pour améliorer l'apprentissage des réseaux convolutifs. Elle consiste à normaliser les activations d'une couche avant leur passage dans la fonction d'activation, comme ReLU. Cette technique limite les variations internes des activations, un phénomène qui peut ralentir l'entraînement du modèle.

En régulant les valeurs en entrée des couches suivantes, la batch normalization stabilise l'apprentissage et permet d'utiliser des taux d'apprentissage plus élevés, réduisant ainsi le temps nécessaire à la convergence du modèle. De plus, elle aide chaque couche à apprendre plus efficacement sans trop dépendre des précédentes. Contrairement au dropout, qui est une autre approche de régularisation, la batch normalization s'avère généralement plus adaptée aux réseaux convolutifs .

2.7.3 Padding

Dans les réseaux de neurones convolutifs (CNN) , le padding[53] joue un rôle clé en ajoutant des pixels supplémentaires autour des images avant l'application des filtres de convolution. Cette technique permet de mieux préserver les informations situées aux bords des images et d'influencer la taille des cartes de caractéristiques produites par le réseau.

Il existe plusieurs types de padding : le zero padding, qui insère des pixels de valeur nulle autour de l'image, le same padding, qui ajuste la convolution de manière à conserver la même taille entre l'entrée et la sortie, et le valid padding, qui n'ajoute aucun pixel supplémentaire et réduit progressivement la taille de l'image après chaque convolution. Le choix de la méthode de padding peut avoir un impact direct sur la performance du modèle et la qualité des caractéristiques extraites.

Un exemple de cette technique est illustré à la figure 2.8 :

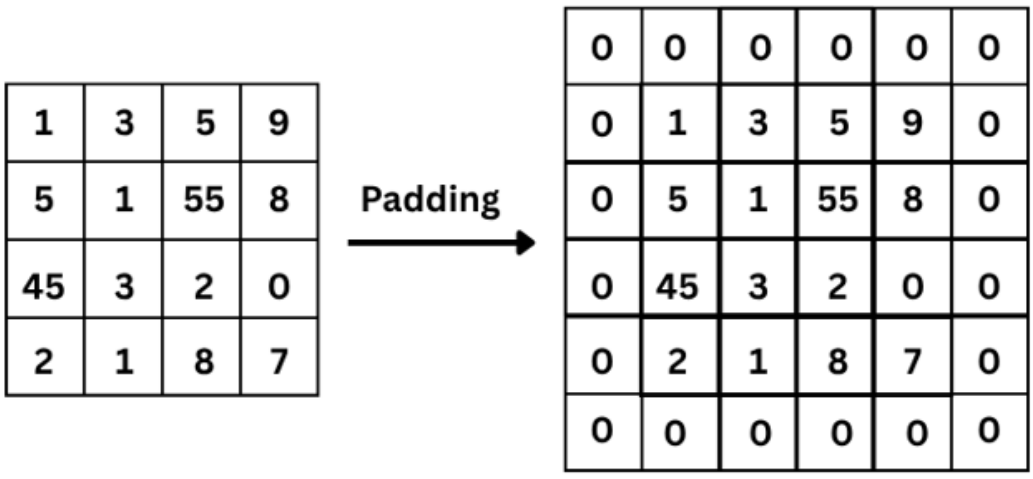

Fig. 2.8 : Exemple de padding

2.7.4 Stride

Dans une convolution, le stride[53] correspond au nombre de pixels de déplacement du noyau sur l'image d'entrée à chaque étape du calcul. Plus précisément, il définit l'écart entre deux applications successives du filtre sur l'entrée.

Un stride faible, comme 1, permet une analyse plus fine des détails, car le filtre parcourt progressivement toute l'image, ce qui génère une sortie plus grande avec davantage de caractéristiques extraites. À l'inverse, un stride plus élevé réduit le nombre de positions où le noyau est appliqué, ce qui diminue la taille de la sortie et accélère les calculs, mais peut entraîner une perte d'informations.

2.8 Matériel Utilisé

2.8.1 Raspberry Pi

Le Raspberry Pi[54] est un micro-ordinateur à bas prix, disponible pour environ 35 dollars, qui propose des performances surprenantes compte tenu de sa taille. Il embarque les principales options d'un ordinateur moderne, ainsi qu'une multitude de ports USB pour divers dispositifs. Ses capacités et sa mémoire facilitent la réalisation aisée des tâches numériques les plus fréquentes. Du fait de sa polyvalence, il peut servir d'ordinateur de bureau, de station multimédia, de contrôleur domotique ou même de cerveau pour des projets robotiques. De plus, sa consommation d'énergie réduite, d'environ 15 watts, le positionne comme une option à la fois économique et écologique .

Le Raspberry Pi se varie en plusieurs versions, chacune offrant des améliorations concernant les performances, la connectivité et les fonctionnalités, afin de s'adapter à des exigences et applications toujours plus diverses. Prenons un exemple :

2.8.1.1 Raspberry Pi 4 Model B

Fig. 2.9 : Raspberry Pi 4 Model B [55]

La figure 2.9 montre le Raspberry Pi 4 Model B, dont les caractéristiques sont détaillées ci-dessous :

- Il possède un processeur quad-core basé sur l'architecture Cortex-A72, fonctionnant à une fréquence de 1,8 GHz. [56]
- La mémoire vive peut être configurée en plusieurs options : 1 Go, 2 Go, 4 Go ou 8 Go de type LPDDR4.[56]
- Ce dispositif supporte le Wi-Fi dual-band ainsi que la technologie Bluetooth version 5.0.[56]
- Il est équipé d'un port Ethernet à large bande.[56]
- L'équipement est doté de deux connexions USB 3.0 ainsi que de deux connexions USB 2.0 pour la liaison avec des appareils externes.[56]
- Il comprend deux ports micro-HDMI aptes à gérer une qualité d'image allant jusqu'à la 4K.[56]
- Le stockage se fait à l'aide d'une carte microSD.[56]
- Une interface GPIO disposant de 40 broches facilite la connexion avec d'autres éléments électroniques.[56]
- Des ports dédiés sont prévus pour le raccordement d'un écran (DSI) et d'une caméra (CSI).[56]

2.8.2 Camera Raspberry Pi

Les caméras conçues pour les systèmes Raspberry Pi[57] sont des extensions matérielles qui s'interfacent directement avec la carte via le port MIPI CSI, assurant une connexion optimisée pour le transfert de données visuelles. En raison de leur conception flexible, ces caméras peuvent être intégrées dans divers domaines, allant de traitement automatisé d'images à la surveillance ou les expérimentations scientifiques. La gamme disponible se décline en plusieurs modèles adaptés à ces usages spécifiques.

2.8.2.1 Caméra Raspberry Pi Module 3

Caméra Raspberry Pi Module 3[59], illustrée à la figure 2.10, intègre un capteur Sony IMX708 de 12 mégapixels avec une résolution de 4608 x 2592 pixels. Elle offre une mise au point rapide par détection de phase (PDAF) et un filtre infrarouge intégré dans les variantes standard. Elle enregistre des vidéos en 1080p à 50 fps, 720p à 100 fps et 480p à 120 fps, et produit des sorties en RAW10. Le connecteur de câble est un FPC 15 x 1 mm, et la longueur du câble est de 200 mm. La température de fonctionnement de la caméra varie de 0°C à 50°C et elle répond aux exigences des normes FCC et RoHS. La production de la Raspberry Pi Camera Module 3 est prévue pour continuer jusqu'à janvier 2030, garantissant ainsi sa disponibilité à long terme.

Fig. 2.10 : Caméra Raspberry Pi Module 3 [58]

2.8.3 Arduino

Arduino[60] est une plateforme technologique open-source conçue pour simplifier la création de projets électroniques interactifs. Elle repose sur des cartes électroniques équipées de microcontrôleurs que l'on peut programmer pour interagir avec leur environnement. Ces cartes sont capables de recevoir des signaux variés — tels que la détection de lumière, la pression sur un bouton, ou des données provenant d'internet — et d'y répondre par des actions concrètes comme l'activation d'un moteur, l'affichage d'un message ou encore l'éclairage d'une LED.

L'utilisateur programme ces cartes via un environnement de développement intégré (IDE) accessible et intuitif. Ce logiciel utilise un langage de programmation dérivé de Wiring, tandis que l'interface s'inspire de Processing, rendant l'ensemble particulièrement adapté aux débutants en électronique et en informatique.

Le projet Arduino est né dans un cadre éducatif, plus précisément à l'Institut de Design d'Interaction d'Ivrea, en Italie. Il avait pour objectif de fournir un outil de prototypage rapide à des étudiants ne disposant pas nécessairement de connaissances techniques avancées. L'idée a très vite trouvé un écho au-delà du cadre académique.

Aujourd'hui, Arduino occupe une place centrale dans de nombreux domaines : objets connectés (IoT), dispositifs portables (wearables), robotique, impression 3D, et même recherche scientifique. Son accessibilité a permis de démocratiser la création électronique, ouvrant la porte à un large éventail d'utilisateurs : étudiants, artistes, passionnés de technologie, enseignants ou ingénieurs. Une communauté mondiale très active s'est constituée autour d'Arduino. Grâce au partage continu de projets, de bibliothèques, de tutoriels et d'exemples, une immense base de ressources s'est développée. Cette dynamique collaborative offre un tremplin précieux pour les débutants tout en proposant des solutions avancées pour les utilisateurs expérimentés.

La figure 2.11 présente un exemple de carte Arduino :

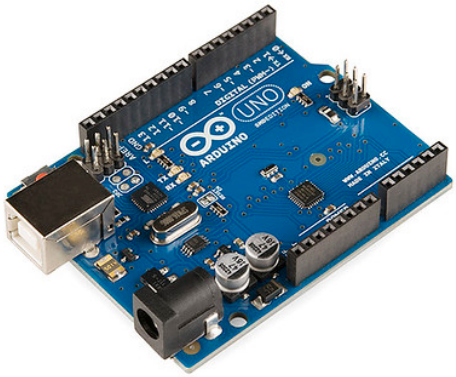

Fig. 2.11 : Arduino Uno R3 [61]

2.9 Conclusion

En conclusion de ce chapitre, nous avons présenté les concepts fondamentaux liés à la vision par ordinateur et à l'apprentissage automatique, en mettant l'accent sur les approches de deep learning, notamment les réseaux de neurones convolutifs (CNN) et le transfer learning. Nous avons également examiné les architectures de modèles, ainsi que les outils, bibliothèques et ressources matérielles qui soutiennent la mise en œuvre de notre solution.

Cette base théorique et technique constitue un socle solide pour la conception et le développement de notre système de reconnaissance de la langue des signes. Elle nous permet de mieux comprendre les choix technologiques adoptés dans la suite du projet, en assurant une cohérence entre les objectifs visés et les moyens utilisés.

Chapitre 3

Contribution et résultats

3.1 Introduction

L'accès à une communication fluide constitue un droit fondamental qui ne saurait exclure quiconque. Cependant, la barrière linguistique entre les utilisateurs de la langue des signes et ceux qui ne la maîtrisent pas représente un obstacle majeur, en particulier dans les échanges du quotidien, Affectant ainsi leur intégration sociale et leur indépendance.

Malgré quelques tentatives d'innovation dans ce domaine, les solutions technologiques disponibles aujourd'hui pour répondre aux besoins des personnes sourdes ou muettes sont encore limitées. En plus d'être rares, ces dispositifs sont généralement coûteux, encombrants et peu accessibles, ce qui limite fortement leur adoption à grande échelle.

Dans ce contexte, nous avons conçu un prototype de lunettes intelligentes visant à faciliter la communication entre les usagers de la langue des signes et le grand public non formé à cette langue. Ce chapitre détaille notre contribution à travers la présentation du prototype, la méthodologie employée, les outils utilisés, ainsi que les résultats obtenus lors des expérimentations et les performances des modèles testés.

3.2 Présentation du prototype

Le prototype est constitué de plusieurs composants électroniques interconnectés comme illustré sur les figures 3.1 et 3.2. , chacun jouant un rôle précis dans le processus de reconnaissance des gestes de la langue des signes et de leur conversion en texte et en parole.

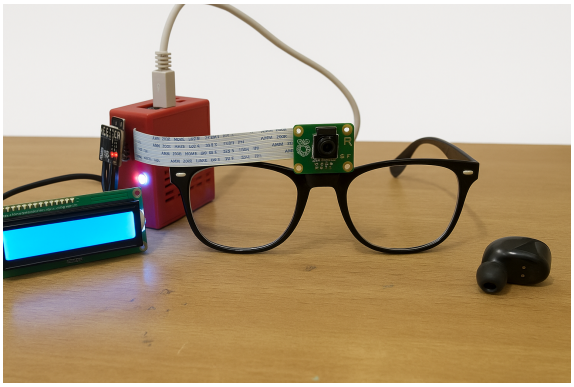

Fig. 3.1 : Première version finalisée des lunettes intelligentes pour la reconnaissance de la langue des signes

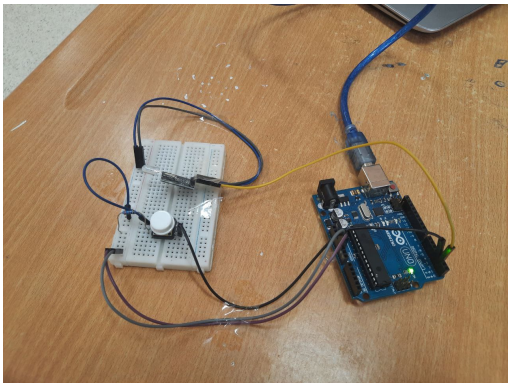

Fig. 3.2 : Dispositif secondaire destiné à alerter lorsqu'une personne sourde/muette veut entrer en communication.

Voici les éléments principaux utilisés :

a) **Raspberry Pi 4 (4 Go de RAM) :** Il agit comme le cœur du système. Il exécute le modèle d'intelligence artificielle pour interpréter les gestes capturés et prédire les signes correspondants.

b) **Caméra Pi Module 3 :** Montée à l'avant des lunettes, elle capte les gestes de la main de la personne sourde ou muette pour les transmettre au modèle de reconnaissance.

c) **Arduino avec bouton-poussoir et LED infrarouge (IR) :** Placé sur La personne sourde ou muette , il permet d'initier une communication en envoyant un signal IR dès que le bouton est pressé.

d) **Récepteur infrarouge (IR) :** Intégré aux lunettes, il détecte le signal IR émis par l'Arduino et déclenche l'activation du système.

e) **Écouteurs :** Ils jouent un double rôle : alerter l'interlocuteur à travers un signal sonore et transmettre la synthèse vocale du texte reconnu pour faciliter la compréhension.

f) **Écran LCD I2C 16x2 :** Il affiche en temps réel les lettres, mots ou phrases générés à partir des gestes détectés, permettant une lecture visuelle de la communication.

Ces composants fonctionnent de manière synchronisée pour assurer une interaction fluide, rapide et accessible entre une personne sourde ou muette et son interlocuteur.

3.3 Méthodologie

3.3.1 Dataset utilisé

Dans le cadre de nos travaux de recherche pour trouver un jeu de données approprié à l'entraînement de notre modèle de reconnaissance de la langue des signes, nous avons choisi de nous baser sur l'alphabet de la langue des signes américaine (ASL), étant donné que la majorité des jeux de données disponibles en ligne utilisent cette langue.

Nous avons consulté divers jeux de données sur Kaggle. Toutefois, nous avons identifié une problématique significative : la plupart de ces ensembles de données ont été constitués en tirant des images de vidéos. Par conséquent, chaque catégorie était représentée par une seule vidéo dont les images ont été extraites, entraînant une grande ressemblance entre les images de la même classe.

Cette ressemblance entre les images d'une même classe constituait un obstacle pour la formation du modèle. Cela provoquerait effectivement une réduction significative de la diversité des données, augmentant par conséquent le risque de surapprentissage plutôt que de développer des aptitudes à la généralisation pour chaque signe. Cela aurait restreint la capacité à identifier les signes performés par différentes personnes dans divers environnements.

Face à ce problème, nous avons opté pour la création de notre propre jeu de données en choisissant des images issues de divers jeux de données existants. Cette approche nous a aidés à diversifier les exemples pour chaque classe, ce qui a augmenté la solidité de notre modèle et amélioré sa capacité à généraliser.

Nous avons conçu une sélection manuelle d'images originaires de 9 ensembles de données différents sur Kaggle comme le montre le tableau ci-dessous. Cette procédure a été soigneusement réalisée pour garantir une qualité d'image impeccable et enrichir la diversité des échan-

tillons, tout en prévenant toute répétition. Bien que cette mission ait nécessité du temps et des efforts considérables, elle était indispensable pour assurer un entraînement fiable du modèle.

Dataset utilisé	**Nombre d'images de l'alphabet prises par classe**
1er dataset [62]	400
2e dataset [63]	150
3e dataset [64]	70
4e dataset [65]	∼50
5e dataset [66]	∼80
6e dataset [67]	∼50
7e dataset [68]	∼80
8e dataset [69]	∼20
9e dataset [70]	∼20
Total	920

Tab. 3.1 : Répartition des images sélectionnées par classe pour chaque dataset

En ce qui concerne les classes Space et Delete, qui ne font pas partie des classes officielles de la langue des signes, nous n'avons pas réussi à rassembler assez de données pour les créer à partir des ensembles de données déjà existants. Pour remédier à cette lacune, nous avons construit ces deux classes en prenant nos propres photos grâce à une caméra de smartphone.

Notre dataset final comprend 28 classes, avec 920 images par classe , soit un total de 25 760 images organisées dans des dossiers, chaque dossier correspondant à une classe.

La figure 3.3 montre un aperçu des 28 classes constituant notre jeu de données, chacune étant représentée par une image illustrant un exemple typique de la classe correspondante. Cela permet de visualiser la diversité des signes capturés et la cohérence de l'organisation des données.

Fig. 3.3 : Aperçu des 28 classes constituant notre jeu de données. Chaque image représente un exemple d'une classe différente.

3.3.2 Algorithmes utilisés

Dans le cadre de notre projet de lunettes intelligentes pour la traduction de la langue des signes, nous avons d'abord testé des modèles **CNN** entraînés depuis zéro, avant de nous orienter vers des solutions plus adaptées aux contraintes du système embarqué, notamment un **MLP** et **des modèles préentraînés** sur le jeu de données ImageNet pour la classification d'images.

Notre choix s'est porté sur le **MLP** pour notre projet, et ce, pour plusieurs raisons.

Tout d'abord , l'existence de MediaPipe Hands facilite l'extraction de caractéristiques sous forme de vecteurs numériques, ce qui rend l'utilisation d'un MLP particulièrement appropriée, car il est bien ajusté pour traiter ce type de données tabulaires, permettant ainsi un entraînement efficace et une exécution de modèle rapide.

De plus, étant donné que notre système est destiné à être utilisé sur un Raspberry Pi, il est crucial d'opter pour un modèle qui consomme peu de ressources et qui est léger. Le MLP consomme moins de puissance de calcul tout en offrant des performances satisfaisantes pour la classification des signes en temps réel.

Enfin, sa capacité à effectuer des prédictions en temps réel garantit une reconnaissance rapide des signes, essentielle pour une expérience utilisateur intuitive et sans latence avec les lunettes intelligentes.

Nous avons aussi exploré l'utilisation des **modèles préentraînés** qui peuvent apporter des bénéfices supplémentaires en termes de performance et de précision.

Tout d'abord, en utilisant des modèles préentraînés, on bénéficie de poids déjà bien ajustés grâce à un entraînement sur des données très variées. Cela permet de ne pas repartir de zéro, comme lorsque l'on crée un modèle depuis le début. Les poids préexistants sont déjà optimisés pour reconnaître des éléments généraux, comme des formes ou des contours, ce qui aide notre modèle à être plus précis dès le départ. En utilisant ces modèles, nous gagnons du temps et des ressources, car l'apprentissage sur notre propre jeu de données est plus rapide et plus efficace. Cela nous permet d'avoir un modèle qui fonctionne mieux tout en réduisant les coûts de calcul et de temps.

Enfin, l'utilisation de modèles préentraînés nous permet de réduire l'impact sur les ressources limitées du Raspberry Pi. Avec des modèles comme **MobileNetV2**, **EfficientNetB0** et **EfficientNetV2s**, qui sont à la fois légers et performants, nous pouvons réaliser des prédictions rapides et précises sans surcharger le système. Cela est essentiel, car le Raspberry Pi dispose de ressources limitées en termes de puissance de calcul et de mémoire.

3.3.3 Implémentation de MLP

a) Prétraitement des données et extraction des caractéristiques :

La première étape pour entraîner notre perceptron multicouche (MLP) pour la reconnaissance des signes consiste à préparer les données nécessaires. Comme mentionné précédemment, notre jeu de données est composé de 28 classes, et chaque classe contient 920 images stockées dans une structure de répertoires où chaque dossier représente une classe spécifique de l'alphabet,

ainsi qu'une classe pour l'espace (”space”) et une autre pour la suppression (”delete”).

Nous avons utilisé MediaPipe Hands pour extraire les caractéristiques qui nous intéressent. Dès qu'on dispose d'un jeu de données statique composé d'images, on active 'static_image_mode' afin que chaque image soit traitée indépendamment. De plus, nous avons choisi un taux de confiance minimal relativement élevé (0.9) pour garantir que seules les détections les plus précises soient retenues, atténuant ainsi les détections erronées et affinant la fiabilité des données recueillies.

Un parcours a été fait avec la bibliothèque os pour parcourir toutes les images du dataset. Suite à la récupération de chaque image, celle-ci a été chargée en mémoire via OpenCV (cv2) dans le format BGR. Comme MediaPipe Hands opère en RGB, une conversion de BGR à RGB a été nécessaire. Après conversion, l'image a été analysée par MediaPipe Hands, un système qui identifie et isole les 21 points de référence de la main (voir la figure 3.4), correspondant aux jointures et aux bouts des doigts.

Fig. 3.4 : Visualisation des 21 landmarks de la main détectés par MediaPipe pour deux images de notre jeu de données, appartenant respectivement aux classes 'espace' et 'delete'

Chaque point est déterminé par ses coordonnées (x, y), normalisées entre 0 et 1 afin d'assurer une plus grande uniformité des données. Ainsi, pour chaque image où une main a été identifiée, nous avons créé un vecteur de caractéristiques comprenant 42 valeurs (21 points × 2 coordonnées). Ce vecteur est ensuite associé à une étiquette correspondant à la classe du signe représenté, garantissant une correspondance entre les données et leur signification.

Afin d'assurer l'uniformité et l'excellence du jeu de données, nous avons retenu uniquement les images présentant une détection complètement valide.

Enfin, les vecteurs obtenus, accompagnés de leurs étiquettes de classe, ont été rangés dans un dictionnaire avant d'être sérialisés grâce à Pickle. Le fichier résultant contient les données sous la forme suivante :

Avant le mélange :

- Données[0] : [0.6782, 0.7086, 0.7342, 0.6753, 0.7794, …] → Label : a
- Données[1] : [0.5670, 0.7814, 0.6341, 0.7487, 0.6808, …] → Label : a

- Données[2] : [0.7035, 0.6984, 0.7777, 0.6569, 0.8340, …] → Label : a

Afin d'éliminer tout risque de biais lié à l'organisation des classes, les données ont été mélangées au hasard, puis stockées dans un nouveau fichier Pickle avant d'être exploitées pour l'entraînement :

Après le mélange :

- Données[0] : [0.7061, 0.5208, 0.6362, 0.5417, 0.5433, …] → Label : p
- Données[1] : [0.5783, 0.7640, 0.6178, 0.6892, 0.5803, …] → Label : h
- Données[2] : [0.5801, 0.6345, 0.5731, 0.5717, 0.6063, …] → Label : g

Un mappage doit être réalisé entre les différentes étiquettes uniques (a, b, c, d, delete, ..., z) et des valeurs entières, afin d'assurer une représentation numérique des classes garantissant la compatibilité avec le modèle. Ce mappage est illustré ci-dessous à la figure 3.5 :

```
 Classes AVANT conversion en entiers :
['a', 'b', 'c', 'd', 'delete', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 'space', 't', 'u', 'v', 'w', 'x', 'y', 'z']

 Labels APRÈS conversion en entiers :
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27]
```

Fig. 3.5 : Les classes avant et après conversion en entiers

b) Équilibrage des données :

L'utilisation d'un seuil de confiance élevé (0,9) pour la détection des points clés a conduit à l'exclusion de certaines images, notamment celles floues ou insuffisamment nettes. Bien que le dataset initial ait été équilibré, cette restriction a engendré un déséquilibre dans la répartition des classes, certaines images n'ayant pas été retenues.

Après avoir enregistré les données, nous avons analysé la distribution des classes afin d'évaluer l'impact de cette filtration stricte. Il a été observé que certaines classes avaient un nombre d'échantillons nettement inférieur aux autres (voir la figure 3.6), ce qui pourrait avoir un impact sur la formation du modèle. Cette observation, nous a conduit à considérer des méthodes pour rééquilibrer les données.

Nous avons appliqué le RandomOverSampler afin de corriger le déséquilibre des classes. Cette méthode génère de nouvelles instances pour les classes sous-représentées en dupliquant aléatoirement des échantillons existants jusqu'à atteindre le même nombre d'exemples que la classe majoritaire, qui compte 715 échantillons dans notre cas. L'adoption de cette méthode contribue à équilibrer l'apprentissage du modèle en lui évitant de favoriser excessivement les classes les plus sur-représentées.

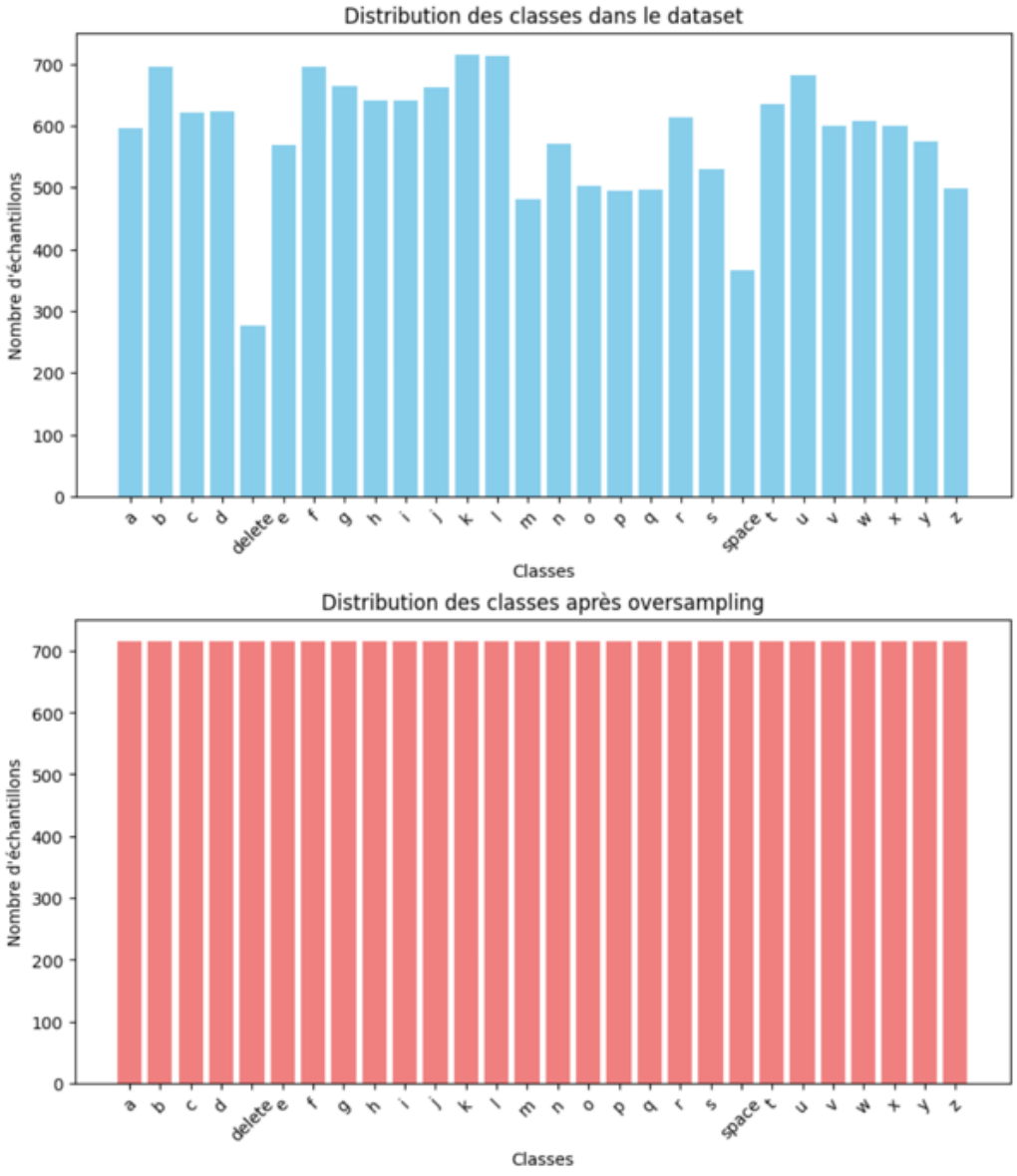

Fig. 3.6 : Répartition des classes avant et après l'oversampling dans notre dataset

Par conséquent, chaque classe contient le même nombre d'échantillons, soit 715, ce qui élimine le déséquilibre initial.

c) Division du jeu de données :

Dans le but de maintenir une distribution équilibrée des classes, les 20 020 échantillons ont été divisés en trois ensembles. L'ensemble d'entraînement comprend 70 % des données (14 014 échantillons), tandis que les 30 % restants ont été divisés de manière égale entre la validation et le test, chacun recevant 3 003 échantillons. L'emploi de la stratification garantit une uniformité des classes au sein de chaque ensemble, comme le montre la figure 3.7 .

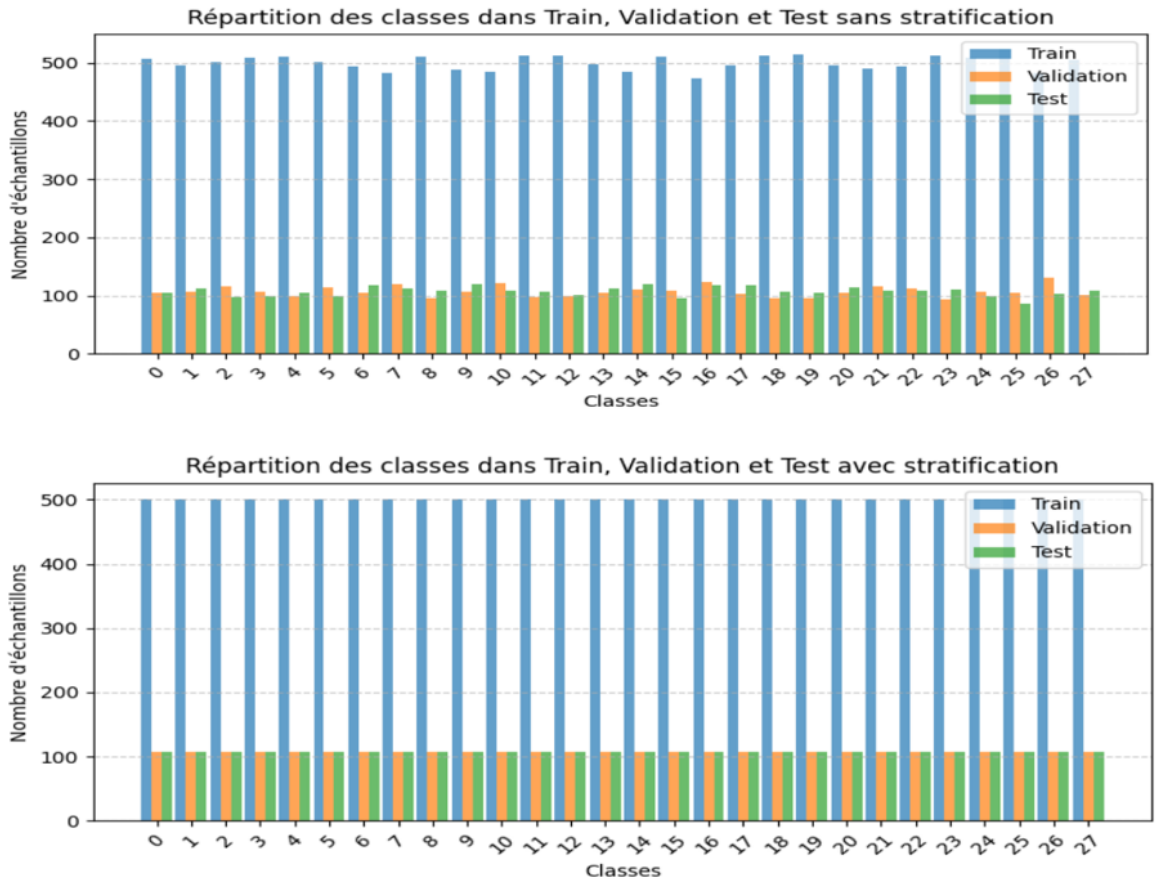

Fig. 3.7 : Distribution des classes de notre jeu de données dans Train, Validation et Test avant et après stratification

d) Modèle et entraînement :

Nous avons conçu un modèle basé sur différentes couches de TensorFlow. Il prend en entrée un vecteur de 42 valeurs correspondant aux coordonnées des 21 points clés extraits des mains. Le modèle est construit autour d'une succession de couches Dense, dont la taille diminue stratégiquement pour filtrer et distiller les informations les plus essentielles.

L'apprentissage commence par une propagation avant (forward pass), où chaque entrée traverse les couches du réseau. Tout d'abord, une couche Dense(256) applique une transformation linéaire sur les données d'entrée, suivie de l'activation ReLU, qui introduit la non-linéarité nécessaire à l'apprentissage de représentations complexes. Cette opération est immédiatement suivie d'une BatchNormalization() pour stabiliser l'apprentissage et d'un Dropout(0.2) afin de limiter le risque de surapprentissage. Ensuite, les données passent successivement à travers trois autres couches Dense de 128, 64 et 32 neurones.

$$z_j = \sum_i w_{ij} \cdot x_i + b_j \tag{3.1}$$

Où :

- z_j est la sortie linéaire du neurone j, calculée en effectuant une somme pondérée des

entrées x_i, suivie de l'ajout du biais b_j,
- w_{ij} sont les poids associés au neurone j et à l'entrée i,
- x_i sont les valeurs d'entrée,
- b_j est le biais associé au neurone j.

Ensuite , la fonction ReLU est appliquée pour renforcer la non-linéarité et capturer des structures plus profondes dans les données.

$$a_j = \text{ReLU}(z_j) = \max(0, z_j) \tag{3.2}$$

Où :

- a_j est l'activation après application de la fonction ReLU.
- z_j est la sortie linéaire du neurone

Enfin, la couche de sortie Dense(28, activation="softmax") va générer un vecteur de probabilités de taille 28, où chaque valeur représente la probabilité que l'entrée appartienne à une classe donnée.

$$\mathbf{p} = [0.03, 0.12, 0.07, \ldots, 0.05, 0.03]$$

Après l'obtention de la prédiction, on calcule l'erreur en comparant le résultat du modèle avec l'étiquette réelle de l'échantillon. Pour cela, la fonction de perte « sparse_categorical_crossentropy » est utilisée afin de mesurer la divergence entre la sortie prédite et la classe correcte.

Ainsi, notre objectif est de minimiser cette erreur autant que possible sur tous les échantillons pour augmenter la précision du modèle. Le modèle est entraîné sur 60 époques avec des lots de 64 échantillons, au cours desquelles ses paramètres sont ajustés progressivement. À chaque cycle d'apprentissage, la rétropropagation des erreurs (backpropagation) est utilisée pour affiner les poids du réseau. Cela implique de calculer le gradient de la fonction de perte par rapport aux poids, puis de les mettre à jour à l'aide de l'optimiseur Adam, en suivant une règle d'adaptation précise. Où un taux d'apprentissage=0.00008, choisi minutieusement pour éviter des fluctuations excessives et un perfectionnement progressif des prédictions.

Tout au long du processus, la métrique de précision est supervisée pour mesurer l'amélioration des performances du modèle au fur et à mesure des itérations. En outre, un ensemble de validation est utilisé simultanément pour évaluer la capacité du modèle à se généraliser sur des données nouvelles et prévenir le surajustement sur les données d'entraînement.

e) Description du pipeline de traitement :

Dans notre pipeline de traitement, chaque image issue du flux vidéo est d'abord capturée et convertie en format RGB, étant donné que MediaPipe nécessite des images RGB. L'option « static_image_mode=False » permet au système d'utiliser le suivi des mains (tracking) plutôt que de redétecter une main à chaque fois, ce qui améliore la réactivité et l'efficacité. Lorsqu'une

main est détectée avec un niveau de confiance suffisant « min_detection_confidence=0.5», ses 21 points clés sont capturés et conservés en tant que coordonnées (x, y), donc pour une main détectée, nous obtenons 42 valeurs au total (21 points × 2 coordonnées). Ces valeurs sont ensuite organisées en un vecteur qui sera transformé en un tableau de type float32 et reformé pour correspondre à l'entrée attendue par le modèle.

Le vecteur est ensuite fourni au modèle TensorFlow Lite pour effectuer une inférence. Les prédictions obtenues sont analysées pour identifier la classe ayant la probabilité la plus élevée, à condition que cette probabilité dépasse un seuil de confiance suffisant. Enfin, la prédiction finale est affichée à l'écran sous forme de texte et les points clés de la main sont dessinés sur l'image pour permettre à l'utilisateur de visualiser les résultats en temps réel.

3.3.4 Implémentation des modèles pré-entraînés

a) Prétraitement des données :

Dans le cadre de l'implémentation des modèles pré-entraînés MobileNetV2, EfficientNetB0 et EfficientNetV2S, une approche de prétraitement rigoureuse a été mise en place pour satisfaire les exigences particulières de chaque structure. Les images, préalablement structurées en ensembles d'entraînement (80%), de validation (10%) et de test (10%) sont chargées à l'aide de la méthode « flow_from_directory() » de TensorFlow, associée à l'outil *ImageDataGenerator*. Ce dernier facilite l'application dynamique des règles de transformation spécifiées. Chaque image est convertie à une dimension fixe de 224 pixels de largeur et de hauteur, puis regroupée en lots de 16 pour une gestion efficace des ressources. Pour MobileNetV2, la fonction « preprocess_input » est appliquée afin de normaliser les pixels dans l'intervalle [-1, 1], conformément aux exigences du modèle. En revanche, pour EfficientNetB0, une couche *Rescaling* intégrée au modèle convertit automatiquement les pixels d'entrée du format [0, 255] vers [0, 1] en les divisant par 255. Pour EfficientNetV2S, la couche *Rescaling* intégrée effectue une normalisation vers l'intervalle [-1, 1] en appliquant la transformation (x / 128) - 1.

b) Augmentation de données :

Pour renforcer la capacité de généralisation des modèles, l'accroissement des données est exclusivement mis en œuvre sur le jeu d'entraînement. Cette augmentation comprend des déplacements horizontaux et verticaux, des transformations de cisaillement, des zooms aléatoires et des inversions horizontales, ce qui permet de reproduire différentes situations de capture d'image. Il est néanmoins crucial de préciser que les transformations mises en œuvre ne produisent pas de variations assez majeures pour imiter des conditions extrêmes. Par conséquent, ces modifications demeurent assez fidèles à la réalité visuelle pour éviter déformations drastiques.

Pour assurer une bonne évaluation, aucune transformation aléatoire n'est appliquée aux ensembles de test et de validation : Uniquement la procédure de prétraitement spécifique au modèle est mise en œuvre sur ces ensembles.

c) Entraînement :

Pour l'entraînement des modèles MobileNetV2, EfficientNetB0 et EfficientNetV2S, une tête de réseau personnalisée a été intégrée à chaque architecture afin d'ajuster la sortie des caractéristiques extraites à la tâche spécifique de reconnaissance de la langue des signes. Les modèles ayant été pré-entraînés sur ImageNet contiennent 1 000 classes, c'est pour ça qu' il est nécessaire de modifier la tête du modèle pour l'adapter au nombre réduit et spécifique de classes de notre tâche. Notre tête personnalisée débute par une couche *GlobalAveragePooling2D*, qui prend chaque canal de l'image traitée et calcule la moyenne de toutes ses valeurs, ce qui permet de transformer un grand volume de données en une forme plus compacte, facile à traiter, tout en gardant l'information importante et en réduisant le nombre de calculs à faire.

Exemple de *GlobalAveragePooling2D* : Supposons que l'on dispose d'une image de taille 3×3 avec 2 canaux :

Canal 1 :

$$\begin{bmatrix} 8 & 1 & 0 \\ 8 & 0 & 3 \\ 7 & 3 & 1 \end{bmatrix}$$

Addition des éléments : $8 + 1 + 0 + 8 + 0 + 3 + 7 + 3 + 1 = 31$
Moyenne : $\frac{31}{9} \approx 3.44$

Canal 2 :

$$\begin{bmatrix} 0 & 5 & 13 \\ 12 & 10 & 0 \\ 1 & 0 & 5 \end{bmatrix}$$

Addition des éléments : $0 + 5 + 13 + 12 + 10 + 0 + 1 + 0 + 5 = 46$
Moyenne : $\frac{46}{9} \approx 5.11$

Donc, le résultat obtenu avec *GlobalAveragePooling2D* est le vecteur : $[3.44,\ 5.11]$

Ensuite, une couche Dense de 256 neurones, activée par ReLU, est ajoutée, suivie d'une couche Dropout avec un taux de 0.4 pour éviter le surapprentissage. Enfin, une couche Dense de sortie, avec activation Softmax, génère une probabilité pour chaque classe, le nombre de neurones étant égal au nombre de classes à prédire.

1) Première phase :

Au début de l'entraînement, nous avons opté pour la congélation de toutes les couches des modèles pré-entraînés, cela nous a permis de n'actualiser que les poids de la tête personnalisée des modèles, en exploitant les caractéristiques visuelles générales déjà apprises par ces modèles sur ImageNet, sans altérer leur capacité à extraire des informations fondamentales telles que les bords, les textures et les formes élémentaires. Les modèles pré-entraînés fonctionnent en apprenant des représentations hiérarchiques des images : les couches initiales détectent des ca-

ractéristiques simples comme les bords, tandis que les couches plus profondes identifient des structures plus complexes, comme les motifs .

En conservant les couches du modèle de base gelées, l'objectif était de concentrer l'apprentissage sur la partie personnalisée des modèles dédiée à la tâche spécifique de la reconnaissance de la langue des signes. Nous avons ajusté les taux d'apprentissage pour chaque modèle : 0.0005 pour MobileNetV2, 0.0007 pour EfficientNetB0 et 0.0007 pour EfficientNetV2S, afin de favoriser une adaptation progressive. De plus, la technique de réduction du taux d'apprentissage (ReduceLROnPlateau) a été mise en œuvre pour adapter de manière dynamique le taux d'apprentissage en fonction des résultats du modèle sur l'ensemble de validation. Cette phase d'entraînement a duré 4 epochs pour MobileNetV2, 3 epochs pour EfficientNetB0, et 4 epochs pour EfficientNetV2S.

Cette phase permet une première adaptation à la tâche, facilitant ainsi l'optimisation pour la tâche de classification des signes.

2) Deuxième phase :

Lors de la deuxième phase de l'entraînement, appelée fine-tuning, certaines couches du modèle pré-entraîné sont dégelées pour permettre un ajustement plus précis. Ainsi, pour MobileNetV2, les 29 dernières couches ont été dégelées, pour EfficientNetB0, les 75 dernières couches, et pour EfficientNetV2S, les 67 dernières couches. Ces couches, responsables de l'extraction de caractéristiques spécifiques, sont réajustées pour mieux s'adapter à notre jeu de données, tout en préservant les informations générales acquises lors de l'entraînement initial sur ImageNet.

Pour cette phase, un taux d'apprentissage plus faible de 0.00007 pour MobileNetV2, 0.0001 pour EfficientNetB0 et EfficientNetV2S a été utilisé avec la stratégie ReduceLROnPlateau, pour ne pas perturber complètement les poids pré-entraînés et pour assurer la stabilité de l'entraînement, mais permettre une légère modification afin de mieux s'ajuster aux nouvelles données. Cette phase a été réalisée sur 8 epochs pour MobileNetV2, 12 epochs pour EfficientNetB0, et 9 epochs pour EfficientNetV2S.

Dans les deux phases, nous avons utilisé l'optimiseur Adam, la fonction de perte categorical_crossentropy, et la métrique d'évaluation choisie est l'accuracy.

d) Description du pipeline de traitement :

Dans ce pipeline de traitement, chaque image capturée à partir du flux vidéo est d'abord convertie en format RGB. Ensuite, l'image est analysée par MediaPipe Hands pour détecter les points clés de la main. Lorsque la main est détectée, les coordonnées des 21 points clés sont extraites et utilisées pour calculer un rectangle de délimitation (bounding box) autour de la main (voir la figure 3.8) . Ce rectangle est agrandi par un padding de 40 pixels pour mieux capturer la main.

Une fois la région de la main découpée, l'image est redimensionnée à 224x224 pixels, convertie en un tableau NumPy de type float32 et prétraitée selon les exigences spécifiques de chaque modèle : la fonction preprocess_input() est appliquée pour MobileNetV2, tandis que pour EfficientNetB0 et EfficientNetV2S, le prétraitement est automatiquement assuré par la couche Rescaling intégrée au modèle. Une dimension de lot est ensuite ajoutée pour correspondre au format d'entrée attendu par le modèle.

Le modèle chargé est alors utilisé pour prédire la classe de la main en se basant sur l'image prétraitée. La classe prédite et son niveau de confiance sont affichés sur l'image, accompagnés d'un rectangle dessiné autour de la main détectée.

Fig. 3.8 : Prédiction en temps réel avec EfficientNetB0 : détection de la main, cadrage et affichage du niveau de confiance.

Cette approche a permis d'obtenir de meilleurs résultats pour tous les modèles utilisés, car au lieu d'effectuer des prédictions sur l'ensemble du cadre vidéo, celles-ci sont réalisées uniquement sur l'image de la main, ce qui garantit une meilleure précision.

3.4 Expérimentation et résultats

3.4.1 Présentation des résultats par modèle

Dans cette partie, nous présentons les résultats obtenus pour chaque modèle testé, en affichant les courbes d'entraînement (accuracy, validation accuracy, loss, validation loss) ainsi que la matrice de confusion correspondante.

3.4.1.1 Multilayer Perceptron (MLP)

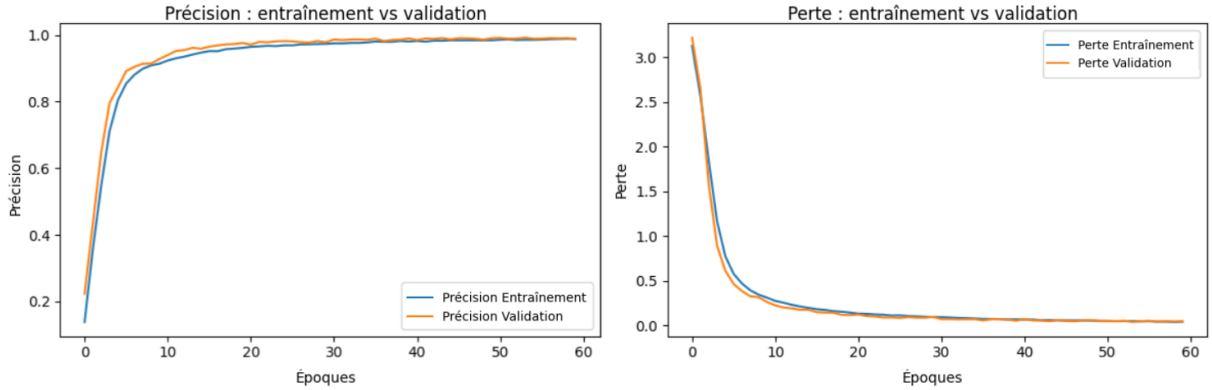

Fig. 3.9 : Courbes de précision et de perte, en entraînement et en validation, pour le Multilayer Perceptron (MLP).

La figure 3.9 montre clairement une évolution très positive des performances du Multi-Layer Perceptron au fil de 60 époques. L'accuracy d'entraînement et de validation atteint rapidement un niveau élevé et se stabilise autour de 0.98, indiquant une excellente qualité de prédiction. De même, la perte diminue régulièrement et reste basse sur les deux ensembles, sans signe de surapprentissage. Le modèle semble donc bien optimisé et bien équilibré.

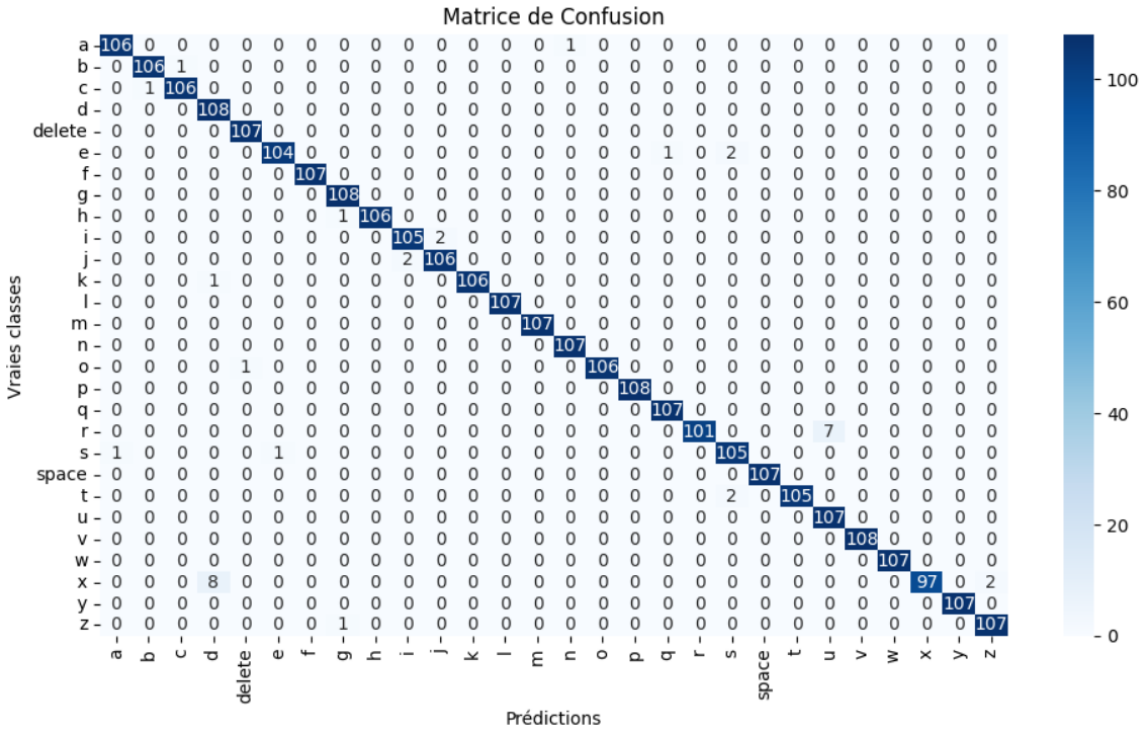

Fig. 3.10 : Matrice de confusion pour le Multilayer Perceptron (MLP).

Dans la figure 3.10, la matrice de confusion montre une forte précision globale, avec des valeurs sur la diagonale, indiquant que les prédictions correspondent bien aux classes réelles. Quelques erreurs mineures sont visibles, notamment pour les classes ”r”, et ”x”.

3.4.1.2 EfficientNet B0

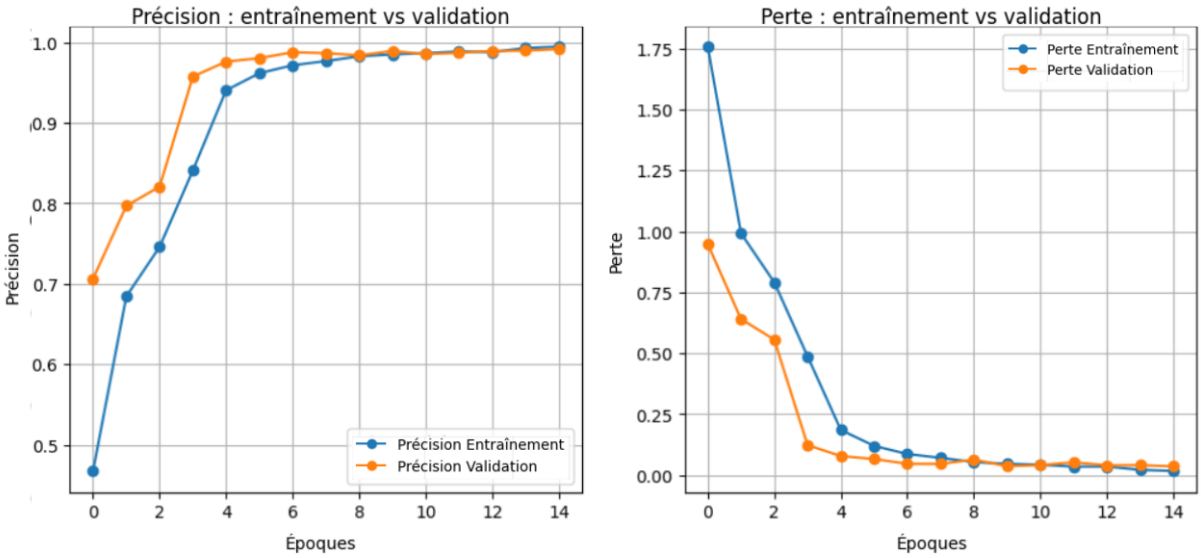

Fig. 3.11 : Courbes de précision et de perte, en entraînement et en validation, pour le modèle EfficientNet B0.

L'analyse des courbes affichées dans la figure 3.11 révèle une évolution remarquable des performances du modèle EfficientNetB0 au fil de 15 époques. L'exactitude sur les ensembles d'entraînement et de validation atteint rapidement un plateau élevé, avoisinant les 0.99. Parallèlement, la perte diminue fortement puis se stabilise à un niveau très bas. Ce comportement indique que le modèle a été bien entraîné, avec une convergence rapide et une excellente capacité de généralisation.

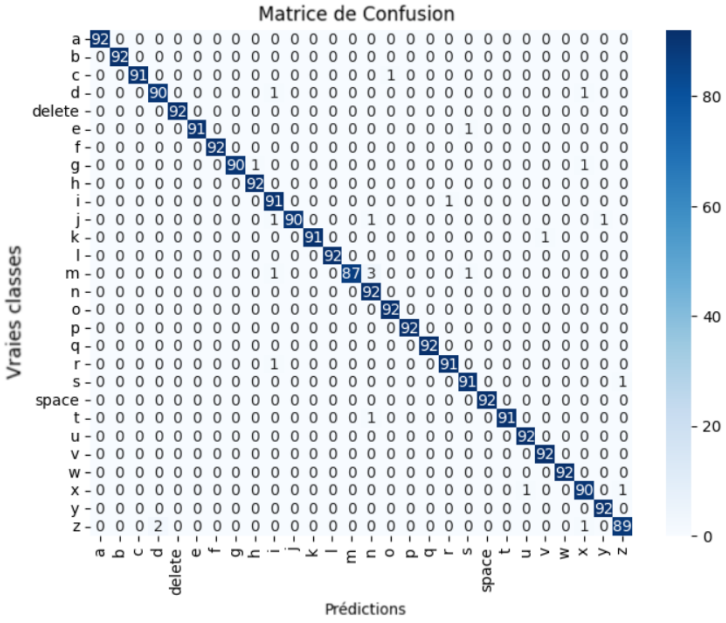

Fig. 3.12 : Matrice de confusion pour EfficientNet B0.

La figure 3.12 précédente, illustrant la matrice de confusion affiche une excellente précision globale, avec des scores diagonaux majoritairement, reflétant une classification fiable par EfficientNetB0. Des erreurs légères apparaissent hors diagonale, surtout pour ”m” (87) et ”z” (89).

3.4.1.3 EfficientNet V2S

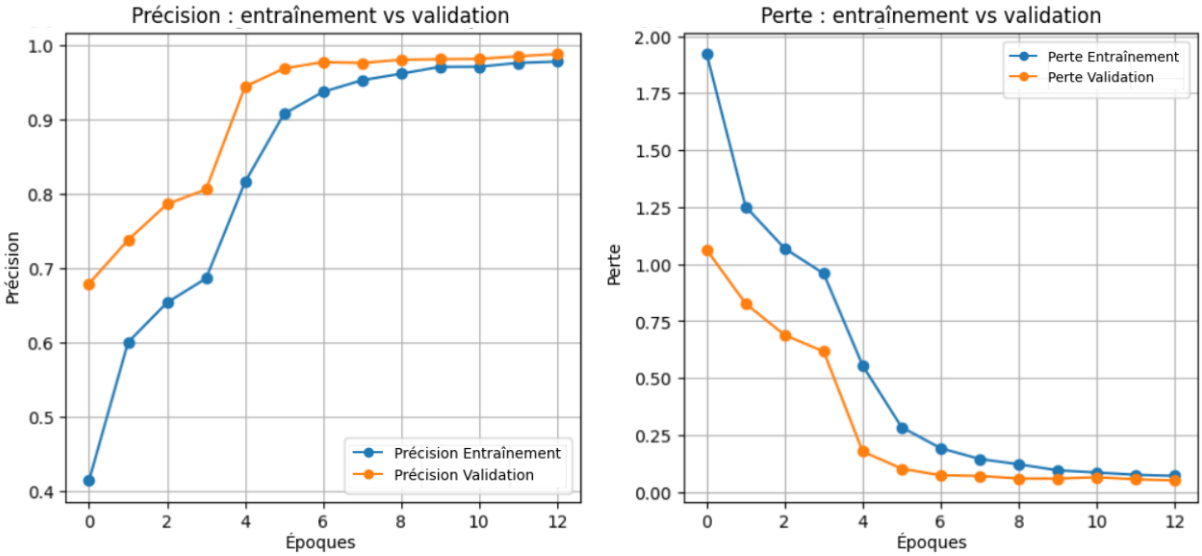

Fig. 3.13 : Courbes de précision et de perte, en entraînement et en validation, pour le modèle EfficientNet V2S.

On observe à travers la figure 3.13 une nette amélioration des performances du modèle EfficientNetV2S au fil des 13 époques. La précision d’entraînement et de validation augmente rapidement pour atteindre des valeurs près de 0.98, tandis que les pertes diminuent fortement avant de se stabiliser à un niveau très bas. L’absence d’écart notable entre les courbes d’entraînement et de validation indique une bonne généralisation sans surapprentissage. Le modèle semble donc bien entraîné, stable et performant.

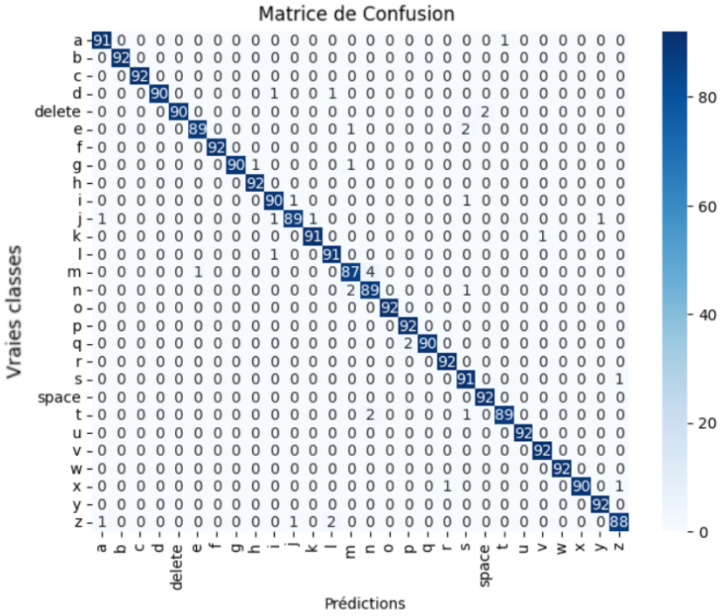

Fig. 3.14 : Matrice de confusion pour EfficientNet V2S.

Comme le montre la figure 3.14 précédente, la matrice de confusion illustre une bonne précision globale, avec des résultats diagonaux de 87 à 92 pour EfficientNetV2s. Les classes ”m” (87) et ”z” (88) montrent des faiblesses par rapport aux autres classes.

3.4.1.4 MobileNet V2

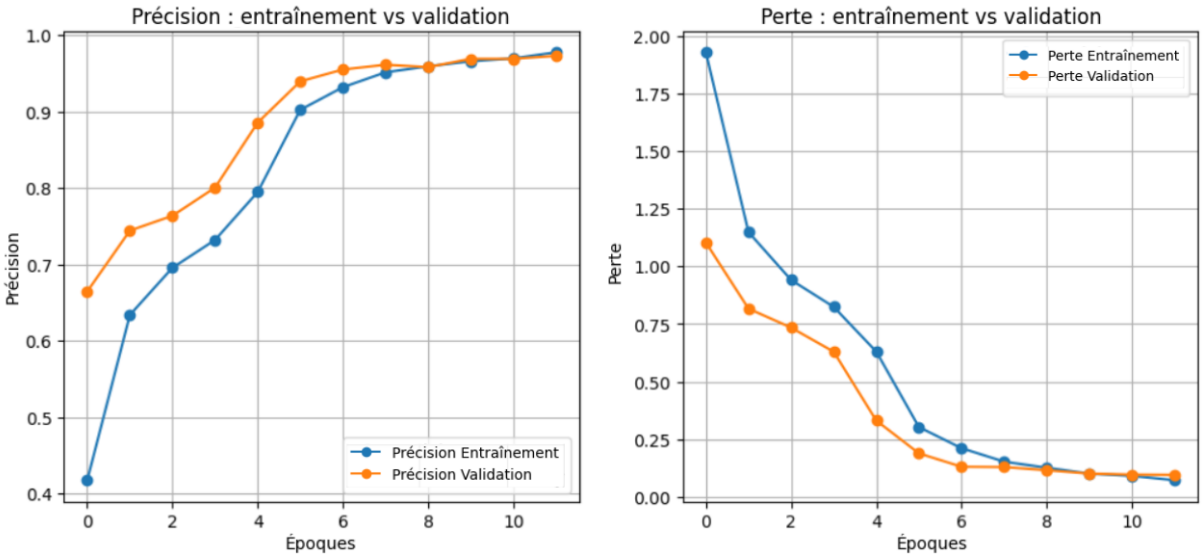

Fig. 3.15 : Courbes de précision et de perte, en entraînement et en validation, pour le modèle MobileNetV2.

La figure 3.15 met en évidence une progression claire et régulière des performances du modèle MobileNetV2 au cours des 12 époques d’entraînement. La précision, tant sur l’ensemble d’entraînement que sur l’ensemble de validation, augmente pour atteindre des valeurs proches

de 0.97. En parallèle, la fonction de perte diminue fortement sur les deux ensembles et converge vers des valeurs très faibles. Il y a une évolution parallèle des courbes d'entraînement et de validation, sans signe notable de surapprentissage.

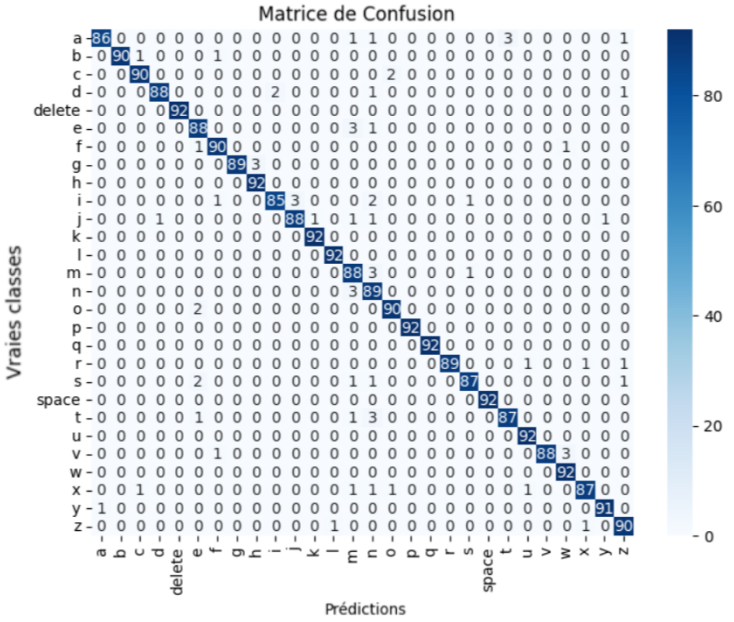

Fig. 3.16 : Matrice de confusion pour MobileNet V2.

D'après la figure 3.16 précédente, représentant la matrice de confusion de MobileNetV2, la précision apparaît inconstante, avec des scores diagonaux allant de 85 à 92. Plusieurs classes présentent des performances plus faibles, telles que "i" (85), "a" (86) et "t" (87). La précision générale reste acceptable, mais laisse une marge d'amélioration.

3.4.2 Analyse comparative des modèles

3.4.2.1 Les courbes d'entraînement

Les courbes d'accuracy et de loss pour les modèles MLP, EfficientNetB0, EfficientNetV2-S et MobileNetV2 montrent des tendances similaires mais avec des différences notables. EfficientNetB0 affiche la meilleure performance avec la précision de validation la plus élevée (99 %) et la perte de validation la plus faible, suivie de près par le MLP (98%) et EfficientNetV2-S (98 %), tandis que MobileNetV2 est légèrement en retrait (97%).

Les courbes d'accuracy augmentent rapidement au début, puis se stabilisent, surtout pour EfficientNetB0 et MLP, qui convergent plus vite vers des valeurs élevées. Dans les dernières époques, on remarque de petites fluctuations, notamment pour MobileNetV2.

Les courbes de loss diminuent régulièrement, avec EfficientNetB0 et MLP atteignant les valeurs les plus basses. Globalement, EfficientNetB0 domine en précision, tandis que le MLP se distingue par sa rapidité .

Modèle	Train Accuracy	Loss	Validation Accuracy	Validation Loss
MLP	98%	4%	98%	4%
EfficientNet B0	99%	1%	99%	3%
EfficientNet V2S	97%	7%	98%	5%
MobileNet V2	97%	7%	97%	9%

Tab. 3.2 : Tableau comparatif des modèles en termes d'accuracy, de loss, de validation accuracy, et de validation loss.

3.4.2.2 Les matrices de confusions

Afin de mieux visualiser les erreurs de classification identifiées dans les matrices de confusion, nous avons résumé les principales lettres confondues pour chaque modèle dans le tableau 3.3.

Modèle	Lettres confondues
MLP	x ↔ d, r ↔ u
EfficientNetB0	m ↔ n, z ↔ d
EfficientNetV2-S	e ↔ s, m ↔ n, t ↔ n, z ↔ l
MobileNetV2	a ↔ t, d ↔ i, e ↔ m, g ↔ h, i ↔ j, m ↔ n, s ↔ e, t ↔ n, v ↔ w

Tab. 3.3 : Principales lettres confondues pour chaque modèle selon les matrices de confusion

Le tableau 3.3 met en évidence les principales confusions observées entre certaines lettres pour chaque modèle, en se basant sur les matrices de confusion. On constate que les modèles MLP et EfficientNetB0 présentent très peu d'ambiguïtés. Pour le MLP, les confusions se limitent principalement à x et d, ainsi que r et u, tandis qu'EfficientNetB0 confond essentiellement m et n ainsi que z et d. Ces résultats témoignent de leur bonne capacité de généralisation.

En revanche, les autres modèles : EfficientNetV2-S et MobileNetV2 montrent un nombre plus important de lettres mal classées. Par exemple, EfficientNetV2-S confond des lettres visuellement proches comme m et n, t et n, ou encore e et s. Quant à MobileNetV2, il présente un taux de confusion plus élevé, touchant un plus grand nombre de lettres, ce qui explique sa performance légèrement inférieure en temps réel.

Les confusions relevées dans les résultats montrent que la ressemblance visuelle entre certains signes contribue fortement aux erreurs de classification. Cette tendance semble commune à tous les modèles, ce qui suggère que ces erreurs proviennent davantage des similitudes gestuelles entre lettres que des limites propres aux modèles .

3.4.2.3 Comparaison globale des modèles

Dans le tableau suivant une comparaison basée sur plusieurs critères clés tels que le temps d'entraînement, la taille du modèle et le temps d'inférence.

Critère	MLP	MobileNet V2	EfficientNet B0	EfficientNet V2S
Nombre total d'epochs	60	12	15	13
Temps d'entraînement	$\sim$ 1m	$\sim$ 2h	$\sim$ 4h	$\sim$ 8h
Taille	221 Ko	5 Mo	8 Mo	40 Mo
Accuracy sur l'ensemble de test	98%	97%	99%	98%
Précision	99%	97%	99%	99%
Recall	99%	97%	99%	99%
F1-score	99%	97%	99%	99%
Temps d'inférence	$\sim$ 25 ms	$\sim$ 60 ms	$\sim$ 70 ms	$\sim$ 130 ms
Nombre de classes correctement reconnues distance proche de la caméra	27/28	26/28	27/28	26/28
Nombre de classes correctement reconnues à une distance de 1,5 mètre avec un arrière-plan noir	18/28	24/28	26/28	24/28
Nombre de classes correctement reconnues à une distance de 1,5 mètre avec un arrière-plan complexe	18/28	23/28	25/28	23/28

Tab. 3.4 : Tableau comparatif des modèles selon plusieurs critères

D'après l'analyse des critères figurant dans le tableau 3.4, le MLP, très léger (221 Ko) et rapide à entraîner (environ 1 minute), présente un Recall et un F1-score remarquables de 99 %, avec une excellente reconnaissance en temps réel à courte distance (27 classes sur 28 correctement reconnues). Cependant, à une distance de 1,5 mètre avec un fond noir ou un fond complexe, ses performances diminuent, reconnaissant correctement seulement 18 classes sur 28.

MobileNetV2, avec un temps d'entraînement d'environ 2 heures, reconnaissant correctement 26 classes sur 28 à courte distance, et conserve une bonne reconnaissance à 1,5 mètre, avec 24 classes détectées sur fond noir et 23 classes sur fond complexe. Cependant, lors des tests en temps réel, MobileNetV2 a montré une instabilité dans ses prédictions, notamment lorsqu'un léger mouvement de la main est détecté. Cette sensibilité aux variations mineures de position rend les prédictions peu stables en conditions dynamiques, ce qui peut compromettre la fiabilité de la reconnaissance dans un usage réel.

EfficientNetB0 propose un bon compromis entre performance et temps d'entraînement (environ 4 heures). Il atteint une accuracy et un F1-score de 99 % et reconnaît 27 classes sur 28 à courte distance. De plus, il conserve de bonnes performances à 1,5 mètre, détectant correctement 26 classes avec un fond noir et 25 classes avec un fond complexe.

Enfin, EfficientNetV2S, malgré ses performances élevées (99 % de F1-score), nécessite un

temps d'entraînement plus long (environ 8 heures) et un temps d'inférence plus élevé. Sa reconnaissance à 1,5 mètre est similaire à MobileNetV2, avec 24 classes détectées sur fond noir et 23 sur fond complexe.

3.4.3 Choix du modèle pour le prototype

Dans un premier temps, le Multi-Layer Perceptron (MLP) avait été retenu pour sa légèreté et sa rapidité d'inférence, des qualités bien adaptées aux contraintes matérielles du Raspberry Pi. Toutefois, les tests réalisés à une distance réaliste d'environ 1,5 mètre ont mis en évidence ses limites, avec une diminution marquée du nombre de classes correctement reconnues.

Ces résultats nous ont amenés à réévaluer notre choix de modèle. Parmi l'ensemble des architectures testées, EfficientNet B0 s'est distingué par sa stabilité et sa robustesse face à l'éloignement de la caméra et aux variations de l'arrière-plan. Il a maintenu un bon niveau de reconnaissance même à distance, ce qui constitue un critère essentiel pour une utilisation en conditions réelles.

C'est donc cette capacité d'EfficientNet B0 à maintenir des performances fiables malgré des conditions moins idéales qui a motivé son intégration dans le prototype final. Il représente un bon compromis entre précision et temps d'inférence, ce qui le rend adapté aux contraintes des systèmes embarqués, tout en garantissant une reconnaissance efficace dans des situations variées.

3.5 Fonctionnement de prototype

Le prototype permet une communication fluide et autonome entre une personne sourde ou muette et son interlocuteur grâce à une séquence interactive bien définie. Lorsque l'utilisateur souhaite initier une communication, il appuie sur un bouton relié à un Arduino, qui déclenche l'envoi d'un signal infrarouge via une LED IR. Ce signal est capté par un récepteur infrarouge intégré dans les lunettes intelligentes. Dès la détection du signal, le système émet une alarme sonore à travers des écouteurs et affiche sur l'écran LCD la phrase *Someone wants to talk* , alertant ainsi l'interlocuteur, comme illustré à la figure 3.17.

Fig. 3.17 : Écran LCD

Ensuite, la caméra Pi (module 3), fixée à l'avant des lunettes, commence à capturer des images de la main de l'utilisateur. Ces images sont traitées en temps réel par la bibliothèque

MediaPipe qui détecte les points clés (*landmarks*) de la main. Lorsqu'une main est détectée, une zone englobante autour de la main est extraite avec une marge, puis lissée par un filtre exponentiel afin d'éviter les variations brusques dues aux mouvements. Cette région est redimensionnée à 224x224, convertie en tableau float32, puis transmise à un modèle TensorFlow Lite EfficientNet B0 basé sur un réseau de neurones convolutif.

Le modèle prédit le signe correspondant, parmi un ensemble de lettres et de commandes spéciales comme *delete* ou *space*. Si le geste reconnu est une lettre, celle-ci est ajoutée au mot en cours de composition ; un délai de 1 seconde est imposé entre deux prédictions pour éviter les doublons. Si le geste détecté est *space*, le mot formé est ajouté à une phrase, qui est ensuite affichée sur l'écran LCD et prononcée à l'aide de *espeak* dans un thread séparé pour ne pas bloquer l'exécution du programme.Si le signe détecté est *delete*, la dernière lettre saisie est supprimée du mot en cours, permettant à l'utilisateur de corriger instantanément ses erreurs sans recommencer toute la séquence.

Le texte complet est affiché sur un écran LCD I2C de 16x2 caractères, où seule la fin de la phrase et du mot courant est visible. Enfin, si aucune main n'est détectée ou qu'aucun changement ne survient pendant 5 secondes, le système considère que la communication est terminée : l'écran LCD est automatiquement vidé, la phrase est effacée de la mémoire du programme, et le système reste actif, prêt à détecter un nouveau geste sans nécessiter de nouveau signal IR. Ce scénario garantit une interaction fluide, discrète et intuitive, facilitant la communication des personnes sourdes ou muettes dans leur vie quotidienne.

3.6 performance

Dans cette section, nous présentons les performances du modèle EfficientNet B0 intégré dans notre prototype fonctionnant sur Raspberry Pi 4. L'objectif est d'évaluer la robustesse du modèle dans différentes conditions de capture des gestes, en tenant compte des variations de luminosité, distance, main utilisée et présence d'accessoires. Le tableau ci-dessous récapitule le nombre de classes bien reconnues sur un total de 28 classes testées pour chaque condition.

Critère de capture des gestes	**Nombre de classes bien classées (sur 28)**
Luminosité normal à 1 mètre	26/28
Luminosité normal à 1,5 mètre	26/28
Luminosité faible à 1 mètre	27/28
Luminosité faible à 1,5 mètre	27/28
Main droite	26/28
Main gauche	27/28
Avec accessoires (bagues, bracelets)	26/28

Tab. 3.5 : Résultats de classification du modèle EfficientNet B0 selon différentes conditions de capture des gestes

Le tableau 3.5 montre que le modèle **EfficientNet B0** conserve une précision élevée dans la majorité des scénarios de capture :

- **Luminosité normale** (à 1 mètre et 1,5 mètre) : le modèle classe correctement 26 sur 28 gestes, indiquant une bonne stabilité en conditions d'éclairage standard.
- **Luminosité faible** : les performances restent satisfaisantes, avec 27 classes bien reconnues sur 28, ce qui reflète une bonne capacité de généralisation même dans des environnements peu éclairés.
- **Main droite vs. main gauche** : une meilleure performance est obtenue avec la main gauche (27/28) par rapport à la main droite (26/28), ce qui pourrait être dû à une différence dans la base de données utilisée pour l'entraînement ou à des biais de capture.
- **Accessoires (bagues, bracelets)** : le modèle maintient une bonne précision (26/28), ce qui montre une résilience partielle aux éléments perturbateurs visuels.

3.7 Outils utilisés

L'entraînement du modèle Multi-Layer Perceptron (MLP) ainsi que des trois modèles préentraînés —MobileNetV2, EfficientNetB0 et EfficientNetV2S — a été réalisé localement sur un ordinateur portable Lenovo Flex 5, équipé d'un processeur AMD Ryzen 5, de 16 Go de mémoire RAM et d'un GPU intégré AMD Radeon Graphics, à l'aide de Visual Studio Code (VS Code), un environnement de développement léger et polyvalent.

Le jeu de données exploité dans ce projet a été principalement collecté depuis la plateforme Kaggle, reconnue pour sa vaste collection de ressources publiques destinées à l'entraînement et à l'évaluation de modèles d'apprentissage automatique. Il est important de noter que le dataset final a été assemblé manuellement à partir de plusieurs sources, comme décrit dans la sous-section 3.3.1

Pour en garantir l'accessibilité, il a été stocké sur Google Drive. L'extraction des caractéristiques à partir des images a ensuite été réalisée sur Google Colab, en tirant parti de la puissance de calcul des GPU offerts par cet environnement.

Sur un ordinateur HP EliteBook 840 G3 avec 8 Go de Ram, l'application RealVNC Viewer a permis d'accéder à distance à l'interface du Raspberry Pi, facilitant ainsi le déploiement et les tests du système. Par ailleurs, Arduino IDE a été utilisé pour la programmation et les tests de composants électroniques intégrés dans le système, facilitant la communication avec différents modules.

3.8 Conclusion

Ce chapitre a présenté l'ensemble des résultats obtenus à travers le développement, l'entraînement et le déploiement de notre prototype de lunettes intelligentes pour la reconnaissance du langage des signes. Il commence par une présentation générale du prototype, suivie d'une explication détaillée de la méthodologie adoptée tout au long du projet. Nous y avons décrit le dataset utilisé, ainsi que les algorithmes de classification testés. Les outils technologiques ayant facilité

le développement ont également été exposés.

Par la suite, une série d'expérimentations a été menée afin d'évaluer les performances des différents modèles. Les résultats obtenus ont été analysés individuellement, puis comparés de manière globale afin d'identifier le modèle le plus performant. Bien que tous aient affiché des niveaux de performance intéressants, le modèle EfficientNet B0 a été retenu pour le prototype final en raison de sa robustesse et de sa capacité à rester fiable même en conditions non optimales. Il a ainsi permis d'atteindre un taux de précision notable de 99%.

Le système final a démontré une stabilité dans des conditions réelles variées, assurant une reconnaissance fiable et une réponse rapide lors des tests sur le terrain.

Chapitre 4

Test du prototype sur le terrain : école pour enfants handicapés auditif

4.1 Introduction

Afin de valider l'efficacité et l'utilisabilité de notre prototype, il était essentiel de le tester en situation réelle. En effet, un dispositif conçu pour faciliter la communication entre personnes entendantes et personnes sourdes ou muettes doit être évalué non seulement en conditions contrôlées, mais aussi dans un environnement réel, avec de vrais utilisateurs.

Dans ce chapitre, nous présentons le stage réalisé dans une école spécialisée accueillant des enfants en situation de handicap auditif. Ce stage avait pour objectif principal d'observer l'utilisation du prototype dans un cadre éducatif réel, d'en analyser les réactions des utilisateurs, d'identifier les éventuelles difficultés rencontrées et de recueillir des pistes concrètes d'amélioration.

4.2 Contexte du stage

Dans le cadre de notre projet de fin d'études en intelligence artificielle, nous avons effectué un stage d'expérimentation du 25 mai au 29 mai 2025 à l'**École pour Enfants Handicapés Auditifs Slimani Bouhafce**, située à **Riat El Kébir, Mansourah, Tlemcen**. Cette école accueille des enfants et des adolescents présentant une déficience auditive, ré- partis par niveaux scolaires. Elle a pour mission principale de garantir l'accès à une éducation de qualité à ces élèves, en adaptant les méthodes pédagogiques à leurs besoins spécifiques.

Le stage a été encadré sur place par **Madame Souhila Ben Ahmed** et s'inscrit dans la phase d'évaluation de notre prototype de lunettes intelligentes pour la reconaissance de la langue des signes.

Cette immersion nous a également permis de mieux comprendre le contexte éducatif spécifique aux élèves sourds, ainsi que les contraintes pédagogiques et environnementales qu'il faudra prendre en compte dans les perspectives d'évolution du système.

4.3 Objectifs du stage

Le stage avait pour but principal de tester notre prototype dans un environnement réel auprès du public concerné. Il s'agissait :

- d'observer l'interaction des élèves avec le dispositif,
- de recueillir leurs impressions et suggestions,
- d'évaluer l'efficacité de la traduction en temps réel de la langue des signes,
- d'identifier les axes d'amélioration technique ou ergonomique.
- et d'enrichir notre compréhension des besoins spécifiques grâce aux conseils, retours et observations des enseignantes spécialisées.

4.4 Organisation du stage

4.4.1 Tâches effectuées

Durant le stage, plusieurs tâches ont été réalisées pour tester efficacement le prototype auprès des élèves sourds ou muets :

- Présentation du prototype aux enseignants et aux élèves,
- Mise en place de séance de démonstration et d'expérimentation,
- Observation des utilisateurs pendant l'utilisation du prototype,
- Recueil et analyse des retours des élèves et du personnel éducatif,
- Rédaction de notes d'observation et proposition d'améliorations,

4.4.2 Personnes rencontrées

Durant ce stage, plusieurs acteurs ont été impliqués dans les activités :

- Les enseignantes spécialisées dans la prise en charge des élèves sourds, qui ont soutenu les participants tout au long des activités,
- Les élèves participants aux tests du prototype,
- D'autres membres du personnel éducatif et administratif, intéressés par les technologies inclusives.

4.5 Méthodologie des tests

4.5.1 Modalités de test

Les tests ont été réalisés sous forme d'atelier pratique organisé au sein de l'établissement. La séance débutait par une courte démonstration du fonctionnement des lunettes intelligentes, suivie d'un temps d'expérimentation libre par les élèves et enseignantes.

Des observations directes ont été faites pendant l'utilisation, et des échanges individuels ou en petits groupes ont permis de recueillir des commentaires qualitatifs. Certains entretiens informels ont également été menés avec les enseignants pour obtenir leur retour pédagogique.

4.5.2 Public ciblé pour les tests

Les tests ont été menés auprès d'un groupe de **5** enseignantes et **2** étudiants en situation de handicap auditif. Les enseignantes ont été invitées à tester le dispositif et à donner leur avis sur son intérêt pédagogique, tandis que les étudiants ont interagi directement avec le prototype afin d'évaluer sa facilité d'utilisation et son utilité dans un contexte réel.

4.5.3 Outils utilisés pour recueillir les retours

Afin de recueillir les retours des utilisateurs, plusieurs outils ont été mobilisés :

- **Questionnaire simplifié** destiné aux élèves, présenté et expliqué par les enseignantes pour en faciliter la compréhension.
- **Questionnaire standard** destiné aux enseignantes afin de recueillir leur évaluation du dispositif.
- **Entretiens semi-directifs** menés avec les enseignantes pour obtenir un retour approfondi sur l'efficacité et l'utilité du prototype.
- **Prise de notes manuelle** pendant les séances, permettant de consigner les remarques spontanées et les observations directes.

4.6 Observations et résultats des tests

4.6.1 Réactions des élèves et des enseignants

Les premières réactions des élèves ont été très positives. Ils ont montré un grand intérêt pour le prototype, curieux de découvrir comment leurs gestes pouvaient être traduits en texte ou en parole. L'aspect interactif et technologique a suscité leur enthousiasme. Les enseignants, de leur côté, ont salué l'initiative et ont apprécié la démarche inclusive du projet. Ils ont souligné l'importance d'introduire des outils technologiques pour favoriser la communication avec les personnes entendantes.

4.6.2 Points positifs du prototype

Les tests ont permis de mettre en évidence plusieurs aspects positifs du prototype :

- Utilisation simple,
- Bonne reconnaissance des signes de la main dans un environnement bien éclairé,
- Réelle utilité pour des échanges simples avec des personnes non signantes,
- Fort potentiel éducatif selon les enseignants.

4.6.3 Limites ou problèmes rencontrés

Malgré ces points positifs, certaines limites ont été observées durant les tests :

- Difficulté à reconnaître certains signes complexes ou rapides,
- Dépendance à la lumière ambiante pour une détection optimale des gestes,
- Temps de latence perceptible dans certains cas,
- Dispositif encore encombrant pour un usage quotidien,

4.6.4 Propositions d'amélioration

Suite aux tests réalisés, plusieurs pistes d'amélioration ont été identifiées pour optimiser le fonctionnement du prototype :

- **Miniaturisation du matériel** pour rendre les lunettes plus légères et confortables à porter,
- **Optimisation de la caméra** pour qu'elle fonctionne mieux dans des conditions de faible luminosité,
- **Élargissement du vocabulaire** en intégrant des expressions courantes, telles que les couleurs ou les animaux, souvent abordées dans les activités pédagogiques, comme suggéré par les enseignantes.
- **Création d'un mode éducatif** pour que les élèves puissent s'exercer à signer correctement et recevoir un retour immédiat.
- **Intégration de plusieurs langues**, notamment la Langue des Signes Algérienne (LSA), afin de rendre le système plus inclusif et accessible selon le contexte linguistique des utilisateurs.

4.6.5 Bilan personnel du stage

Ce stage a représenté une étape essentielle dans notre parcours, tant sur le plan technique que personnel.

Apports techniques

Ce stage nous a permis de :

- Comprendre les contraintes liées à l'utilisation d'un dispositif dans un environnement réel.
- Recueillir et analyser les retours d'utilisateurs pour améliorer un prototype.
- Identifier les limites de notre prototype, surtout celles liées aux modèles d'intelligence artificielle. Cela nous a motivés à tester d'autres modèles pour améliorer la précision, même quand la personne est loin de la caméra ou quand la luminosité est faible.

Apports humains

Sur le plan humain, cette expérience nous a apporté :

- Une meilleure compréhension des défis quotidiens rencontrés par les personnes sourdes ou muettes .
- Une prise de conscience de l'importance de concevoir des outils réellement inclusifs.
- Un fort sentiment d'utilité et de motivation à poursuivre dans le domaine des technologies au service du handicap.

4.7 Conclusion

Le test du prototype dans un environnement éducatif réel nous a permis de confronter notre solution aux usages concrets et aux besoins spécifiques des utilisateurs finaux. Les retours recueillis, à la fois enthousiastes et constructifs, ont mis en lumière le potentiel réel de notre dispositif, tout en soulignant certains axes d'amélioration technique et ergonomique.

Cette expérience de terrain a renforcé notre conviction quant à l'utilité sociale de notre projet. Elle a également confirmé que l'intégration de technologies d'intelligence artificielle dans des contextes inclusifs peut réellement contribuer à améliorer la communication et l'autonomie des personnes en situation de handicap.

Les observations et les perspectives identifiées à l'issue de ce stage constitueront une base précieuse pour les évolutions futures du prototype.

Conclusion Générale

Dans un monde en constante évolution, l'intelligence artificielle s'impose comme un levier puissant pour relever les défis sociétaux. Parmi eux, l'inclusion des personnes en situation de handicap auditif ou handicap de la parole demeure une priorité. C'est dans cette optique que notre projet a vu le jour, avec pour objectif de développer une solution technologique capable de réduire les barrières de communication entre les personnes sourdes ou muettes et leur entourage.

Le système que nous avons conçu vise à traduire la langue des signes de manière fluide et instantanée, à travers un dispositif léger combinant une caméra .Une fois les signes captés, ils sont analysés et interprétés automatiquement, permettant une interaction en temps réel avec l'entourage.

Afin d'assurer une reconnaissance optimale, plusieurs modèles de classification ont été testés. Le modèle EfficientNet B0 s'est révélé être la solution la plus pertinente, alliant rapidité, fiabilité et simplicité d'intégration. Ses performances en ont fait une option optimale pour une implémentation sur un système embarqué à faible consommation. Par ailleurs, une interface de sortie textuelle et vocale a été intégrée, facilitant une communication claire, instantanée et accessible à tous.

Ce projet va bien au-delà d'une simple innovation technologique. Il répond à un véritable besoin sociétal dans un contexte où les solutions de ce type restent très rares à l'échelle mondiale, et quasiment inexistantes sur le marché algérien. En ce sens, notre système se positionne comme une première réponse concrète et locale à une problématique mondiale : l'inclusion des personnes sourdes et muettes dans la vie quotidienne.

Il s'inscrit ainsi dans une démarche profondément humaine et sociale, visant à briser les barrières de communication et à favoriser l'égalité des chances dans des domaines essentiels tels que l'éducation, la santé, l'emploi et les services publics. En plaçant l'accessibilité, l'autonomie et la dignité des utilisateurs au centre de nos préoccupations, ce projet ambitionne de contribuer activement à une société plus inclusive et solidaire.

En perspective, plusieurs axes d'amélioration seront explorés afin de renforcer l'efficacité et l'impact de notre solution, notamment l'extension du vocabulaire reconnu, incluant des gestes complexes et contextuels, la traduction multilingue automatique (en particulier de la langue des signes algérienne, du français et d'autres langues pertinentes) ainsi que la miniaturisation du système pour un port encore plus discret et confortable.

Grâce à ces perspectives d'évolution, notre système de lunettes intelligentes s'affirme comme une solution intelligente, inclusive et tournée vers l'avenir, prête à répondre aux défis de communication dans une société plus juste, plus connectée, et plus humaine.

Business Model Canvas

1. Proposition de valeur

a) Quels problèmes résolvons-nous pour nos clients ?

1. **Manque d'accessibilité pédagogique :** Les étudiants sourds/muets rencontrent des difficultés pour suivre les cours en l'absence de supports visuels ou d'interprètes.
2. **Barrière de communication :** Les personnes sourdes et muettes éprouvent des difficultés à échanger avec ceux qui ne maîtrisent pas la langue des signes dans leur vie quotidienne.
3. **Exclusion des personnes sourdes/muettes du marché de l'emploi :** Ils sont souvent exclues du marché du travail en raison de la barrière linguistique.
4. **Manque d'intimité :** Certaines personnes préfèrent ne pas dépendre d'un interprète pour des conversations privées, par exemple lors de consultations médicales ou de démarches bancaires.
5. **Besoin d'interprètes :** Les services d'interprètes ne sont pas toujours disponibles, peuvent être coûteux et obligent souvent les personnes sourdes et muettes à dépendre des autres, limitant ainsi leur autonomie.
6. **Manque de solutions accessibles :** Il existe peu de solutions accessibles, légères et abordables permettant de traduire la langue des signes en temps réel.
7. **Accès limité aux droits :** Les personnes sourdes et muettes rencontrent des difficultés à accomplir des démarches administratives sans l'assistance d'un accompagnateur.

b) Quels besoins de nos clients satisfont nos produits ou services ?

1. **Accès à une éducation inclusive :** Offrir une solution pour l'intégration des élèves sourds/muets dans les établissements scolaires classiques.
2. **Traduction en temps réel :** Les lunettes intelligentes captent les gestes de la langue des signes via une caméra intégrée et les traduisent instantanément en texte et/ou en voix pour faciliter la communication directe.
3. **Accès à l'emploi :** Favoriser l'intégration professionnelle des personnes sourdes/muettes en brisant la barrière linguistique.
4. **Confidentialité respectée :** Ce dispositif traite les données en local sur l'appareil, protégeant ainsi la vie privée des utilisateurs.
5. **Autonomie et réduction des coûts :** Communiquer sans interprète, pour plus d'indépendance et moins de frais.
6. **Innovation dans le domaine de l'accessibilité et de l'inclusion :** Offre une solution adaptable pour briser les barrières de communication.
7. **Facilitation administrative :** Traduction instantanée pour aider lors des démarches sans accompagnateur.

c) En quoi notre offre est-elle différente de celle de nos concurrents ?

1. **Utilisation mains libres :** Contrairement aux applications mobiles classiques où il faut tenir un téléphone, notre dispositif permet à l'utilisateur de garder ses mains totalement libres.
2. **Coût réduit :** Notre solution offre une alternative plus économique par rapport aux services d'interprètes humains. L'achat du dispositif représente un investissement unique, bien plus rentable que de payer un interprète à chaque besoin.
3. **Simplicité d'utilisation :** Le dispositif garantit une utilisation simple et facile pour tous les utilisateurs.
4. **Confidentialité :** Les informations sont exclusivement traitées sur l'appareil, donc les utilisateurs n'ont pas à s'inquiéter de leurs données personnelles.

d) Quelles est notre proposition unique de valeur ?

Notre solution repose sur une paire de lunettes intelligentes capables de traduire en temps réel la langue des signes en texte ou en voix, afin de faciliter la communication des personnes sourdes et muettes. Contrairement aux solutions traditionnelles, notre dispositif est portable, discret, accessible et pensé pour un usage quotidien.

Il se distingue par son orientation spécifique vers l'inclusion scolaire, en aidant les élèves sourds et muet à s'intégrer dans les écoles classiques. À ce jour, aucune autre solution commerciale mondiale ne cible directement le domaine éducatif avec une approche aussi concrète.

Nous avons également conçu une fonctionnalité exclusive : un système d'alerte permettant à l'utilisateur d'envoyer une notification quand il souhaite communiquer.

Bien que centré sur l'éducation, notre dispositif est aussi adaptable à d'autres domaines comme le secteur médical, administratif ou professionnel. En combinant l'intelligence artificielle, accessibilité et impact social, notre projet se positionne comme une innovation pionnière au service de l'inclusion.

2. Segments de clients

a) Quels sont nos clients principaux ?

1. Établissements d'éducation classique (écoles, lycées, etc.).
2. Institutions de santé (hôpitaux, cliniques, etc.).
3. Entreprises qui intègrent des employés sourds et muets.
4. Associations et ONG œuvrant dans le domaine du handicap auditif.

5. Organismes gouvernementaux engagés pour l'inclusion sociale.
6. Entreprises technologiques développant des outils d'assistance.
7. Proches des personnes sourdes et muettes.
8. Services d'accueil publics (hôtels, gares, aéroports, administrations).
9. Particuliers sourds et muettes recherchant plus d'autonomie.

b) Quels sont les différents segments de clients que nous visons ?

1. Secteur de l'éducation
2. Secteur de la santé
3. Secteur public
4. Secteur privé
5. Secteur associatif
6. Clients individuels

c) Quels sont les besoins spécifiques de chaque segment de clients ?

1. **Secteur de l'éducation :** Besoin d'une solution d'accessibilité pour les étudiants sourds/muettes, facilitant leur compréhension des cours en temps réel.
2. **Secteur de la santé :** Besoin d'une communication claire entre les patients sourds/muettes et le personnel médical, en particulier dans les situations urgentes.
3. **Secteur public :** Besoin de rendre les services publics accessibles, en assurant la communication avec les individus sourds/muets dans les administrations et espaces publics.
4. **Secteur privé :** Besoin de solutions accessibles pour les employés et clients sourds/muets, améliorant l'intégration et la communication au sein des entreprises privées.
5. **Secteur associatif :** Besoin de solutions adaptées pour sensibiliser et soutenir l'inclusion des personnes sourds/muettes, en fournissant des outils pour améliorer la communication et l'autonomie dans leurs interactions sociales.
6. **Clients individuels :** Besoin d'un dispositif portable et discret pour améliorer les interactions entre les personnes sourdes/muettes et leurs proches.

d) Comment pouvons-nous catégoriser nos clients en groupes distincts ?

1. **Par type de client :**
   (a) Petites et moyennes entreprises (PME)
   (b) Grandes entreprises
   (c) Organisations à but non lucratif
   (d) Institutions gouvernementales
   (e) Individus
2. **Par secteur d'activité :**
   (a) Secteur d'éducation
   (b) Secteur de la santé
   (c) Secteur associatif
   (d) Secteur public
   (e) Secteur privé

3. Relation avec les clients

a) Quel type de relation chaque segment de clients attend-il de nous ?

1. **Secteur de l'éducation :**
   - Sessions de formation pour les enseignants et les étudiants.
   - Suivi post-achat pour assurer la satisfaction à long terme.
2. **Secteur de la santé :**
   - Garantir la confidentialité des échanges et des données.
   - Fournir un support technique fiable et réactif.
3. **Secteur public :**
   - Assistance continue pour assurer l'accessibilité aux services publics.
   - Suivi régulier pour adapter la solution aux besoins évolutifs.
4. **Secteur privé :**
   - Formation interne pour l'adoption optimale de la solution par les employés.
   - Service d'assistance disponible en continu.
5. **Secteur associatif :**
   - Sensibilisation aux enjeux liés à la surdité.
   - Engagement social à travers des actions conjointes.

6. **Clients individuels :**
    - Accès facile et rapide à un support client intuitif et réactif.
    - Création d'une communauté en ligne pour favoriser les échanges et l'entraide.

b) Comment entretenons-nous actuellement les relations avec nos clients ?

1. **Canaux de communication ouverts :** Nous maintenons des canaux de communication ouverts (email, réseaux sociaux) pour être facilement accessibles et réactifs.
2. **Formation et assistance :** Des sessions de formation sont organisées pour les utilisateurs afin de garantir une utilisation optimale du dispositif.
3. **Feedback et amélioration continue :** Nous recueillons régulièrement des retours utilisateurs pour améliorer et ajuster nos produits en fonction de leurs suggestions.
4. **Engagement dans la communauté :** Nous participons activement à des conférences et autres événements pour répondre aux préoccupations des utilisateurs et promouvoir l'inclusion.

c) Comment pouvons-nous améliorer ou personnaliser nos interactions avec nos clients ?

1. **Support amélioré :** Support 24/7.
2. **Programmes de fidélité :** Avantages exclusifs et réductions.
3. **Engagement proactif :** Vérifications régulières de la satisfaction client.
4. **Personnalisation des offres :** Proposer des offres sur mesure en fonction de segment de clients.
5. **Retour d'information régulier :** Canaux de retour pour améliorer les produits.

4. Canaux de distribution

a) Par quels canaux nos clients veulent-ils être atteints ?

1. **Boutiques en ligne** : De nombreux clients souhaitent acheter directement en ligne via des sites de vente.
2. **Réseaux sociaux et marketing digital** : Les plateformes sociales comme Facebook et Instagram sont des canaux clés pour atteindre un large public et générer de l'engagement via des campagnes ciblées.

3. **Salons et événements spécialisés** : Les clients intéressés par les innovations technologiques et l'accessibilité cherchent souvent à interagir lors de salons, conférences, ou événements spécialisés dans la santé et la technologie d'assistance.
4. **Webinaires et démonstrations en ligne** : Sessions interactives pour présenter le produit et répondre aux questions.
5. **Revendeurs et distributeurs spécialisés** : Points de vente physiques pour tester et acheter le dispositif en boutique.
6. **Partenariats institutionnels** : Collaboration avec établissements scolaires, hôpitaux et associations pour distribution et formation sur place.
7. **Presse professionnelle et médias spécialisés** : Articles et reportages dans la presse dédiée à présenter le produit.

b) Quels canaux sont les plus efficaces pour atteindre chaque segment de clients ?

1. **Secteur de l'éducation**
   - Partenariats institutionnels (écoles, lycée, etc.)
   - Webinaires dédiés aux enseignants
   - Salons et conférences pédagogiques
2. **Secteur de la santé**
   - Revendeurs spécialisés en matériel médical
   - Partenariats avec hôpitaux et cliniques
   - Articles dans la presse médicale spécialisée
3. **Secteur public**
   - Salons et forums des services publics
   - Presse professionnelle et rapports officiels
4. **Secteur privé**
   - Boutiques en ligne B2B
   - Revendeurs et showrooms d'entreprise
   - Réseaux sociaux
5. **Secteur associatif**
   - Réseaux sociaux et groupes dédiés au handicap
   - Partenariats avec ONG et associations locales
   - Webinaires de sensibilisation et d'entraide
6. **Clients individuels**

- Boutiques en ligne grand public
- Réseaux sociaux grand public (Facebook, Instagram)
- Sessions de démonstration en ligne (webinaires)

c) Comment pouvons-nous intégrer différents canaux pour améliorer l'expérience clients ?

1. **Réseaux sociaux + Boutique en ligne :** Les publications sur des plateformes telles qu'Instagram ou Facebook peuvent diriger directement les utilisateurs vers les sites de vente en ligne, facilitant ainsi le passage de la découverte à l'achat immédiat.
2. **Webinaires + Partenariats institutionnels :** Les sessions de démonstration en ligne peuvent être organisées en collaboration avec des établissements scolaires, hôpitaux ou associations.
3. **Salons spécialisés + Revendeurs physiques :** Lors des événements comme les salons ou conférences, des démonstrations peuvent être associées à des rencontres directes avec des revendeurs ou distributeurs locaux, facilitant l'achat immédiat ou la commande sur place.
4. **Presse spécialisée + Webinaires :** Les publications dans la presse spécialisée peuvent inclure des liens vers des inscriptions à des démonstrations en ligne.

5. Partenaires clés

a) Qui sont nos partenaires clés ?

1. **Associations de personnes sourdes et muettes :** Leur participation aux tests utilisateurs et à la validation fonctionnelle, accompagnée de retours réels, améliore l'aspect inclusif du produit.
2. **Fournisseurs de composants électroniques :** Ils assurent l'approvisionnement de matériel nécessaire à la fabrication du dispositif.
3. **Distributeurs spécialisés en matériel médical :** Ils assurent la mise à disposition du produit auprès des professionnels de la santé et des établissements médicaux.
4. **Détaillants technologiques ou IoT :** Ils rendent la solution accessible au grand public via des points de vente spécialisés.
5. **Mentors et experts en intelligence artificielle :** Ils apportent leur expertise pour perfectionner les algorithmes de reconnaissance des gestes, assurant ainsi la performance des modèles.
6. **Startups et entreprises spécialisées dans l'accessibilité :** Elles coopèrent pour développer des solutions technologiques qui facilitent l'inclusion sociale.

b) Quels sont les partenariats qui nous aident à réduire les coûts, à accéder à de nouvelles ressources ou à améliorer notre proposition de valeur ?

1. **Fournisseurs électroniques :** Diminution des coûts par des achats en grande quantité.
2. **Associations de sourds/muets :** Un lien direct avec la communauté cible, réduisant les coûts des études et améliorant la pertinence du produit.
3. **Experts en IA :** Accélérer l'optimisation technique tout en limitant les dépenses grâce à des conseils spécialisés.
4. **Startups en accessibilité :** Partage de compétences et de technologies pour améliorer la solution.
5. **Distributeurs et détaillants tech :** Permettre l'élargissement du marché tout en limitant les investissements dans l'infrastructure commerciale.

c) Comment pouvons-nous aligner nos intérêts avec ceux de nos partenaires ?

1. **Objectifs communs :** Collaborer autour de valeurs partagées telles que l'inclusion, l'innovation technologique et l'amélioration de la qualité de vie.
2. **Partage de ressources et de compétences :** Combiner des moyens (données, outils technologiques) afin de réaliser des objectifs communs tout en réduisant les coûts.
3. **Création de valeur conjointe :** Créer des offres ou des services complémentaires ensemble.
4. **Reconnaissance mutuelle :** Souligner l'importance de la participation de chaque partenaire en utilisant des outils comme les références dans des publications.
5. **Renforcement de leur image de responsabilité sociale :** Contribuer à améliorer l'image sociétale des partenaires en les associant à un projet à fort impact social.
6. **Co-développement :** Impliquer certains partenaires (associations, experts, startups) dans le processus de développement pour répondre aux besoins réels des utilisateurs.
7. **Communication transparente et feedback régulier :** permet d'aligner les décisions stratégiques pour l'évolution du projet.

6. Activités clés

a) Quelles sont les actions principales que nous devons entreprendre pour livrer notre proposition de valeur ?

1. **Développement du modèle d'intelligence artificielle :** Perfectionner les algorithmes de reconnaissance des gestes afin de garantir une traduction précise et sans erreur.

2. **Collecte et traitement des données :** Rassembler des jeux de données en langue des signes, les annoter et les prétraiter pour améliorer la performance du modèle IA.
3. **Amélioration continue du prototype :** Incorporer les retours des utilisateurs (associations, testeurs) afin d'ajuster le produit à leurs besoins spécifiques et optimiser l'ergonomie ainsi que la fiabilité.
4. **Développement d'une stratégie de commercialisation :** Définir les approches pour introduire et promouvoir le produit sur le marché.
5. **Suivi des performances et feedback utilisateur :** Mettre en place un suivi après le lancement pour collecter les avis et évaluer les performances du produit.

b) Quelles sont les opérations essentielles pour notre entreprise ?

1. **Tests et validation technique :** Vérifier de manière régulière la performance du dispositif sur différents profils d'utilisateurs.
2. **Formation :** Proposer des sessions de formation afin de simplifier la prise en main du produit.
3. **Support technique et assistance utilisateur :** Assurer une aide continue en cas de problème technique.
4. **Veille technologique et réglementaire :** Se tenir au courant des progrès en IA, des normes d'accessibilité et des obligations légalesn.
5. **Gestion administrative et logistique :** Organiser l'achat des composants, superviser les stocks et planifier les livraisons afin de garantir la continuité du service.
6. **Gestion de la relation client :** Évaluer régulièrement la satisfaction afin d'adapter le produit.

c) Quelles sont les activités qui créent le plus de valeur pour nos clients ?

1. **Précision de la traduction en temps réel :** Un modèle IA performant et réactif garantit une expérience fluide pour les utilisateurs.
2. **Accessibilité et simplicité d'utilisation :** Le confort et la facilité d'usage sont des facteurs importants.
3. **Écoute des retours utilisateurs :** La prise en charge rapide des retours assure une meilleure satisfaction des utilisateurs.

7. Ressources clés

a) Quels sont nos actifs matériels, immatériels et humains essentiels ?

1. **Matériels**
    - Composants électroniques (Raspberry Pi, caméras, etc.)
    - Matériel informatique (ordinateurs puissants pour l'entraînement des modèles IA)
    - Outils de prototypage
2. **Immatériels**
    - Technologies IA développées (modèles pour la reconnaissance des gestes)
    - Données en langue des signes (corpus collecté pour entraîner et tester les algorithmes)
3. **Humains**
    - Ingénieurs IA
    - Experts en marketing
    - Équipe de support
    - Experts en langue des signes

b) Quels sont les outils, les technologies ou les partenariats dont nous avons besoin pour réussir ?

1. **Outils nécessaires**
    - Environnements de développement intégré (IDE)
    - Outils d'accès à distance (VNC pour gérer le Raspberry Pi)
    - Service de stockage en ligne, utilisé pour sauvegarder et partager des jeux de données.
    - Logiciel Arduino (pour programmer et transférer le code vers la carte Arduino)
    - Outils de prototypage (logiciels de modélisation)
2. **Technologies nécessaires**
    - MediaPipe pour la détection et l'extraction des points clés des mains (hand landmarks).
    - TensorFlow Lite pour exécuter les modèles d'intelligence artificielle en local, avec des performances optimisées.
    - Drivers et bibliothèques nécessaires pour faire fonctionner les composants électroniques.

3. **Partenariats nécessaires**
    - Partenariats avec des fournisseurs de composants électroniques pour garantir la qualité et la disponibilité des composants nécessaires.
    - Partenariats avec des associations de sourds/muets et des experts en accessibilité.
    - Partenariats avec détaillants et distributeurs spécialisés en IoT pour rendre le dispositif accessible au grand public.
    - Collaborations avec des universités ou des centres de recherche pour l'amélioration continue des modèles.

c) Quels sont les principaux avantages concurrentiels de nos ressources ?

1. **Expertise technique et compétences clés :** Une équipe spécialisée dans l'intelligence artificielle, offrant une expertise pointue dans la reconnaissance de la langue des signes .
2. **Technologies avancées :** L'utilisation de modèles d'intelligence artificiel, et d'outils modernes de prototypage et de développement garantissent une innovation continue et une solution de qualité.
3. **Partenariats stratégiques :** Des collaborations avec des experts, des associations, garantissant des retours de la communauté cible et des avancées technologiques constantes.
4. **Accessibilité et diffusion :** Des partenariats avec des détaillants spécialisés, permettant une large distribution du produit à un public cible tout en réduisant les coûts d'infrastructure commerciale.

8. Charges et coûts

a) Quels sont les coûts fixes et variables associés à notre modèle économique ?

1. **Coûts fixes :**
    - **Salaires** : Pour les ingénieurs d'IA (50 000 Da/mois) et les experts marketing (40 000 Da/mois).
    - **Abonnements logiciels et services cloud** : (57 000 Da/an).
    - **Campagnes marketing** : Publicité digitale, salons technologiques (500 000 Da/an).
    - **Frais d'établissement et légaux** : Enregistrement, frais juridiques (500 000 Da la première année).
2. **Coûts variables :**
    - **Composants électroniques** : Environ (44 145 Da/unité).

- **Coûts de distribution et logistique** : Emballage, livraison des dispositifs, selon le volume de production.
- **Maintenance technique** : Mises à jour, corrections de bugs.
- **Support client** : Service après-vente et assistance.

Le tableau ci-dessous permet de synthétiser les différents dépenses liés à notre projet, en distinguant les coûts de la première année et ceux estimés pour la deuxième année.

Tab. 4.1 : Répartition des dépenses annuels (en Dinar Algérien)

Catégorie	**Type de dépense**	**Première année (Da)**	**Après un an (Da)**
Établissement	Frais d'enregistrement, frais juridiques	500 000	–
Bureau et infrastructure	Loyer, électricité, Internet	280 000	280 000
Matériel et logiciel	PC, licences logicielles, espace de stockage cloud	400 000	60 000
Salaires	2 ingénieurs IA, 1 expert marketing	1 680 000	1 860 000
Campagne marketing	Publicité digitale, salons technologiques, démonstrations publiques	500 000	400 000
Coût des matières premières	Composants pour la fabrication des lunettes (Raspberry Pi, caméras, etc.)	3 973 050	4 141 500
Autres dépenses	Maintenance, déplacements, comptable, cotisations sociales	500 000	500 000

b) Quels sont les coûts les plus importants pour notre entreprise ?

Les coûts les plus importants pour notre entreprise sont :

1. **Les salaires** des ingénieurs IA et experts marketing.
2. **Les coûts des composants électroniques** pour la fabrication du dispositif.
3. **Les coûts liés à la recherche et au développement** des modèles d'intelligence artificielle.

4. **Les campagnes marketing** qui sont essentielles pour la promotion du produit et la sensibilisation du public à l'accessibilité.

c) Comment pouvons-nous réduire les coûts ou améliorer l'efficacité de nos opérations ?

Pour réduire les coûts ou améliorer l'efficacité, nous pouvons envisager les stratégies suivantes :

1. **Optimiser les achats de composants électroniques** en établissant des partenariats avec des fournisseurs pour obtenir des prix réduits par des achats en volume.
2. **Utiliser des outils open-source** pour réduire les coûts de licences logicielles.
3. **Réduire les coûts de R&D** en collaborant avec des universités et des centres de recherche pour bénéficier d'expertise à moindre coût tout en améliorant les modèles IA.
4. **Optimiser les dépenses marketing** en favorisant des canaux numériques à coût réduit et mettant l'accent sur des campagnes virales pour accroître la visibilité.

9. Revenus

a) Quels produits ou services nos clients sont-ils prêts à payer ?

1. **Achat des lunettes intelligentes** : Prix unitaire fixé à 95 800 Da.
2. **Formations personnalisées** : propose différentes séances de formation adaptées aux profils et aux besoins spécifiques des utilisateurs, à partir de 15 000 DA.
3. **Support technique dédié** : Services d'assistance pour les entreprises et les particuliers.
4. **Abonnement mensuel** : Inclut des outils supplémentaires et des améliorations continues pour les utilisateurs souhaitant une version enrichie.
5. **Location à l'heure** : Mise à disposition du dispositif pour des événements ou des besoins ponctuels à 750 Da/heure.
6. **Partenariats institutionnels et associatifs** : Revenus générés à travers des collaborations et des accords de fourniture d'équipements à des structures spécialisées.

b) Quels sont les différents moyens par lesquels nous pouvons générer des revenus ?

1. **Vente directe** : Source de revenus assurée par la vente du dispositif à un tarif fixe.

2. **Services additionnels** : Gains tirés de services additionnels proposés aux utilisateurs, comme le support technique et de la location ponctuelle à l'heure.

3. **Partenariats** : Financements ou contrats de collaboration avec différentes associations ou structures.

4. **Abonnements** : Paiements mensuels pour l'accès à des fonctionnalités supplémentaires.

c) Quel est notre modèle de tarification ?

1. **Tarification à la carte** : Facturation des services de formation et des mises à jour selon la durée ou la complexité, avec des prix modulés en fonction de ces critères.

2. **Tarification par utilisation** : Facturation en fonction de l'utilisation réelle du dispositif, comme la location horaire de lunettes intelligentes, permettant aux clients de ne payer que pour le temps ou les services effectivement utilisés.

3. **Tarification personnalisée** : Proposer des solutions de tarification personnalisées en fonction des exigences spécifiques de chaque client.

4. **Tarification basée sur l'abonnement** : Proposer des tarifs mensuels donnant accès à des fonctionnalités avancées selon le niveau d'abonnement choisi.

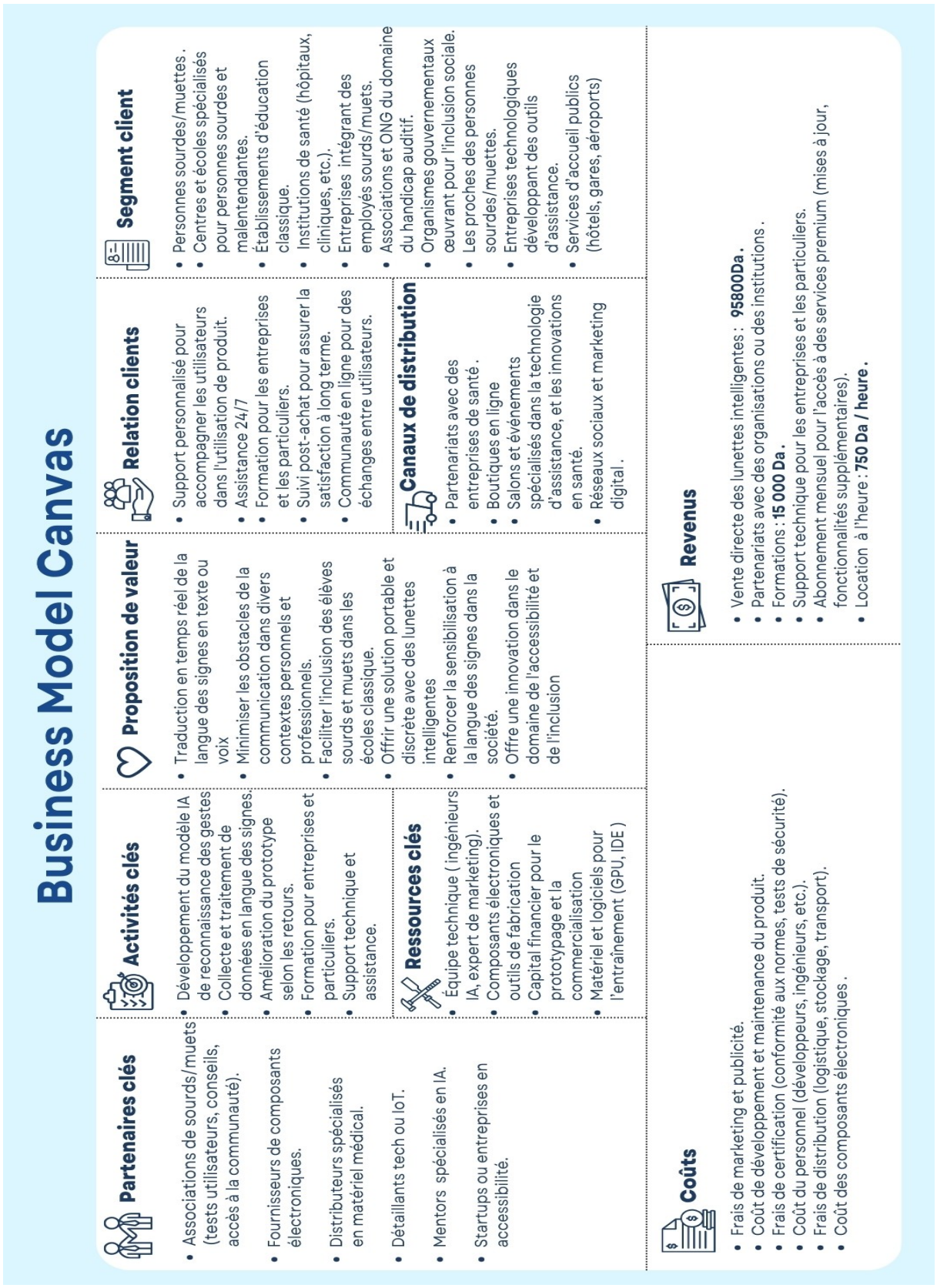

Fig. 4.1 : Business Model Canvas

Bibliographie

[1] Ray-Ban, “Découvrez les lunettes ia ray-ban | meta.” Disponible sur : https://www.ray-ban.com/france/discover-ray-ban-meta-ai-glasses/clp. [Consulté le 1 février 2025].

[2] Meta, “Introducing the new ray-ban | meta smart glasses.” Disponible sur : https://about.fb.com/news/2023/09/new-ray-ban-meta-smart-glasses/, 2023. [Consulté le 1 février 2025].

[3] Vuzix Corporation, “Vuzix-blade2-smart-glasses-v2.7.” Disponible sur : https://vuzix-website.s3.amazonaws.com/files/Content/Vuzix+Blade+2/Vuzix-Blade2-Smart-Glasses-Product-Sheet.pdf. [Consulté le 1 février 2025].

[4] VR Experts, “Acheter le vuzix blade 2 - vr experts, le spécialiste vr/ar.” Disponible sur : https://vr-experts.fr/lunettes-de-realite-augmentee/vuzix-blade-2-acheter/. [Consulté le 1 février 2025].

[5] BrightSign, “Brightsign glove - translate any sign into any language.” Disponible sur : https://www.brightsignglove.com/#features. [Consulté le 2 février 2025].

[6] BrightSign, “Order your brightsign glove now !.” Disponible sur : https://shop.brightsignglove.com/products/glove. [Consulté le 2 février 2025].

[7] M. Emery, “Uh students develop prototype device that translates sign language - university of houston.” Disponible sur : https://www.uh.edu/news-events/stories/2012/may/0529MyVoice.php, mai 2012. [Consulté le 3 février 2025].

[8] G. Wu, “Kinect sign language translator – part 1.” Disponible sur : https://www.microsoft.com/en-us/research/blog/kinect-sign-language-translator-part-1/, 2013. [Consulté le 3 février 2025].

[9] ELBILADTV, “Gant intelligent qui traduit la langue des signes en mots écrits et parlés..” Disponible sur : https://www.youtube.com/watch?v=mpkmn7L06so, 2024. [Consulté le 6 mai 2025].

[10] W. I. P. Organization, “Cn103714732.” Disponible sur : https://patentscope.wipo.int/search/en/detail.jsf?docId=CN97387065, 2014. [Consulté le 4 février 2025].

[11] W. I. P. Organization, “Cn215730427.” Disponible sur : https://patentscope.wipo.int/search/en/detail.jsf?docId=CN350556537, 2022. [Consulté le 4 février 2025].

[12] W. I. P. Organization, “Cn106683533.” Disponible sur : https://patentscope.wipo.int/search/en/detail.jsf?docId=CN198387425&_cid=P22-M6JJJN-72177-2, 2017. [Consulté le 4 février 2025].

[13] W. I. P. Organization, “Cn110020442.” Disponible sur : https://patentscope.wipo.int/search/en/detail.jsf?docId=CN249764145&_cid=P22-M6JJJN-72177-2, 2019. [Consulté le 4 février 2025].

[14] W. I. P. Organization, “Kr1020230001548.” Disponible sur : https://patentscope.wipo.int/search/en/detail.jsf?docId=KR390079956&_cid=P22-M6JJJN-72177-1, 2023. [Consulté le 5 février 2025].

[15] W. I. P. Organization, “Cn210574528.” Disponible sur : https://patentscope.wipo.int/search/fr/detail.jsf?docId=CN295810012&_cid=P21-M6NSPB-39895-1, 2020. [Consulté le 5 février 2025].

[16] W. I. P. Organization, “Kr1020200107572.” Disponible sur : https://patentscope.wipo.int/search/fr/detail.jsf?docId=KR307248572&_cid=P21-M6N9CW-32692-1, 2020. [Consulté le 5 février 2025].

[17] W. I. P. Organization, “Cn109923462b.” Disponible sur : https://patents.google.com/patent/CN109923462B/en?q=(sign+language+glasses)&oq=sign+language+glasses, 2021. [Consulté le 6 février 2025].

[18] W. I. P. Organization, “Us10446059b2.” Disponible sur : https://patents.google.com/patent/US10446059B2/en?q=(sign+language+glove)&oq=sign+language+glove+&page=1, 2019. [Consulté le 6 février 2025].

[19] H. Orovwode, I. D. Oduntan, and J. Abubakar, “Development of a sign language recognition system using machine learning,” in *2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1–8, 2023.

[20] S. H. Taher Karim, M. L. Mahmood, S. S. Abdulla, and S. A. Abdulla, “Kurdish sign language recognition using convolutional neural network (cnn),” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 16, p. 19–26, Sep. 2024.

[21] S. K. Paul, M. A. A. Walid, R. R. Paul, M. J. Uddin, M. S. Rana, M. K. Devnath, I. R. Dipu, and M. M. Haque, “An adam based cnn and lstm approach for sign language recognition in real time for deaf people,” *Bulletin of Electrical Engineering and Informatics*, 2024.

[22] S. Katoch, V. Singh, and U. S. Tiwary, “Indian sign language recognition system using surf with svm and cnn,” *Array*, vol. 14, p. 100141, 04 2022.

[23] Y. Rokade and P. Jadav, “Indian sign language recognition system,” *International Journal of Engineering and Technology*, vol. 9, pp. 189–196, 07 2017.

[24] J. Olszewska and M. Quinn, “British sign language recognition in the wild based on multi-class svm,” pp. 81–86, 09 2019.

[25] S. Nagarajan and T. Subashini, “Static hand gesture recognition for sign language alphabets using edge oriented histogram and multi class svm,” *International Journal of Computer Applications*, vol. 82, pp. 28–35, 11 2013.

[26] T. N. Abu-Jamie and S. S. Abu-Naser, “Classification of sign-language using vgg16,” *International Journal of Academic Engineering Research (IJAER)*, vol. 6, no. 6, pp. 36–46, 2022.

[27] D. Rathi, “Optimization of transfer learning for sign language recognition targeting mobile platform,” *CoRR*, vol. abs/1805.06618, 2018.

[28] D. R. Kothadiya, C. M. Bhatt, T. Saba, A. Rehman, and S. A. O. Bahaj, “Signformer : Deepvision transformer for sign language recognition,” *IEEE Access*, vol. 11, pp. 4730–4739, 2023.

[29] A. Imran, M. S. Hulikal, and H. A. A. Gardi, “Real time american sign language detection using yolo-v9,” 2024.

[30] D. Martin and P. Martin, “Image numérique et image de synthèse.” Disponible sur : `https://www.universalis.fr/encyclopedie/image-numerique-et-image-de-synthese/`. [Consulté le 18 mars 2025].

[31] J. Alzubi, A. Nayyar, and A. Kumar, “Machine learning from theory to algorithms : An overview,” *Journal of Physics : Conference Series*, vol. 1142, p. 012012, nov 2018.

[32] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[33] T. Mitchell, *Machine Learning*. McGraw-Hill International Editions, McGraw-Hill, 1997.

[34] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA : Springer, 2006.

[35] Organisation internationale de normalisation (ISO), “Apprentissage profond : la mécanique de la magie.” Disponible sur : `https://www.iso.org/fr/intelligence-artificielle/apprentissage-profond-deep-learning`. [Consulté le 19 mars 2025].

[36] Red Hat, “Le deep learning, qu’est-ce que c’est ?.” Disponible sur : `https://www.redhat.com/fr/topics/ai/what-is-deep-learning`, 2025. [Consulté le 18 mars 2025].

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[38] S. Almalki and M. Abdullah, “Arabic handwriting recognition using convolutional neural network (cnn),” 11 2024.

[39] V. L. Helen Josephine, A. Nirmala, and V. L. Alluri, “Impact of hidden dense layers in convolutional neural network to enhance performance of classification model,” *IOP Conference Series : Materials Science and Engineering*, vol. 1131, p. 012007, apr 2021.

[40] GeeksforGeeks, “What is fully connected layer in deep learning ?.” Disponible sur : `https://www.geeksforgeeks.org/what-is-fully-connected-layer-in-deep-learning/`, 2024. [Consulté le 20 mars 2025].

[41] E. Oye and R. Lucas, “Convolutional neural networks (cnns),” 12 2024.

[42] Amazon Web Services (AWS), “Qu’est-ce que l’apprentissage par transfert ?.” Disponible sur : `https://aws.amazon.com/fr/what-is/transfer-learning/`. [Consulté le 21 mars 2025].

[43] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks : Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018.

[44] TensorFlow, "tf.keras.applications.mobilenet_v2.preprocess_input." Disponible sur : `https://www.tensorflow.org/api_docs/python/tf/keras/applications/mobilenet_v2/preprocess_input`, 2024. [Consulté le 18 avril 2025].

[45] M. Tan and Q. V. Le, "Efficientnet : Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019.

[46] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," *CoRR*, vol. abs/2104.00298, 2021.

[47] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. Chang, M. G. Yong, J. Lee, W. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe : A framework for building perception pipelines," *CoRR*, vol. abs/1906.08172, 2019.

[48] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. Chang, and M. Grundmann, "Mediapipe hands : On-device real-time hand tracking," *CoRR*, vol. abs/2006.10214, 2020.

[49] Google AI Edge, "Guide de détection des points de repère de la main." Disponible sur : `https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker?hl=fr`, 2025. [Consulté le 25 mars 2025].

[50] TensorFlow, "Guide tensorflow lite." Disponible sur : `https://www.tensorflow.org/lite/guide?hl=fr`, 2021. [Consulté le 27 mars 2025].

[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout : A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[52] Swapna, "Convolutional neural network | deep learning." Disponible sur : `https://developersbreach.com/convolution-neural-network-deep-learning/`, Aug. 2020. [Consulté le 29 mars 2025].

[53] B. Kromydas, "Convolutional neural network (cnn) : A complete guide." Disponible sur : `https://learnopencv.com/understanding-convolutional-neural-networks-cnn/`, Jan. 2023. [Consulté le 29 mars 2025].

[54] Raspberry Pi , "Raspberry pi for home." Disponible sur : `https://www.raspberrypi.com/for-home/`. [Consulté le 7 avril 2025].

[55] The Pi Hut, "Raspberry pi 4 model b." Disponible sur : `https://thepihut.com/products/raspberry-pi-4-model-b`. [Consulté le 7 avril 2025].

[56] Raspberry Pi, "Raspberry pi 4 model b specifications." Disponible sur : `https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/`. [Consulté le 7 avril 2025].

[57] Raspberry Pi, “À propos des modules caméra.” Disponible sur : https://www.raspberrypi.com/documentation/accessories/camera.html#about-the-camera-modules. [Consulté le 9 avril 2025].

[58] CanaKit, “Raspberry pi camera module 3.” Disponible sur : https://www.canakit.com/raspberry-pi-camera-module-3.html. [Consulté le 9 avril 2025].

[59] Raspberry Pi Ltd, “Raspberry pi camera module 3 product brief.” Disponible sur : https://datasheets.raspberrypi.com/camera/camera-module-3-product-brief.pdf, June 2024. [Consulté le 9 avril 2025].

[60] Arduino, “What’s arduino ?.” Disponible sur : https://docs.arduino.cc/learn/starting-guide/whats-arduino/, 6 2025. [Consulté le 11 avril 2025].

[61] SparkFun Electronics, “Arduino uno - r3.” Disponible sur : https://www.flickr.com/photos/sparkfun/8406865680/, 1 2013. [Consulté le 11 avril 2025].

[62] lexset, “Synthetic asl alphabet.” Disponible sur : https://www.kaggle.com/datasets/lexset/synthetic-asl-alphabet, 6 2022. [Consulté le 22 janvier 2025].

[63] D. Sau, “Asl american sign language alphabet dataset.” Disponible sur : https://www.kaggle.com/datasets/debashishsau/aslamerican-sign-language-aplhabet-dataset, 10 2021. [Consulté le 18 janvier 2025].

[64] ayuraj, “American sign language dataset.” Disponible sur : https://www.kaggle.com/datasets/ayuraj/asl-dataset, 4 2019. [Consulté le 20 janvier 2025].

[65] B. Pandey, “Hand sign recognition.” Disponible sur : https://www.kaggle.com/datasets/bikashpandey17/hand-sign-recognition, 3 2020. [Consulté le 18 janvier 2025].

[66] A. G. Ortiz, “American sign language.” Disponible sur : https://www.kaggle.com/datasets/angelgortiz/american-sign-language, 2 2022. [Consulté le 24 janvier 2025].

[67] C. Kakade, “American sign language dataset.” Disponible sur : https://www.kaggle.com/datasets/chaitanyakakade77/american-sign-language-dataset, 3 2024. [Consulté le 24 janvier 2025].

[68] V. Murray, “American sign language - asl.” Disponible sur : https://www.kaggle.com/datasets/victormurray/asldataset, 3 2022. [Consulté le 27 janvier 2025].

[69] M. A. Nair, “American sign language training dataset.” Disponible sur : https://www.kaggle.com/datasets/madhavanair/american-sign-language-dataset, 10 2024. [Consulté le 27 janvier 2025].

[70] K. Muvezwa, “Significant (asl) sign language alphabet dataset.” Disponible sur : https://www.kaggle.com/datasets/kuzivakwashe/significant-asl-sign-language-alphabet-dataset, 3 2019. [Consulté le 27 janvier 2025].

Annexe A : Questionnaire destiné aux enseignants

Date : . . . . . .

Nom de l'enseignant(e) (facultatif) : . . . . . .

Fonction : . . . . . .

1. Contexte et utilité perçue

1.1. Avez-vous déjà entendu parler de dispositifs technologiques permettant la traduction de la langue des signes ?
□ Oui □ Non
Si oui, lesquels ? . . . . . .

1.2. Pensez-vous que ce type de dispositif (lunettes intelligentes) pourrait être utile dans l'environnement scolaire ?
□ Oui □ Non □ Peut-être
Pourquoi ? . . . . . .

1.3. À votre avis, quels publics pourraient le plus en bénéficier ?
□ Élèves sourds et malentendants
□ Enseignants
□ Parents
□ Autres : . . . . . .

2. Observation du fonctionnement

2.1. La détection des signes vous semble-t-elle :
□ Rapide □ Moyennement rapide □ Trop lente

2.2. La reconnaissance des gestes est-elle correcte selon vous ?
□ Très précise □ Moyennement précise □ Peu précise □ Inexacte

2.3. Avez-vous remarqué des erreurs ? Si oui, lesquelles ?
. . . . . .
. . . . . .

2.4. Le système est-il facile à comprendre pour un utilisateur non technique ?
□ Oui □ Moyennement □ Non

3. Ergonomie et accessibilité

3.1. Avez-vous remarqué des difficultés d'utilisation pour les élèves (ex : position des mains, distance, lumière) ?
□ Oui □ Non
Si oui, lesquelles ? . . . . . .

4. Suggestions et remarques

4.1. Quelles améliorations proposeriez-vous pour ce prototype ?

. . . . . .

. . . . . .

4.2. Autres commentaires :

. . . . . .

. . . . . .

5. Appréciation du concept et des besoins

5.1. Pensez-vous que le développement d'un outil technologique de reconnaissance de la langue des signes répond à un besoin réel dans le contexte éducatif des sourds ou malentendants ?

□ Oui □ Non □ Partiellement

Commentaires : . . . . . .

5.2. Dans votre établissement, quelle langue des signes est prioritairement utilisée ?

□ Langue des Signes Algérienne (LSA)

□ Langue des Signes Française (LSF)

□ Langue des Signes Américaine (ASL)

□ Langue des Signes Arabe Unifiée (LSAU)

6. Vision à long terme

6.1. Pensez-vous qu'un tel projet pourrait avoir un impact positif dans l'avenir ?

□ Oui □ Non □ Potentiellement

Expliquez votre point de vue : . . . . . .

7. Enjeux et perspectives

7.1. Quelles seraient selon vous les conditions nécessaires pour envisager l'intégration future d'un tel outil dans un cadre pédagogique classique ?

□ Simplicité d'utilisation

□ Exactitude des traductions

□ Respect de la langue des signes locale

□ Acceptation par les élèves et enseignants

□ Formation du personnel

□ Autres critères : . . . . . .

7.2. Quels risques ou réserves pourriez-vous soulever concernant ce type d'innovation ?

□ Risque de mauvaise interprétation des signes

□ Perte du lien humain

□ Coût et maintenance

□ Résistance au changement

□ Autres : . . . . . .

Annexe B : Questionnaire destiné aux étudiants

Date : . . . . . .

Nom de l'étudiant(e) (facultatif) : . . . . . .

Âge : . . . . . .

1. Compréhension du fonctionnement

1.1. As-tu compris à quoi servent les lunettes ?
□ Oui □ Un peu □ Non

2. Facilité d'utilisation

2.1. As-tu pu faire les gestes normalement devant la caméra ?
□ Oui □ Un peu difficile □ Non

2.2. Les signes que tu fais sont-ils bien compris par le système ?
□ Oui □ Parfois □ Non

3. Affichage

3.1. As-tu vu le bon mot ou texte s'afficher ?
□ Oui □ Pas toujours □ Non

4. Vision pédagogique et institutionnelle

4.1. Selon toi, ce type d'outil technologique favorise-t-il l'inclusion des élèves sourds ou muets dans les écoles classiques ?
□ Oui □ Partiellement □ Non
Commentaires : . . . . . .

4.2. Serait-il envisageable d'intégrer ce type de technologie comme outil d'apprentissage complémentaire ?
□ Oui □ Non □ Peut-être
Conditions nécessaires selon toi : . . . . . .

5. Ton expérience avec la langue des signes

5.1. Depuis combien de temps utilises-tu la langue des signes ?
□ Depuis l'enfance
□ Depuis quelques années
□ Je ne l'utilise pas souvent

5.2. Quelle langue des signes utilises-tu principalement ?
□ LSA (langue des signes algérienne)
□ LSF (française)
□ ASL (américaine)
□ Autre : . . . . . .

5.3. Est-ce que les gens autour de toi (famille, enseignants, amis) comprennent la langue des signes ?
□ Oui, bien □ Un peu □ Non

6. Communication et difficultés

6.1. En dehors de l'école (les magasins, l'hôpital…), est-ce que tu arrives à bien communiquer ?
□ Oui □ Avec aide (papier, gestes, téléphone…) □ Non, c'est difficile

6.2. Est-ce que tu aimerais avoir des outils pour mieux parler avec les personnes qui ne connaissent pas la langue des signes ?
□ Oui beaucoup □ Un peu □ Non

7. Ton avis sur les outils technologiques

7.1. Connais-tu des applications ou objets qui aident à traduire la langue des signes ?
□ Oui □ Non
Si oui, lesquels ? . . . . . .

7.2. Qu'est-ce qui est le plus important pour toi dans un outil de communication ?
□ Qu'il soit simple à utiliser
□ Qu'il respecte la vraie langue des signes
□ Qu'il fonctionne partout (école, ville, maison…)
□ Autres : . . . . . .

8. Tes idées et tes rêves

8.1. Si tu pouvais inventer un objet pour mieux communiquer, ce serait quoi ?
. . . . . .
. . . . . .

Résumé

SignLens est un système innovant de lunettes intelligentes conçu pour la reconnaissance et la traduction en temps réel de la langue des signes, grâce à un modèle d'apprentissage profond embarqué. Équipé d'une caméra et d'un microcontrôleur Raspberry Pi, SignLens capture et analyse les gestes des mains à l'aide d'un modèle d'intelligence artificielle afin de les convertir en texte lisible ou en parole synthétisée. Cette solution portable et économique vise à faciliter l'accès à la communication des personnes sourdes et muettes dans les milieux éducatifs, médicaux et professionnels, favorisant ainsi l'inclusion sociale et l'autonomie.

Mots-clés : Reconnaissance de langue des signes, Traduction de la langue des signes, lunettes intelligentes, technologie d'assistance, apprentissage profond, réseau neuronal convolutionnel, Raspberry Pi, EfficientNet B0, EfficientNet V2S, MobileNet V2, Perceptron Multicouche, MLP
.

Abstract

SignLens is an innovative smart glasses system designed to enable real-time recognition and translation of sign language using embedded deep learning models. Integrating a camera and a Raspberry Pi microcontroller, SignLens captures and analyzes hand gestures through an artificial intelligence model to convert them into readable text or synthesized speech. This portable and cost-effective solution aims to improve communication accessibility for deaf and mute individuals across educational, medical, and professional environments, promoting social inclusion and autonomy.

Keywords : Sign language recognition,Sign language translation, smart glasses, assistive technology, deep learning, convolutional neural network ,Raspberry Pi, EfficientNet B0, EfficientNet V2S, MobileNet V2, Multilayer Perceptron , MLP .

الملخص

نظام SignLens هو نظارة ذكية مبتكرة تهدف إلى تمكين التعرف والترجمة الفورية للغة الإشارة باستخدام نماذج التعلم العميق المدمجة. يعتمد النظام على كاميرا متصلة بوحدة تحكم صغيرة من نوع Raspberry Pi ، حيث يلتقط الإيماءات اليدوية ويحللها عبر نموذج ذكاء اصطناعي لتحويلها إلى نص مقروء أو كلام مسموع. وقد صُمم هذا الحل المحمول ومنخفض التكلفة بهدف تحسين سُبل التواصل للأشخاص الصمّ و البكم في مختلف البيئات التعليمية والطبية والمهنية، مما يساهم في تعزيز اندماجهم الاجتماعي واستقلاليتهم.

الكلمات المفتاحية : التعرف على لغة الإشارة، ترجمة لغة الإشارة ، النظارات الذكية، التكنولوجيا المساعدة ، التعلم العميق، الشبكات العصبية، Raspberry Pi ، MobileNet V2S ، EffecientNet V2S ، EfficientNet B0 ، MLP ، Multilayer Perceptron .