République Algérienne Démocratique et Populaire
Université Abderrahmane Mira - Béjaia -
Faculté des Sciences Exactes Département d'Informatique

Université A/MIRA-BEJAIA
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Fin d'Etude

En vue de l'obtention d'un Master académique en Informatique
Option : Intelligence Artificielle

Thème

Système de reconnaissance de la langue des signes

Réalisé par : Mlle. IARICHEN YASMINE

Devant le jury composé de :

Président	Dr Houda ELBOUHISSI	Maitre de conf. A	U. A/Mira Béjaia.
Examinateur	Mr Zahir AITMATEN	Doctorant	U. A/Mira Béjaia.
Encadrante	Dr Soraya ALOUI	Maitre de conf. A	U. A/Mira Béjaia.
Co-encadrante	Dr Malika YAICI	Maitre de conf. B	U. A/Mira Béjaia.

Béjaia, Septembre 2022.

⁂ *Remerciments* ⁂

La réalisation de ce mémoire a été possible grâce à plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je remercie mes très chers parents, qui ont toujours été là pour moi. Je remercie ma soeur Amal, mon frère Khalil, pour leurs encouragements.

J'adresse mes sincères remerciements à Yanis B qui fut le premier à me faire découvrir le sujet qui a guidé mon mémoire ainsi que sa famille, Fazou, mes cousins et ma tante Hakima et son mari qui, par leurs paroles, leurs écrits, leurs conseils ont guidé mes réflexions et n'ont cessé de m'encourager et m'ont apporté leur soutien moral et intellectuel tout au long de ce projet. Je remercie aussi les membres du jury Dr Houda ELBOUHISSI et Mr Zahir AITMATEN qui me ferront l'honneur de lire et d'évaluer ce traveil, ainsi que Dr Soraya ALOUI et Dr Malika YAICI qui m'ont encadrée dans ce projet.

⁂ *Dédicaces* ⁂

Nous dédions notre travail à : Les deux êtres les plus chères au monde, les bougies qui m'ont toujours guidé vers le bon chemin : Ma mère, la lumière de notre vie, qui a tout fait pour notre réussite et notre bonheur. Mon père, à lui, nous devons offrir tout le respect et l'amour pour son soutien et sa tendresse.

Mlle. Yasmine IARICHEN

Table des matières

Liste des figures

Liste d'acronymes

2D Two Dimentional. 14

3D Three Dimentional. 14

CNN convolutional neural network. iv, 28

LS Langage des signes. 4, 5

LSF Langue des Signes Francaise. iv, 7–9, 12

LSTM Long-Short Time Memory. 29, 39

RNN reccurent neural network. 28, 29

SSD Single-Shot Detector. 36, 38

Introduction Générale

La communication est essentielle dans la construction d'une nation. Une bonne communication conduit à une meilleure compréhension et englobe tous les membres de la communauté, y compris les malentendants.

Partout où les personnes malenendantes ont l'occasion de se rassembler et d'interagir régulièrement, une langue des signes est née. En règle générale, les personnes sourdes représentent un très faible pourcentage de la population (trois enfants sur 1 000 en Algérie souffrent de surdité) selon le service de presse algérien. La langue des signes, en tant que forme différente de la langue de communication, est importante pour de grands groupes de personnes dans la société. Ils existent différents signes dans chaque langue des signes avec une variabilité dans la forme de la main, le mouvement des mains, les expressions faciales ..etc [35]. Il y a une idée fausse selon laquelle les langues des signes sont simplement des gestes avec des règles simples, en fait ce n'est pas le cas.

La reconnaissance visuelle de la langue des signes vise à aider les personnes malentendantes à communiquer avec les autres et à permettre aux personnes entendantes de comprendre les personnes sourdes. Ainsi, la reconnaissance visuelle du langage des signes est un domaine de recherche complexe. De nombreux modèles ont été proposés par différents chercheurs avec une amélioration significative par les approches d'apprentissage profond ces dernières années.

Le but de notre système est qu'il servira d'outil de communication pour les personnes malentendantes pour qu'elles soient comprises par les personnes qui ne connaissent pas la langue des signes. Les technologies d'intelligence artificielle jouent un rôle important dans la suppression des barrières de communication des personnes sourdes ou malentendantes avec d'autres communautés, contribuant de manière significative à leur inclusion sociale et les aidant à communiquer avec des personnes non sourdes[14]. Ces dernières années, l'implication de techniques basées sur la vision par ordinateur est devenue plus populaire, dont l'entrée provient de la caméra (caméra Web, caméra stéréo ou caméra 3D).

Dans mon projet, je présenterai ma solution en développant une application WEB qui reconnaît et traduit le geste de la main en mots, cela permettra aux personnes malentendantes de communiquer plus facilement.

Dans le premier chapitre, nous plongeons brièvement dans les définitions de base et les informations à connaître sur la langue des signes, dans ce projet nous travaillerons avec la langue des signes français.

Ensuite, dans le deuxième chapitre, nous discuterons de la vision par ordinateur et de son impact dans le futur, avec quelques informations importantes et essentielles à connaître sur ce domaine et sa relation avec l'apprentissage automatique.

Enfin dans le dernier chapitre, nous aborderons l'implementation de cette application ainsi que les outils et modèles utilisés pour sa réalisation.

On termine par une conclusion générale, et des perspectives en évoquant les problèmes rencontrés lors de la réalisation de ce projet.

Chapitre 1

Généralités sur la langue des signes

1.1 Introduction

Comme l'a enoncé Platon [44] *« Si nous n'avions point de voix ni de langue et que nous voulussions nous montrer les choses les uns aux autres, n'essaierions-nous pas, comme le font en effet les muets, de les indiquer avec les mains, la tête et le reste du corps ? »*

Pendant longtemps, les personnes malentendantes, isolées, n'ont pas pu enrichir leurs langues des signes et n'ont dû communiquer qu'avec des gestes simplistes.

A l'heure actuelle, il existe des centaines de langues des signes qui sont apparues spontanément au fil du temps partout où existent des communautés de personnes malentendantes. Presque chaque pays a sa propre langue des signes et même chaque région a sa propre langue des signes [38].

Selon la Fédération mondiale des personnes atteintes de surdité, il existe plus de **300 langues des signes** dans le monde que 70 millions de personnes sourdes utilisent. [19].

1.2 Definitions

1.2.1 Linguistique

La linguistique est l'étude scientifique du langage humain.

1.2.2 Langage

Le langage est la capacité d'exprimer une pensée et de communiquer au moyen d'un système de signes (vocaux, gestuels, graphiques, tactiles...) [5]. Les malentendants, de par leur handicap auditif, utilisent un système de signes gestuels pour communiquer : utilisation des mains, du corps et des expressions du visage.

1.2.3 Langue

Une langue est un système de signes vocaux, graphiques ou gestuels spécifiques aux membres d'une même communauté, elles sont donc liées à la culture du groupe qui l'utilise. Par exemple, Les français utilisent la langue française pour communiquer, les anglais utilisent la langue anglaise, les allemands la langue allemande...etc [5].

Par analogie, la communauté des personnes malentendantes utilise la langue des signes, dans ce cas, une personne malentendante francaise utilisera la langue des signes française, une personne malentendante allemande utilisera la langue des signes allemandes. . ..etc [5].

1.2.4 Signe

Chaque phrase est constituée d'une suite de gestes qui utilise les mains, les bras, les expressions faciales, les mouvements de la tête et les postures du corps pour transmettre des messages, le buste et appelés signes [36], qui sont ordonnés pour transmettre un message [7].

Dans la langue orale la communication se fait par un canal auditif-vocal alors que la langue des signes possède un **espace gestuel visuel**.[26]

1.2.5 Langue des signes

La langue des signes LS est le moyen principale de communication entre les **personnes malentendantes**.

De plus, la structure de chaque langue des signes est **indépendante** de la structure de la langue parlée environnante. Par exemple, la langue des signes utilisée dans la communauté malentendante francaise est différente du français parlé [36].

On appelle un signeur, la personne qui communique avec la langue des signes.

1.2.6 Phonétique

La phonétique est une branche de la linguistique qui étudie les phones (les sons) en tant que plus petits segments de la parole.

1.2.7 Phonologie

La branche de la linguistique qui traite des systèmes de sons (incluant ou excluant la phonétique), au sein d'une langue ou entre différentes langues.

1.2.8 Dactylologie

La dactylologie (ou alphabet manuel) est la transcription en signes gestuels de l'alphabet des langues écrites..

1.3 La structure linguistique de la langue des signes

L'expression des phrases en LS ne se réduit pas aux gestes produits par les deux mains, c'est tout le corps qui peut être utilisé pour exprimer une phrase. On peut distinguer trois parties principales impliquées : **les mains**, **la tête (mimique et regard)** et **le buste**.

1.3.1 Les caractéristiques d'un signe

Chaque geste a signer depend de **cinqts paramètres** [7] :

— **La configuration** correspond à la forme de la main définie par les doigts et la paume [27].

Par exemple, pour signer le chiffre 3 : on **ferme** l'anulaire et l'auriculaire , et on **relève** le pouce, l'indexe et le majeur, pour signer le chiffre 4 : on **relève** quatres doigts (l'index, le majeur, l'annulaire et l'auriculaire) et on **ferme** le pouce, pour le chiffre cinqt, on **relève** tous les doigts de la main.

FIGURE 1.1 – Exemple d'une configuration : signation des chiffres trois, quatre et cinqt [7]

— **L'orientation** de la main est définie par deux axes de la main (haut, bas, droite, gauche, vers soi, vers l'interlaucuteur) [27].
Par exemple, pour remercier une personne, l'orientation de la main part de soi vers l'interlaucuteur.

FIGURE 1.2 – Exemple d'une orientation du signe : signation du mot "merci" [21]

— **Le mouvement** correspond à la trajectoire décrite par la main (ligne, arc de cercle...) [27]. Par exemple, pour signer le mot "content", le signeur fait un mouvement circulaire ou niveau du buste en ouvrant sa main.

FIGURE 1.3 – Exemple d'un movement : signation du mot "content" [21]

— **L'emplacement** est la position de la main par rapport au corps [27].
Par exemple, pour signer le mot "anxiété/stresse", le signeur place sa main au niveau de son front, en realisant un mouvement circulaire.

FIGURE 1.4 – Exemple d'emplacement : signation du mot "anxiété/stresse" [21]

— **L'expression du visage** est representée par des mouvements de sourcils, des yeux ou encore de la bouche qui, vont être importants pour exprimer des notions : interrogation, sentiments, intensité...etc [27]. Par exemple, pour signer le mot "colère", les sourcils du signeur s'abaissent et ses lèvres se pincent en levant ses deux mains jusqu'au niveau de ses épaules.

FIGURE 1.5 – Exemple d'expression du visage : signation du mot "anxiété/stresse" [21]

1.4 Langue des signes française (LSF)

1.4.1 Historique :

L'**Abbée de l'Epée** fut, en 1760, le premier entendant français à s'intéresser aux modes de communication entre malentendants. En observant un couple de jumelles sourdes communiquer entre elles par gestes il découvre l'existence d'une langue des signes. Il décide de s'appuyer sur cette langue pour instruire les enfants sourds [20].

Par ailleurs, il regroupe les enfants malentendants pour les instruire et ouvre une école pour malentendants qui deviendra l'Institut national des jeunes malentendants appellé l'Institut Saint-Jacques situé à Paris [20].

L'abbée de l'Epée est aujourd'hui une figure historique de l'histoire de la langue des signes francaise et des malentendants. Sa figure est connue des malentendants dans le monde entier [20].

1.4.2 L'alphabet de la langue des signes française :

LSF a son propre alphabet et sa propre grammaire. Son alphabet manuel sert à épeler les noms propres et les mots qui n'existent pas en langue des signes [19].

La figure ci-dessous montre les différentes signations de l'alphabet de LSF.

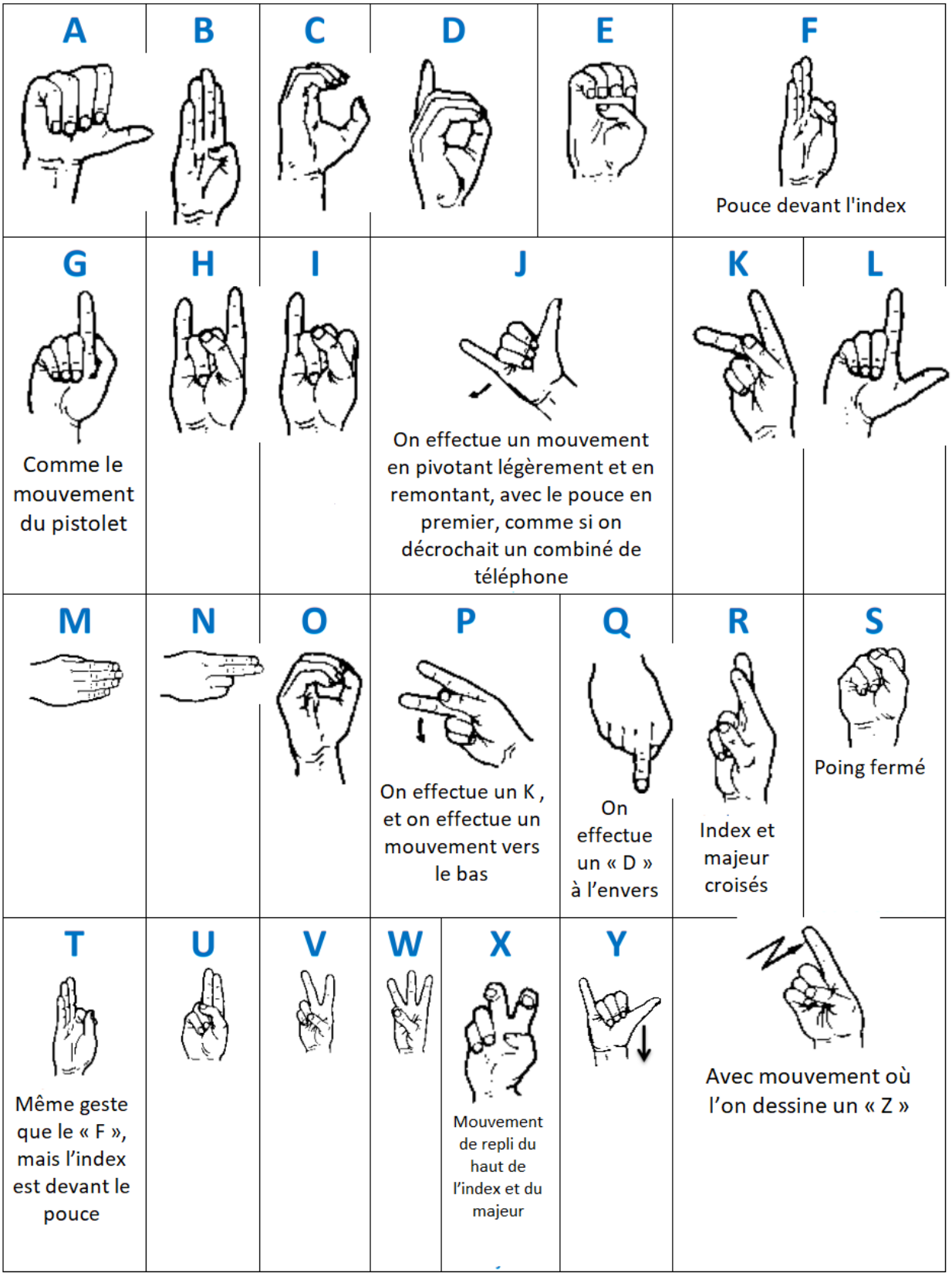

FIGURE 1.6 – Alphabet de LSF [7]

1.4.3 Les nombres en langue des signes française :

— Pour signer **les chiffres de 1 à 9**, il suffit de compter sur ses doigts :

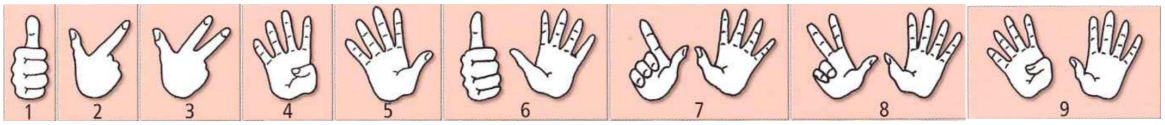

FIGURE 1.7 – Signation des chiffres de 1 à 9 en LSF [9]

— **Le nombre 10** se réalise avec les 2 mains, en faisant un « O » avec son pouce et son index, les autres doigts restent tendus. Puis on réalise un mouvement vers le bas en tournant, de sorte que l'index et le pouce se retrouvent derrière les autres doigts :

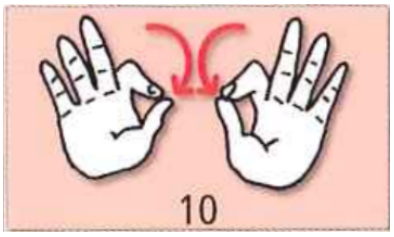

FIGURE 1.8 – Signation du nombre 10 [9]

— **De 11 à 15**, on montre le 1 en le remontant de bas en haut. On fait de même pour 12 on montant le 2, le 13 en montant le 3, le 14 en montant le 4 et le 15 en montant le 5 en gardant le même mouvement :

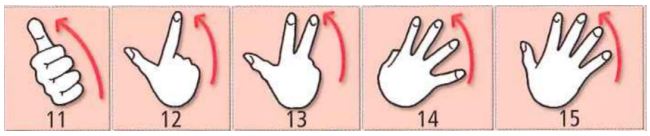

FIGURE 1.9 – Signation des nombres de 11 à 15 [9]

— **De 16 à 19** , c'est l'inverse, on descend les doigts.Pour le 16, on descend le 6, pour le 17, on descend le 7, pour le 18, on descend le 8, pour le 19, on descend le 9 :

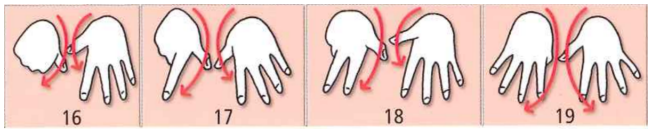

FIGURE 1.10 – Signation des nombres de 16 à 19 [9]

— **Pour le 20**, on joint le pouce et l'index en les gardant tendus, les autres doigts seront fermés :

FIGURE 1.11 – Signation du nombre 20 [9]

Pour 21, on montre 20, puis 1. Pour 22, on montre 20, puis 2, etc jusqu'à 29.

— **Pour 30**, on montre 3 doigts (le pouce, l'index, le majeur), que l'on ferme en « bec d'oiseau ». Puis 31 à 39 se font de la même façon que la méthode pour 21 à 29 : on montre un 30, puis un 1 pour 31, etc.

FIGURE 1.12 – Signation du nombre 30 [9]

— **Pour 40**, on ferme les quatres doigts :

FIGURE 1.13 – Signation du nombre 40 [9]

— **Pour 50**, on tend 5 doigts, en baissant par la suite que le pouce et l'index pour faire un « bec d'oiseau » :

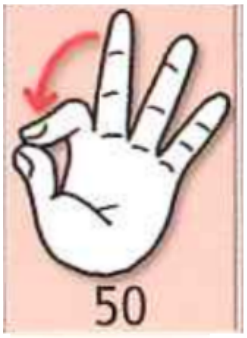

FIGURE 1.14 – Signation du nombre 50 [9]

Pour 60, on montre 6, puis on replie les 6 doigts, sans bouger les mains. Pour 70, on montre 7 doigts, puis on les replie. Etc jusqu'à 90.

Pour 61 et 81, on montre 60, puis 1, on montre 80, puis 1. Etc. Par contre, pour 71 et 91, on montre 70, puis 11, 90 puis 11 ! Etc jusqu'à 79 et 99. Exemples :
71 : 70 et 11
72 : 70 et 12
96 : 90 et 16...

— **Pour 100**, on montre un « C » que l'on fait glisser de la gauche vers la droite :

FIGURE 1.15 – Signation du nombre 100 [9]

— **Pour 1000**, on place sa main gauche latéralement, droite, devant soi. Puis on vient faire taper sa main droite repliée au centre de la main gauche immobile :

FIGURE 1.16 – Signation du nombre 1000 [9]

Pour exprimer par exemple 1997, on montre 1000, puis 9, puis cent, puis 90, puis 17.

1.4.4 La grammaire de la LSF

La LSF étant une langue à part entière, elle possède sa propre syntaxe, sa propre grammaire et n'est pas une traduction pure et simple du francais parlé. L'ordre des mots est différent entre le francais signé et le francais parlé [43]. Par example :

— Le locuteur français dira : « Ravi(e) de vous rencontrer » en mettant les mots dans cet ordre.

— Le signeur signera le mot **"Ravi"** en posant sa main droite sur sa main gauche en faisant un mouvement de balayage de soi vers l'interlocuteur. Puis procedera a signer le mot **"Rencontrer"** en laissant l'index de ses deux mains tendus, pointant vers le haut et en rapprochant sa main gauche a sa main droite. Enfin, le signeur pointera son index vers l'interlocuteur pour signer le mot **"Vous"** .

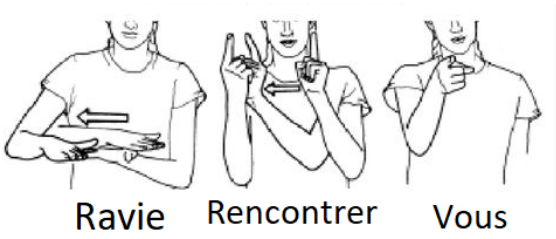

FIGURE 1.17 – Signation de la phrase "Ravi(e) de vous rencontrer [21]

1.5 Conclusion

Au debut de ce chapitre, nous avons abordé l'importance de la langue des signes qui a permis à la communauté des malentendants de se fondre dans la société accompagnée par des définitions importantes : Langage, langue...etc. Ceci nous a poussés à parcourir les differentes caractéristiques de la langue des signes ainsi que l'histoire de la langue des signes francaise et sa grammaire en donnant des exemples de signations de quelques mots et nombres.

Chapitre 2

La visoin par ordinateur

2.1 Introduction

La vision par ordinateur est un domaine de l'informatique qui se concentre sur **l'imitation** de la vision humaine pour **interpréter** des actions, **détecter** des objets et **effectuer** des actions basées sur des images capturées par des appareils photo numériques [16].

2.2 Définitions

2.2.1 La vision

La vision dispose de deux composants : un **dispositif de détection** et un **dispositif d'interprétation**.
Le dispositif de détection capture les données. Pour les humains et les animaux, l'œil capte la lumière à travers l'iris, et pour un ordinateur, c'est une **caméra** qui capture les données d'une manière très similaire à l'œil.

Le dispositif d'interprétation est responsable de l'**analyse des données**, de l'**extraction des informations utiles** et la prise de décisions sur ces informations [34].

2.2.2 La vision par ordinateur :

Aussi appellée "vision artificielle", désigne les différentes techniques permettants aux machines de voir et de comprendre le contenu des images. Il s'agit d'une sous-catégorie de l'intelligence artificielle et du Machine Learning [24] .
De manière générale, les différentes méthodes ont pour but de reproduire la vision humaine.

2.2.3 Sous-domaines de la vision par ordinateur :

Les sous-domaines de la vision par ordinateur comprennent :

— **Prétraitement d'image** traite des fonctionnalités de bas niveau des images.

— **Detection de caractéristiques (features)** fournit une représentation raffinée des images.

— **Segmentation** détecte des parties de l'image qui serviront dans l'interprétation.

— **La reconstruction 3D** créer les modèles 3D d'objets à partir des images 2D.

— **Reconnaissance d'objets** : étiquetter ce qui apparait sur l'image.

— **Analyse de mouvements** traite des objets en mouvement dans les vidéos.

2.2.4 Applications :

Les caméras sont partout et le nombre d'images téléchargées sur Internet augmente de façon exponentielle. La vision par ordinateur est essentielle car ces images doivent être triées pour permettre aux ordinateurs de comprendre leurs contenus [34].

Voici une liste non exhaustive des applications de la vision par ordinateur :

— **Reconnaissance de caractères** : une technologie qui permet de convertir des caractères detectés sur une photo sous forme de texte.

FIGURE 2.1 – Reconnaissances de plaques d'immatriculation [34]

— **Reconnaissance faciale** : Plusieurs cameras sont dotées de cette technologie : Sony, Nikon..etc.

FIGURE 2.2 – Détection de visage à travers la camera [34]

— **Pilotage automatique** : connu sous le nom "véhicules autonomes".

FIGURE 2.3 – véhicule autonome [34]

2.2.5 Pixel

Le pixel est l'élément fondamental et le plus petit élément d'une image numérique appelés "unité" d'une image.

2.2.6 Image numérique

Une image numérique est un **tableau de pixels**, la longueur et la largeur du tableau de pixels définissent la résolution de l'image, et ils peuvent être stockés en mémoire sous différents formats.
Il existe un certain nombre de formats d'images numériques importants et couramment disponibles, brièvement résumés comme suite :

- **Image Bitmap (bmp)** : c'est le format d'image de base, généralement limité, à compression sans perte (des variantes avec perte existent). bmp est né du développement de Microsoft Windows [4].
- **Format d'échange d'image (gif)** : Il s'agit d'un format d'image numérique, à mi-chemin entre une image fixe et une courte vidéo [4].
- **Groupe conjoint d'experts en photographie (jpeg)** : Il désigne un format d'enregistrement et de compression numérique, imaginé et mis au point par un groupe d'experts en compression d'images fixes. C'est dans le domaine de la photographie numérique que l'on est donc amené à rencontrer le plus fréquemment ce format. Il prend alors la forme d'une extension .JPEG, aussi déclinée en .jpeg, .JPG ou .jpg [4].
- **Réseaux graphiques portables (png)** : Png est un format d'image bitmap qui utilise une compression de données sans perte [4].
- **Image vectorielle (svg)** : ce format est couramment employé pour afficher des images, des graphiques et des illustrations 2D (deux dimensions) sur des sites web. Ce format vectoriel permet par ailleurs les agrandissements et les réductions de taille sans perte de résolution [4].
- **Format de fichier d'image taguée (tiff)** : est le format standard pour les photos haute résolution et les données d'impression [4].

2.2.6.1 Types d'image numérique

— **Image binaire** est représentée par un tableau à deux dimensions , où chaque élément du tableau attribue à chaque pixel un nombre parmi 0, 1. Le noir correspond à un 0 (pixel éteint ou de fond) [41].

— **Images d'intensité (niveaux de gris)** sont des tableaux deux dimensions, où chaque élément du tableau attribue une valeur numérique dans la plage de 8 bits de 0 à 255).

— **Image colorée** assigne 3 valeurs numériques à chaque pixel, où chaque valeur assignée correspond à une quantité de couleur pour un canal de couleur particulier (canal d'image rouge ou vert ou bleu) [42].

— **Image Multispectrale** qui contient des informations en dehors de la plage de perception humaine normale comme des données radiographiques, des données ulraviolet [42].

2.2.6.2 Représentation d'image numérique :

L'image numérique consiste en un **ensemble fini de pixels**. Elles peut être représentée par une matrice : chaque élément de la matrice correspondra à une valeur du pixel associé a cet élément avec ses coordonées.

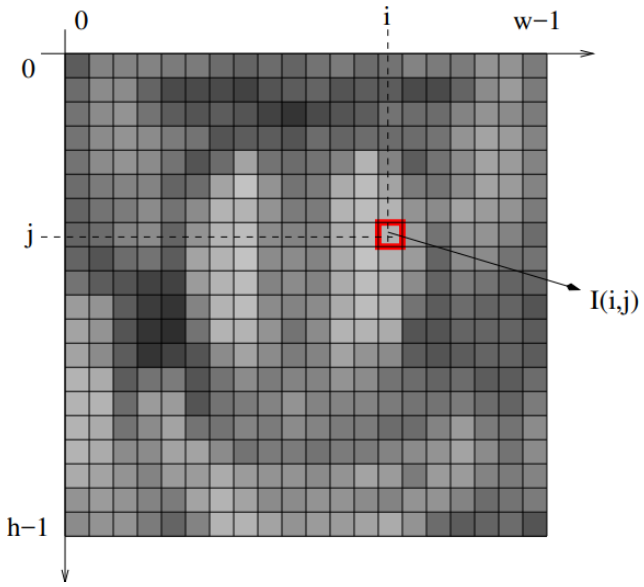

FIGURE 2.4 – Exemple d'une representation matricielle d'une image numerique [2]

Sur la figure ci-dessus (l'image à niveaux de gris), chaque pixel a une valeur entière comprise entre 0 et 255. Une valeur de 0 correspond au **noir absolu**, une valeur de 255 à **blanc pur**, et les valeurs entre ces extrêmes produisent différents niveaux de gris entre le noir et le blanc.

2.2.7 Vidéo

Une vidéo est un **flux d'images**, elle a une fréquence d'images représentant le nombre d'images lues chaque seconde (généralement entre 24 et 60 images par seconde).

Elle est stockée en mémoire sous différents formats parmi lesquels :

— **MP4** : pris en charge par la plupart des appareils et plates-formes numériques, ce qui en fait le format vidéo le plus universel. Le MP4 peut stocker des données vidéo, des données audio, du texte et des images fixes. De plus, ils peuvent conserver une qualité vidéo élevée tout en conservant des tailles de fichiers relativement petites.

— **MOV** : développé par Apple, MOV est le format vidéo spécifiquement conçu pour Quick-Time Player. Les fichiers MOV se révèlent souvent de meilleure qualité et de plus grande taille contrairement au format MP4.

— **WMV** : c'est le format de compression vidéo de Windows Corporation. Il permet de réduire significativement la taille des fichiers volumineux tout en conservant leur qualité.

— **AVI** : l'un des formats vidéo les plus anciens, le format vidéo AVI est l'un des formats vidéo les plus polyvalents, compatible avec Windows, Mac et Linux et pris en charge par la plupart des navigateurs Web.

— **MKV** : format de fichier pouvant contenir un nombre illimité de pistes vidéo, audio, image ou de sous-titres dans un seul fichier. C'est un format universel pour stocker du contenu multimédia commun, comme des films ou des émissions de télévision.

2.3 Traitement d'images numériques

Le traitement d'images est un sous-domaine du traitement de signal dédié aux images et vidéos [31]. C'est l'ensemble des opérations effectuées sur l'image, afin d'en améliorer la lisibilité et d'en faciliter l'interprétation.

En vision artificielle, le traitement d'images se place après les étapes **d'acquisition et de numérisation** .

— **L'aquisition :** Pour acquérir des images numériques on utilise des dispositifs comme les scanners, les appareils photo, les caméscopes numériques...etc.

— **La numérisation :** La numérisation est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal qu'elle représente (une infinité de valeur dans l'intensité lumineuse par exemple), à l'état d'image numérique qui est caractérisée par l'aspect discret (l'intensité lumineuse ne peut prendre que des valeurs quantifiées en un nombre fini de points distincts).

En vision par ordinateur, le traitement de l'image necessite une variété de techniques qui peuvent être separées en deux grandes categories [42] :

— **Techniques de prétraitement de l'image** : les entrées et les sorties sont des images.

— **Techniques d'analyse de l'image** : les entrées sont des images et les sorties sont les attributs néccesaires extraits de ces images [42].

Dans le cadre de ce projet, nous allons nous concentrer sur les techniques d'analyse de l'image.

2.3.1 Techniques du prétraitement d'images digitales

Les techniques du prétraitement de l'image inclues : l'amélioration de l'image, la restauration de l'image et la compression de l'image.

2.3.1.1 Amelioration de l'image

En vision par ordinateur, l'amélioration de l'image a pour but de mettre en valeur des details obscures, ou simplement mettre en lumière des détails interessants de l'image [42]. Il existe plusieurs techniques d'amélioration de l'image, parmi elles nous citons : égalisation de l'histogramme de l'image (un histogramme noté **H(k)** est la distribution des niveaux de gris dans une image), étirement de l'histogramme...etc [42].

2.3.1.2 La restauration de l'image

Dans le processus d'acquisition ou de numérisation, une image peut subir des dégradations appelés **"bruits"** [39]. La restauration de l'image est donc nécessaire pour pallier aux problèmes de dégradation de l'image. Il existe plusieurs techniques de retauration de l'image, parmi elles : le filtrage (filtres médiane..etc), par le gradient..etc [12].

2.3.1.3 Compression d'images

La compression d'images est une application de la compression de données sur des images numériques, son but est de reduire leur coût de stockage ou de transmission.

2.3.2 Techniques d'analyse d'images numériques

Les techniques d'analyse d'images numériques inclues : la segmentation de l'image, l'extraction des caractéristiques et la reconnaissance de l'image.
Ces techniques ont pour but de donner une représentation symbolique de l'image où un lien avec la scène observée.

2.3.2.1 la segmentation

La segmentation est l'une des techniques d'analyse d'images qui sert a **partionner** l'image en zones homogènes selon un critère déterminé : couleur, texture, niveau de gris...etc [11].

La segmentation est très utile pour la reconnaisance faciale ou d'objet ou même pour la détection de cellules sur des images biometriques [42].

Les techniques de segmentation sont réparties en **3 grandes classes classiques** : segmentation par seuillage, segmentation par contours et segmentation par régions. La méthode

la plus récente est la segmentation par Deep Learning. Dans le cadre de ce projet, c'est la segmentation par Deep Learning qui est utilisées.

2.3.2.2 La segmentation par seuillage

Le seuillage est le traitement permettant de selectionner les informations significatives d'une image, ce traitement nécessite le reglage d'un parametre appellé **"Seuil S"**.

L'opération du seuillage consiste à décider de **l'appartenance d'un pixel à un objet**. **Le Seuil :** le niveau de gris référence qui orientera la décision qu'un niveaux de gris appartient à l'objet ou non. Il est calculé à partir de l'histogramme de l'image.

Il existe plusieurs techniques pour determiner la valeur du seuillage telles que : seuillage globale (un seul seuil) , seuillage locale (plusieurs seuils), seuillage recursive, seuillage dynamique, les champs de markov ...etc [11].

2.3.2.3 La segmentation par contours

Un contour est un ensemble des points d'une image numérique qui correspond à un changement brutal de l'intensité lumineuse [6]. Un contour peut être défini comme **une frontière entre deux régions distinctes**.

Il sépare des régions de niveaux de gris différents et relativement homogènes, ou bien des régions de textures différentes [6].

2.3.2.4 La segmentation par regions

Les méthodes de l'approche région cherchent à différencier les régions en utilisant les propriétés de l'image telles que la couleur, texture, forme...etc. Ces méthodes utilisent principalement les critères de décision pour segmenter l'image en différentes régions selon la similarité des pixels [6].

Il existe **3 méthodes** qui permettent la détection de régions :

— **Segmentation par fusion** : Ce type de segmentation permet de sélectionner un pixel (appellé "germe") ou un ensemble de pixels de l'image autour duquel on fait croitre une région. Les régions sont construites en ajoutant successivement à chaque pixel les pixels qui lui sont connexes et qui vérifient un critère de similarité. La croissance s'arrête lorsque tous les pixels ont été traités. Cette méthode procède par un balayage séquentiel de l'image et considère le premier pixel comme un germe. Il tente alors de faire croitre ce germe le plus longtemps possible en y ajoutant les pixels voisins qui verifie un critère de similarité [22].

— **Segmentation par division** : L'image est divisée d'une manière récursive tant que le critère d'homogénéité n'est pas vérifié. Si le critère est vérifié, l'algorithme s'arrête, sinon,

on divise l'image en des régions. Chaque région est testée et redivisée si elle ne valide pas le critère. L'algorithme se termine lorsque toutes les régions sont homogènes [22].

— **Segmentation par fusion/division** : Ces méthodes combinent les deux méthodes citées précédemment, cette méthode commence par diviser l'image en petites régions homogènes, puis fusionne les régions connexes et similaires en respectant un prédicat de regroupement [22].

2.3.3 Extraction des caractéristiques

Les indices visuels (caractéristiques sont des objets extraits de l'image qui contiennent de manière concise une **information importante**, pour son analyse.

2.3.3.1 Filtrage

Les filtres permettent aussi d'extraire des caractéristiques, cela est réalisé grâce à l'opération de convolution.

La convolution est l'application d'un filtre (ou kernel) d'une taille précise qui va balayer toute l'image de haut en bas, de gauche a droite qui va resulter une image filtrée appellé feature map (ou carte d'activation) dans le cas du Deep Learning. On obtient alors pour chaque paire (image, filtre) une feature map, qui nous indique où se situent les caractéristiques dans l'image.

Ces caractéristiques peuvent correspondre a des lignes, des courbes, des coins... etc. Pour chaque feature il existe une multitude de filtres qui permettent de la detecter. Par exemple pour la detection de contours il existent plusieurs filtres pour cela : le filtre Sobel, Prewitt, Laplacian... etc.

2.3.3.2 Squelittisation

On appelle "squelettisation", une méthode d'extraction de primitives (features) qui implémente la squelettisation [37].

La squelettisation consiste à amaigrir une forme jusqu'à en obtenir un ensemble de courbes centrées. L'ensemble obtenu est alors appelé squelette ou « axe médian ». L'axe median d'une region R avec une bordure B est décrit comme suit : pour chaque point p dans R, on definit ses plus proche voisin dans B, si p a plus d'un voisin, alors p appartient a l'axe mediane (le squelette) [37].

Il existe deux approches permettant de construire des squelettes à partir de formes parmis elle : squelletisation par amincissement et squelletisation par transformée de distance.

— **Squelletisation par amincissement :** consiste a supprimer les pixels au niveau de la bordure de la région à chaque itération jusqu'a ce qu'il n'ait plus de pixels a supprimer qui perturbe la connectivité [37].

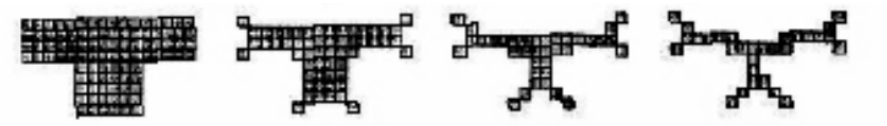

FIGURE 2.5 – Squelettisation par amincissement [37]

— **squelletisation par transformée de distance :** le processus se déroule ainsi : L'image d'origine (binaire) est convertie en éléments caractéristiques et non-caractéristiques. Les éléments caractéristiques appartiennent à la limite de l'objet. La carte de distance est générée où chaque élément donne la distance à l'élément caractéristique le plus proche. Les crêtes (extrêmes locaux) sont détectées comme des points squelettiques [37].

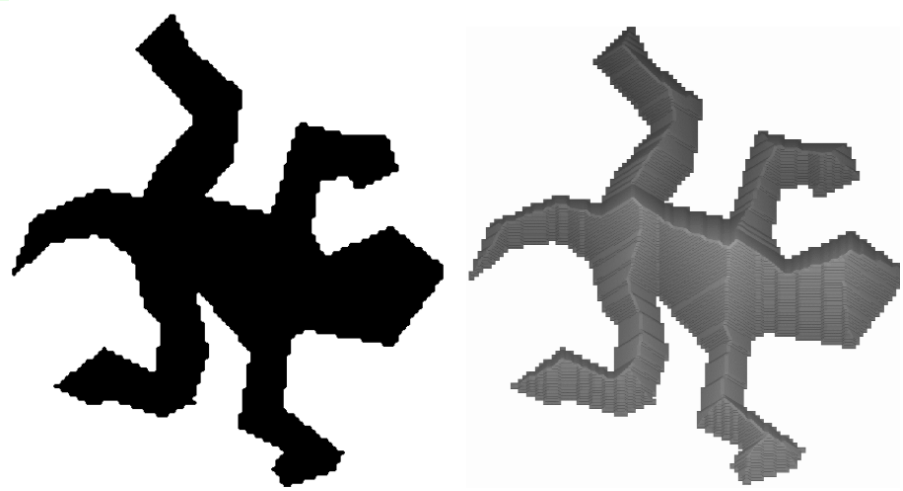

FIGURE 2.6 – Squelettisation par transfromée de distance [37]

2.3.4 La reconnaissance d'images et la detection d'objet

La reconnaissance d'images et la detection d'objets sont des techniques très similaires. Cependant, la reconnaissance d'images identifie l'objet qu'il ya dans l'image globale. La detection d'objets identifie et localise l'objet ou un ensemble d'objets qui se retrouvent dans une images.

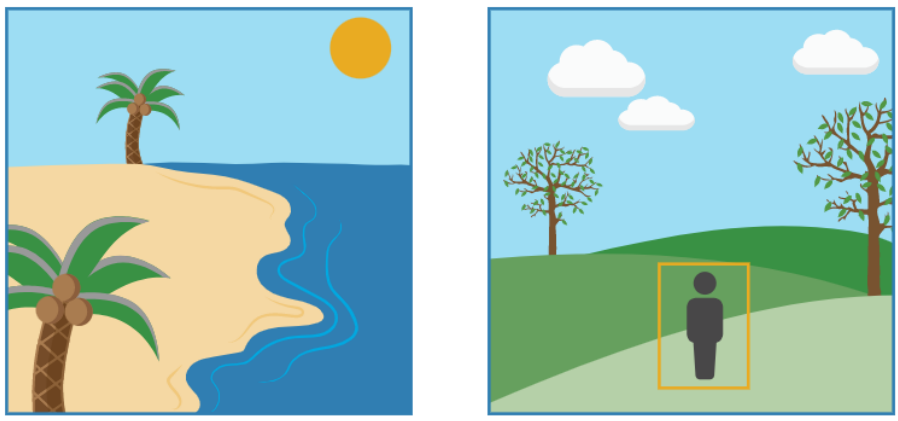

FIGURE 2.7 – Reconnaissance d'images (à gauche) et la detection d'objets (à droite) [**?**]

L'objectif général de l'analyse d'images est de construire un système interactif qui peut attribuer des descriptions symboliques à une scène [31]. Ceci est realisé grâce a l'intelligence artificielle et les modéles qui en découlent de celle ci.

2.4 La vision par ordinateur et l'apprentissage automatique

Tout d'abord, L'apprentissage automatique (ou artificiel) (machine learning en anglais) est l'un des champs d'étude de l'intelligence artificielle.

L'apprentissage automatique fait référence au **développement**, **l'analyse et l'implémentation de méthodes** qui permettent à une machine d'évoluer et de remplir des tâches associées à une intelligence artificielle grâce à un processus d'apprentissage. Cet apprentissage permet d'avoir un système qui s'optimise en fonction de l'environnement, les expériences et les résultats observés [1].

Cette notion englobe toute méthode permettant de construire un modèle de la réalité à partir de données, soit en améliorant un modèle, soit en créant complètement le modèle [1].

Les solutions d'apprentissage automatique et de vision par ordinateur s'articulent autour de trois points principaux :

— **Collecte de données**
— **Entrainement du modèle**
— **Faire des predictions en utilisant le modèle entrainé**

Il existe également quatres approches différentes de l'apprentissage automatique et de la vision par ordinateur : supervisée, non supervisée et semi-supervisée et apprentissage par renforcement.

— **Apprentissage supervisé :** a des données d'apprentissage étiquetées. Si les classes sont prédéterminées et les exemples sont connus, le système apprend à classer selon un modèle de classement.

— **Apprentissage semi-supervisé :** a une partie des données étiquetées, Il est mis en œuvre quand des données (ou « étiquettes ») manquent [1].

— **Aprentissage non supervisé :** a des données non étiquetées (où se situent la plupart des problèmes du monde réel), c'est à dire, quand le système ou l'opérateur ne dispose que d'exemples, mais non d'étiquettes, et que le nombre de classes et leur nature n'ont pas été prédéterminés, on parle d'apprentissage non supervisé (ou clustering) [1].

— **Apprentissage par renforcement :** L'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme [1].

Il existe plusieurs algorithmes d'apprentissage automatique, nous citons quelques uns :

— Les machines à vecteurs support (SVM)
— Les réseaux de neurones pour un apprentissage supervisé ou non-supervisé
— La méthode des k plus proches voisins
— Les arbres de décision
— La régression logistique (classification)
— La logique floue
— Les algorithmes génétiques.

Ces méthodes sont souvent combinées pour obtenir diverses variantes d'apprentissage. L'utilisation de tel ou tel algorithme dépend fortement de la tâche à résoudre (classification, estimation de valeurs, etc.).

2.4.1 Facteurs de pertinence et d'efficacité

La qualité de l'apprentissage et de l'analyse dépend de la complexité du modèle, l'efficacité de l'apprentissage depend aussi de la base de données utilisée : si le nombre d'exemples est petit l'analyse sera difficile, mais plus il y en a, plus le besoin de mémoire informatique est élevé et plus longue est l'analyse [1].

2.4.2 La classification en pratique

La machine d'apprentissage, dont la tâche est d'apprendre une fonction à travers **des exemples** ne nous indique nullement comment obtenir un classificateur adapté à la tâche considérée [1]. Les différentes étapes que l'approche Machine Learning préconise pour atteindre un tel objectif sont :

2.4.2.1 Le Choix de l'ensemble de données

Pour obtenir un resultat non faussé, il est crucial de mesurer l'erreur sur des exemples qui n'ont pas servi à entraîner le modèle [1]. Pour cela, on divise l'ensemble des données disponibles en deux parties :

- **Un sous ensemble d'entraînement**, dont les données serviront à l'apprentissage(ou entraînement) du modèle [1].
- **Un sous ensemble de test**, dont les données seront utilisées uniquement pour évaluer la performance du modèle entraîné [1].

- **Un sous ensemble de validation** un ensemble de validation est utilisé pour «régler les paramètres» d'un classificateur pour pallier au problème du sur-apprentissage.

Les différentes opérations que l'on doit effectuer avant de présenter les données à l'algorithme d'apprentissage sont :

2.4.2.2 Acquisition des données

Si les données proviennent d'une source analogique, il faut commencer par les transformer de manière à en avoir une représentation manipulable par un programme informatique. Par exemple une image est transformer en matrice [1].

2.4.2.3 Prétraitement

Cette phase consiste en une succession de traitements sur les données brutes afin d'améliorer l'image globale.

2.4.2.4 Conversion

Il s'agit de convertir les données dans le format spécifié par l'algorithme utilisé lors de la phase de classification. La représentation vectorielle est assez populaire. Dans ce format, les données sont représentées sous forme de vecteurs dont chaque composante correspond à une caractéristique de l'objet [1].

2.4.2.5 Post-traitement

Dans certains cas, on doit normaliser les données dans le format d'entrée. Par exemple, dans le cas d'un réseau de neurones, le mieux est que la moyenne de chaque composante du vecteur sur l'ensemble des exemples d'entrainement soit proche de zéro et que l'ecart-type soit égale à 1.

2.4.2.6 Entraînement ou apprentissage

La phase d'entraînement consiste à trouver une évaluation des paramètres . La sélection de ces paramètres est effectuée par un algorithme d'apprentissage qui reçoit en entrée l'ensemble des données d'apprentissage ainsi qu'un ensemble d'hyperparamètres d'apprentissage [1]. En ce sens, ce sont les données (de l'ensemble d'apprentissage) qui induisent l'apprentissage. L'ensemble des paramètres résultant de l'apprentissage est appelé **modèle**.

2.4.2.7 Evaluation du modèle (test)

Une fois le modèle obtenu, il est intéressant d'évaluer ses performances sur un ensemble indépendant de données : le test set. Cette phase permet de se rendre compte du pouvoir de généralisation du classificateur, c'est à dire sa capacité à obtenir de bons résultats sur n'importe quel ensemble de données provenant de la même distribution [1].

2.4.2.8 Exploitation

Lorsque l'on dispose d'un modèle efficace pour une tâche considérée, on peut utiliser la machine d'apprentissage pour faire des prédictions sur de nouveaux exemples. Un classificateur correspond donc à **une machine entraînée** [1].

Au cours des décennies suivantes, différentes techniques de Machine Learning ont été développées pour créer des algorithmes capables d'apprendre et de s'améliorer de manière autonome.

Dans le cadre de notre projet, nous allons utiliser des modèles de reseaux de neurones artificiels afin de detecter et classifier les signes obtenues.

2.5 Le deep learning et la vision par ordinateur

Le deep learning ou apprentissage profond est un sous-domaine du machine learning, sous-domaine de l'intelligence artificielle.

C'est sur les réseaux de neurones que se base le Deep Learning, mais aussi des technologies comme la reconnaissance d'images ou la vision robotique. Les réseaux de neurones artificiels sont inspirés par les neurones du cerveau humain. Ils sont constitués de plusieurs neurones artificiels connectés entre eux. Plus le nombre de neurones est élevé, plus le réseau est « profond » [8].

2.5.1 Les réseaux de neurones

Les réseaux de neurones ont été développés comme un modèle mathématique générique afin de modéliser les neurones biologiques. Ils comportent un certain nombre d'éléments de traitement d'information appelés neurones. C'est une imitation algorithmique des fonctions du cerveau humain [8].

2.5.1.1 Un neurone

Dans un réseau, le neurone est une unité qui reçoit l'information, procède à des calculs simples, et la transmet à une autre unité [8]. On distingue trois types de neurones dans un réseau artificiel :

— Les neurones d'entrées qui reçoivent les données du monde extérieur.

— Les neurones de traitement (les couches cachées).

— Les neurones de sorties.

2.5.1.2 Les poids et biais

Les connexions entre neurones dans un réseau artificiel ne sont pas équivalentes. Chaque connexion a en effet un poids spécifique (weight en anglais) qui influence la transmission de l'information d'une unité à une autre [8].

2.5.1.3 La fonction d'activation

Formule mathématique, la fonction d'activation permet à chaque neurone de normaliser les données d'entrée qu'elle reçoit avant de les transmettre [8].

2.5.1.4 Structure d'un reseau de neurones

Un réseau de neurones est en général composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (i) est composée de Ni neurones, prenant leurs entrées sur les Ni-1 neurones de la couche précédente [8].

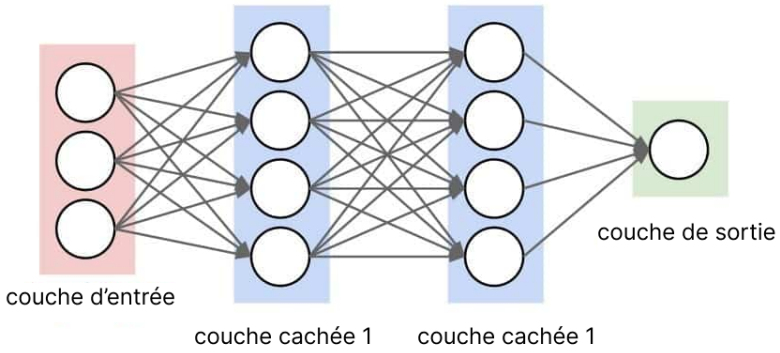

FIGURE 2.8 – Structure d'un réseau de neurone [23]

2.5.2 L'architecture des reseaux de neurones

Les réseaux de neurones artificiels peuvent être classifiés en 4 grandes familles.

2.5.2.1 Les réseaux de neurones feed-forward

Dans une architecture de type feed-forwarded, l'information est propagée uniquement vers l'avant. Cette famille de réseaux neuronaux comprend à son tour deux catégories de réseaux de neurones : le perceptron simple et le perceptron multicouches [18].

— Le perceptron simple est un réseau de neurones qui ne possède que deux couches de neurones : une couche en entrée et une couche de sortie. Les deux couches sont directement reliées entre elles, ce qui fait que le réseau ne possède qu'une seule matrice de poids [18].

— Le perceptron multicouche quant à lui possède entre la couche d'entrée et celle de sortie une ou plusieurs couches cachées. Il intègre plusieurs matrices de poids [18].

Il est possible, pour le traitement de données très complexes, de créer des neural networks distincts qui vont traiter chacun une partie de l'information. On parle dans ce cas de Convolutional neural networks ou « réseaux neuronaux convolutifs » [18].

— **Les réseaux de neurones convolutifs (CNN)**

Ils sont des réseaux feed-forward qui correspondent à un empilement de perceptron multicouche. Chacun traitant une portion de l'information globale. Ces réseaux sont surtout

utilisés pour la reconnaissance d'images, de vidéos [10].

Un réseau de neurones convolutif contient une "partie convolutive" qui, fonctionne comme un extracteur de caractéristiques (features) des images. Une image est passée à travers une succession de filtres, créant de nouvelles images appelées feature maps (ou cartes de convolution) [10]. Chaque filtre est donc spécialisé pour la detection d'une caracteristique précise. Avec la profondeur du traitement, les couches proches de l'entrée ont tendance à avoir moins de filtres et les caractéristiques sont plus souvent simplistes tandis que les couches plus proches de la sortie ont des caractéristiques plus complexes .

Par exemple, pour détecter une image d'un chat, les premières couches vont detecter des patterns simple comme les lignes, les coins, les textures simple ... etc. Les prochaines couches utiliseront les patterns précédents pour detecter des patterns plus complexes comme par exemple des oreilles, un museau, des pattes ...etc. Par la suite, le réseau de neurone comprendra facilement si un chat se trouve sur une image en recherchant ces caractéristiques.

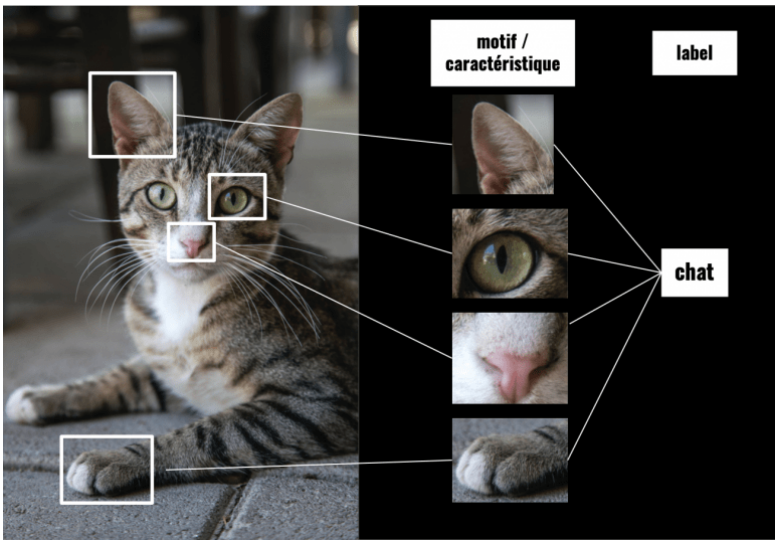

FIGURE 2.9 – Exemple de charactéristiques pour la détéction d'un chat[40]

Cette méthode est bien plus efficace que l'approche classique pour deux principales raisons :
- **Moins d'erreur dans l'apprentissage** car le modèle n'apprend pas des images mais des caractéristiques, des motifs.
- **Plus de précision dans la détection**, car le modèle doit justement reconnaître des caractéristiques, des motifs.

Après chaque étape de convolution, il ya l'étape du **pooling**. Le pooling permet de réduire la taille spatiale d'une feature map, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives. La méthode de pooling la plus connue est le max-pooling qui consiste en une tuile de 2x2. L'image d'entrée est découpée en une série

de rectangles de taille 2x2 qui ne se chevauchent pas. Chaque rectangle prend alors la valeur maximal de chaque partie de l'image (max-pooling). Le max-pooling permet de selectionner que les features qui nous interessent ce qui permet un gain de temps et un gain de complexité spatiale [32].

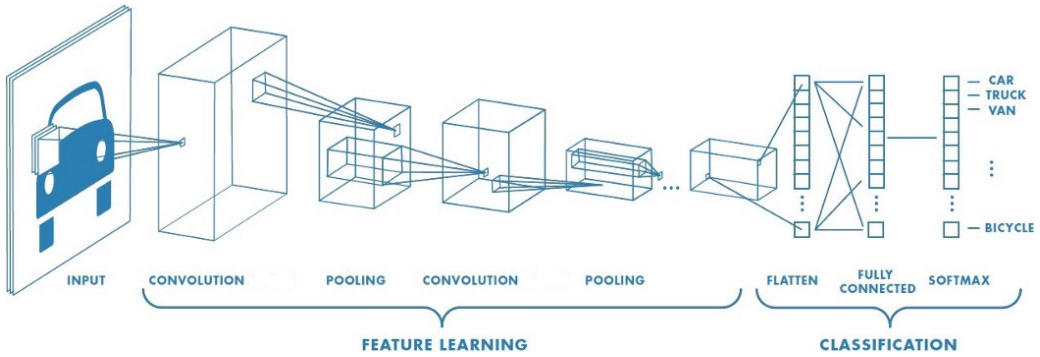

FIGURE 2.10 – Architecture de CNN [33]

Pour classifier l'image, toutes les feature maps sont alors combinées (cette opération s'appelle Flatenning, applatissement en francais) et sont envoyées à une couche dense entierement connectée pour pouvoir classifier l'image grâce à la fonction softmax (une fonction d'activation spécialisée pour les réseaux de classification multi sortie) qui permet d'indiquer la probabilité que x appartienne à chacune des classes de sortie [32].

2.5.2.2 Les réseaux de neurones à résonance

dans un réseau de neurones à résonance, l'activation d'un neurone est renvoyée à tous les autres neurones, ce qui provoque des oscillations (des variations). Cette architecture peut se présenter sous diverses formes, avec un niveau de complexité élevé [10].

2.5.2.3 Les réseaux de neurones récurrents

Les réseaux de neurones reccurents (RNN) utilisent **les sorties précédentes comme entrées supplémentaires** et sont parfaitement adaptés au traitement de données séquentielles, c'est à dire quand les données forment une suite et ne sont pas indépendantes les unes des autres.

Ils capturent la dynamique des séquences temporelles à travers l'utilisation de cycle. En fait, contrairement aux réseaux acycliques, les réseaux de neurones récurrents **"retiennent"** les états antérieurs des nœuds pour que ces états antérieurs (appelés états cachés, hidden state en anglais) puissent impacter la sortie au temps actuel [17]. En d'autres mots, les réseaux de neurones reccurents permettent de modéliser une forme de **mémoire** des états précédents des neurones.

L'état caché dans un réseau de neurone réccurent est comme les couches cachées dans un réseaux de neurones feed-forward habituel, sauf que, cet etat caché est utilisée comme une entrée additionnelle dans le RNN dans l'étape suivante à l'instant suivant.

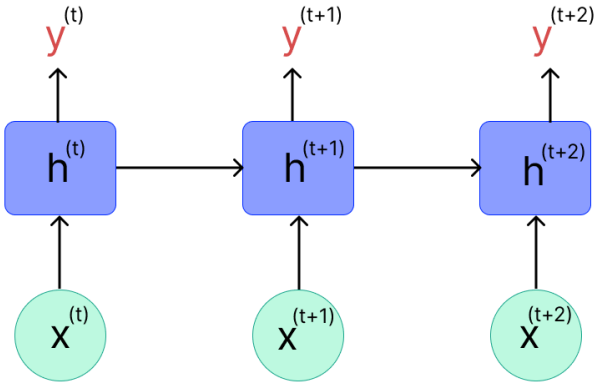

FIGURE 2.11 – Architecture d'un réseau de neurone réccurent

Un simple réseau de neurone réccurent a donc une entrée x, un état caché h (comme entrée additionnelle) et une sortie y a chaque intant t.

Néanmoins, pour les applications faisant intervenir de **longs écarts temporels** (typiquement la classification de séquences vidéo), cette **« mémoire à court-terme » n'est pas suffisante** [25]. En effet, les réseaux de neurones réccurents « classiques » (réseaux de neurones récurrents simples ou Vanilla RNNs) ne sont capables de mémoriser que **le passé dit proche**, et commencent à « oublier » au bout d'une cinquantaine d'itérations environ. Ce transfert d'information rend leur entrainement beaucoup plus compliqué, et ce n'est que récemment que des méthodes efficaces ont été mises au point comme **les LSTM (Long Short Term Memory)** [25].

Ces réseaux à large « mémoire court-terme » ont notamment révolutionné la reconnaissance de la voix par les machines (Speech Recognition) ou même la compréhension et la génération de texte (Natural Langage Processing). C'est l'architecture utilisée dans notre application.

— **Les LSTM (Long Short Term Memory)**

Le LSTM (Long Short-Term Memory), cellule de longue mémoire à court terme est un RNN particulier qui permet de pallier aux problèmes liés à la courte mémoire des RNN. Le LSTM introduit la notion de **"cellule de mémoire"**, son principe de fonctionnement est le suivant : le LSTM apprend ce qu'il faut stocker dans l'état à long terme, ce qui peut être oublié par le réseau et ce qu'il faut y lire [15]. il consiste en 3 operations (ou couches) appellées "portes" (Gate en anglais) :

— **Forget Gate :** cette opération permet d'oublier les informations inutiles (ou à en diminuer fortement le poids) qui étaient utiles au temps t-1 mais qui ne le sont plus à l'instant t [15], contrairement aux RNN classique, ils doivent mémoriser toutes les informations, meme celles inutiles (une valeur est considérée comme inutile si sa valeur est très proche de 0).

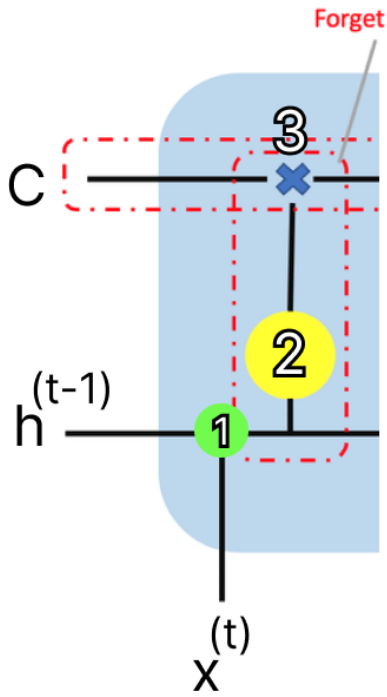

FIGURE 2.12 – Schéma explicatif de la porte Forget Gate

Premièrement, l'état caché précédent et les entrées actuelles sont **combinés** (étape 1). Cette combinaison passe dans la porte d'oubli (Forget Gate) qui, permet d'enlever les informations non-pertinentes. Cette couche génère un vecteur ou chaque élément de ce dernier est compris entre 0 et 1. De ce fait, les éléments ayant une valeur proche de 0 sont considérés comme non-pertinents. La cellule va alors déterminer qu'elles sont les informations à oublier sachant l'etat de la cellule précédente h et sachant les entrée x a l'instant t (étape 2) : les elements ayant une valeur proche de 0 seront considérés non-pertinents. Cet oubli sera appliqué sur la mémoire de la cellule à l'etape 3.

— **Input Gate :** Le but de cette étape est de déterminer quelles nouvelles informations doivent être ajoutées à la mémoire, compte tenu de l'état caché précédent et des nouvelles données d'entrée. Il convient de noter que les entrées ici sont les mêmes que les entrées de la porte d'oubli (Forget Gate) [15].

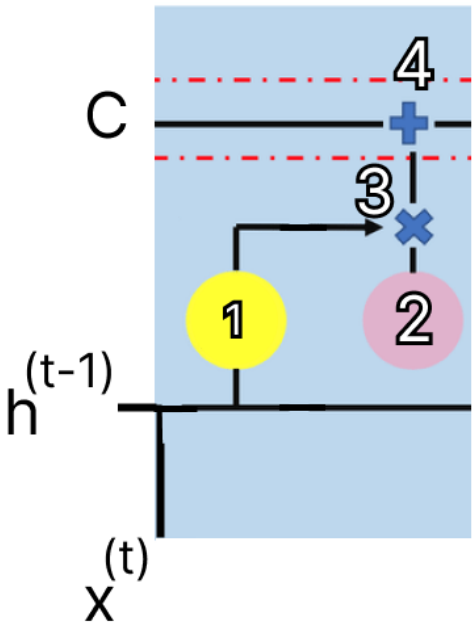

FIGURE 2.13 – Schéma explicatif de la du Input Gate

Premièrement, l'état caché précédent et les entrée actuelles sont combinés (étape 1). Les nouvelles informations utiles candidates qui seront ajoutées dans la mémoire seront séléctionnées dans l'etape 2. à l'étape 3, les informations candidates, les informations sur l'etat cachée précédent et les nouvelles entrées seront filtrés pour sélectionner que les informations qui seront ajouter (les elements ayant une valeur proche de 1). C'est a l'étape 4 que sera la **définition de la mémoire** à l'intant t (en prenant en compte les informations oubliées). De ce fait, la mémoire de la cellule va contenir que les informations qui sont utiles à garder depuis le passé et les nouvelles informations qui sont ajoutées a celle-ci ce qui permet de mettre a jour la mémoire de la cellule.

— **Output Gate :** pilote l'information qui sera transmise au temps t+1, cette opération permet de définir l'etat de la cellule à l'instant t en ayant comme sortie l'état caché h de cette cellule [15].

2.5.2.4 Les réseaux de neurones autoorganisés

les réseaux de neurones autoorganisés sont exploités pour le traitement d'informations spatiales. Ils sont utilisé pour produire une dimension réduite (souvent à deux dimensions) à partir d'une base de données ayant une haute dimension en préservant la topology de la structure des données.

2.6 Conclusion

Dans ce chapitre on a présenté les notions importantes qui sont en relation avec la vision par ordinateur (définitions, techniques ....etc). Ainsi qu'une vision générale sur l'apprentissage automatique et profond et l'impact des réseaux de neurones sur la vision par ordinateurs (

présentation des techniques courantes, architectures.. etc) et des explications plus detaillées sur les réseaux de neurones reccurents et convolutifs qui seront utilisées pour l'implémentation de l'application. Le prochain chapitre, traite les détails de la conception, ainsi que les méthodes et les outils utilisés pour la réalisation de l'application.

Chapitre 3

Conception et implémentation

3.1 Introduction

Dans ce chapitre, on va définir la conception de l'application en présentant les différents modèles utilisés avec leurs architectures appropriées, les outils utilisés ainsi que les technologies qui ont permis l'implémentation de cette application.

3.2 Logiciels et librairies Utilisés dans l'implémentation

3.2.1 Le serveur

- **Python :** Langage de programmation interprété, multiplateforme et très utilisé dans le domaine de l'intelligence artificielle et le traitement de données.
- **Flask :** Un micro-framework open-source pour le développement web.
- **Flask-SocketIO :** Un package python qui permet a Flask d'utiliser le protocol WebSocket.
- **TensorFlow :** Un framework open-source développé par Google pour faciliter le développement et l'entraînement de modèles d'apprentissage automatique.
- **Gunicorn :** Serveur web HTTP WSGI.
- **Eventlet :** Bibliothèque permettant a flask de tourner de façon asynchrone.

3.2.2 L'application Web :

- **HTML, CSS, TypeScript :** langages utilisés pour le développement d'applications web.
- **Vuejs :** un framework Javascript open-source utilisé pour la création d'interfaces utilisateur sur le web.
- **Mediapipe :** une bibliothèque multiplateforme développée par Google qui fournit d'étonnantes solutions ML prêtes à l'emploi pour les tâches de vision par ordinateur.
- **Socket.IO :** librarie qui permet l'utilisation du protocole WebSocket dans l'application web.

3.2.3 L'Extracteur de Données :

— **Requests :** bibliothèque Python utilisée pour envoyer des requêtes HTTP.

— **Numpy :** bibliothèque Python pour la manipulation des matrices et tableaux multidimensionnels.

— **Pandas :** Librairie Python pour l'analyse et la manipulation de données.

— **OpenCV :** bibliothèque open-source pour le traitement d'images en temps réel.

— **Mediapipe :** version Python de Mediapipe pour l'extraction des données d'entraînement du dataset.

3.2.4 Les Outils d'entraînement du modèle :

— **Pandas :** déjà décrite précédemment.

— **Scikit-learn :** bibliothèque open-source python pour l'apprentissage automatique.

— **TensorFlow :** déjà décrit précédemment.

— **Numpy :** déjà décrit précédemment.

— **Adam Optimizer** : Adam est un algorithme d'optimisation qui peut être utilisé à la place de la procédure classique de descente de gradient pour mettre à jour les poids de réseau itératifs en fonction des données d'apprentissage.

3.2.5 Configuration utilisée dans l'implémentation :

La configuration du matériel utilisé dans l'implémentation est :

— Un PC portable Lenovo X1 Carbon avec un CPU i7 8ème génération.

— RAM de taille 16GO DDR4 avec une vitesse de 2133 Mhz.

— Système d'exploitation Windows 11 64bits.

3.3 La base de données :

La base de données utilisée est celle de la langue des signes franaise belge LSFB realisé par l'université de Namur en belgique, c'est la seule base de données de la langue des signes francaise qui existe de nos jours. Elle contient plus de 50 heures de videos contenant des conversations en langue des signes. Cette base de données contient 2 sous bases de données : LSFB-CONT qui est une base de données pour de la reconnaissance continue, et **LSFB-ISOL** qui est une base de données pour de la reconnaissance de signes isolés. C'est la base LSFB-ISOL qui est utilisée pour entrainer le modèle. Elle est disponible sur le liens suivant `https://lsfb.info.unamur.be/`.

3.4 Déscription de l'approche

Notre approche se déroule en deux étapes principales :

— l'étape de la détéction : l'extraction des points clés qu'on appelle "Landmarks", qui serviront pour la prédiction.

— l'étape de la classification : l'étape de la prédiction du signe.

3.4.1 L'étape de la détéction :

L'étape de la détéction et de l'extractions des landmarks se fait grâce au modèle Mediapipe Holistics [29] proposé par la librairie Mediapipe décrite dans la section précédente. Tout d'abord, pour détecter les deux mains, une segmentation du corps humain par rapport à l'arrière plan, puis un squellette est généré grâce au modèle pré-trainé BlazePose : c'est un modèle qui permet d'extraire les landmarks du corps humain. Ce modèle donnera comme resultat un squellette couverant le corps avec 33 landmarks qu'on peut définir comme des "points clés" ou "repères". Dans notre système, seulement la partie du haut du corps compte.

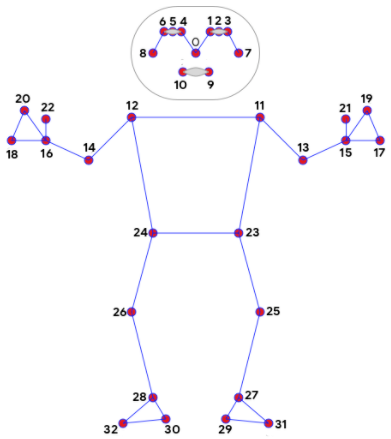

FIGURE 3.1 – les 33 landmarks résultantes par le modèle BlazePose [30]

Ensuite, l'image sera découpée en utilisant les points clés précédents pour extraire les 2 régions qui correspondent à l'endroit ou il ya des deux mains pour ensuite réaliser l'opération de détéction et de localisation des deux mains avec leurs landmarks respectives.

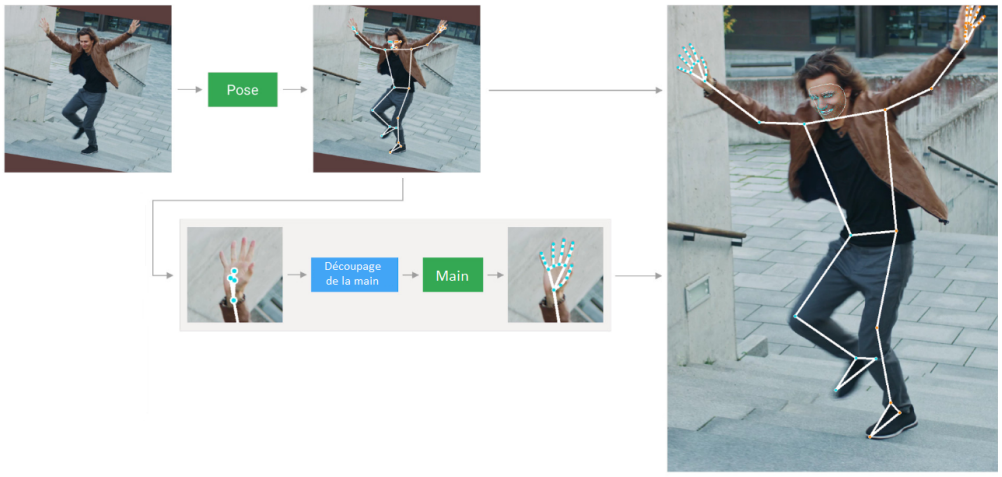

FIGURE 3.2 – Les étapes du processus de détéction des deux mains [29]

Pour détecter la main, l'approche utlisée est l'approche **One-Shot**. One-Shot Detector, appellé aussi **détecteur à un coup** est une méthode de détection qui effectue **la classification d'image** et **la détection de position (localisation)** en même temps. Dans de nombreux cas, il peut être classé en type YOLO (You Only Look Once) ou en type SSD (Single-Shot Detector). Parmi les détecteurs à un coup avec une excellente vitesse de détection, le détecteur **SSD (Single-Shot Detector)** est souvent plus rapide et c'est celui utilisé dans notre système.

3.4.1.1 Single-Shot Detector (SSD) :

Single Shot Detector (SSD) effectue **un passage unique** sur l'image pour détecter plusieurs objets. SSD est considéré comme l'un des types de modèle de détection des objets les plus rapides avec une exactitude relativement élevée. SSD comporte deux composantes : un modèle de **backbone** et **une tête de détection (Head)**. Le modèle de backbone est un réseau de classification d'images **pré-entraîné en tant qu'extracteur de features dont la couche de classification est enlevée**. Dans ce projet le classificateur utilisé est le modèle **VGG-16** entraîné sur Imagenet (ImageNet est une gigantesque base de données de plus de 14 millions d'images labellisées réparties dans plus de 1000 classes) dont la couche de classification entièrement connectée a été retirée.

— **Le backbone VGG-16 :**

VGG-16 est un réseau de neurones convolutif qui est consideré comme très bon classificateur d'images de nos jours.

VGG16 prend en entrée une image de taille 224x244, la couche Conv-1 a 64 filtres, Conv-2 a 128 filtres, Conv-3 a 256 filtres, Conv 4 et Conv 5 a 512 filtres suivi par 3 couches entiérement connectées. La figure 3.3 décrit l'architecture de ce modèle.

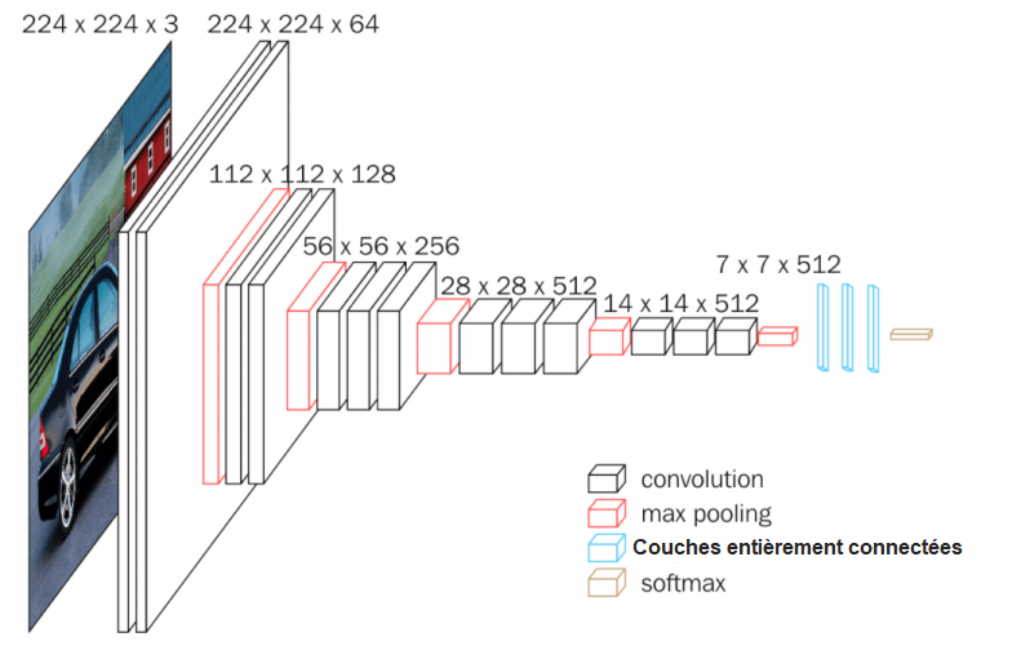

FIGURE 3.3 – Architecture du modèle VGG-16 [13]

L'image est passée à travers le premier stack (une pile de couches) de 2 couches de convolution, suivie d'une fonction d'activation ReLU. Chacune de ces deux couches contient 64 filtres. La résolution spatiale et la taille de la feature map de sortie est la même que les dimensions de l'image d'entrée. Les feature maps sont ensuite passées à travers une couche Max-pooling de taille 2x2. Cela réduit de moitié la taille des feature maps. Ainsi leurs à la fin du premier stack est de 112 x 112 x 64.

Les feature maps passent alors par un deuxième stack similaire, mais avec 128 filtres contre 64 dans la première. Par conséquent, la taille après le deuxième stack devient 56 x 56 x 128. Elle est suivie par le troisième stack avec trois couches convolutives et une couche de max pooling. Le nombre de filtres appliqués ici sont 256, ce qui donne la taille de sortie de la pile 28 x 28 x 256. Ceci est suivi par deux stacks de trois couches convolutionnelles, chacune contenant 512 filtres. La taille des feature maps à la sortie à la fin de ces deux stacks sera de 7 x 7 x 512.

Les stacks de couches convolutives sont suivies de trois couches entièrement connectées avec une couche d'aplatissement entre les deux. La couche de sortie est suivie par la couche Softmax utilisée pour la classification catégorielle. Sauf que cette partie de classification est retiré dans le modèle SSD.

— **La tete de détection (Head) :**

Le Head ou la tête de détection est seulement composée d'une ou de plusieurs couches convolutives ajoutées à ce backbone (dont la couche de classification est enelevée). Cela nous donne la sortie sous forme de boîtes englobantes sur les objets appellées "Bounding Boxes" qui definissent la localisation des mains. Dans ce modèle 6 couches additionnelles

sont employées.

L'archirecture du modèle SSD se présente ainsi :

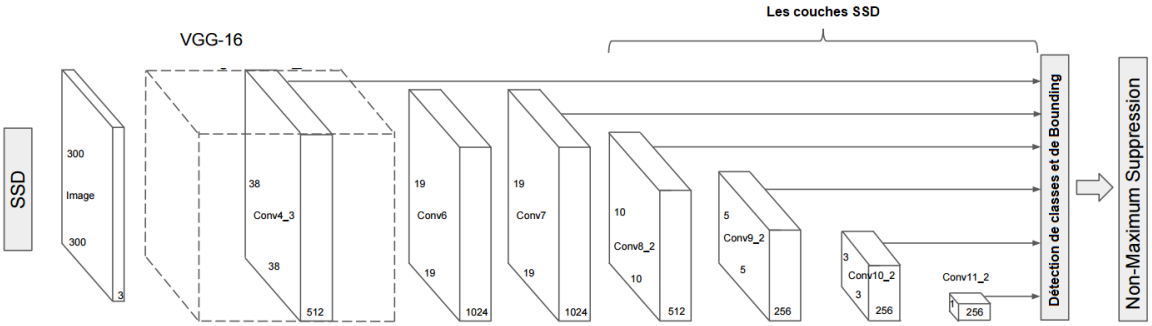

FIGURE 3.4 – Architecture du modèle SSD [3]

Après le passage de l'image sur le réseau VGG, 6 couches de convolution sont ajoutées pour produire des feature maps de tailles 19x19, 10x10, 5x5, 3x3, 1x1. Celles-ci, ainsi que la feature map de taille 38x38 produite par la couche conv4-3 de VGG, sont les feature maps qui seront utilisées pour prédire les bounding boxes (les boîtes englobantes), ce processus permet la détéction en multi-scale (multi-echelle) ce qui rend la détection plus précise.

SSD divise l'image à l'aide d'une grille, chaque cellule de la grille étant chargée de détecter des objets dans cette région de l'image. La détection des objets implique de prédire la classe et la localisation d'un objet dans cette région. En l'absence d'objets, on considère qu'il s'agit d'une classe d'arrière-plan et la localisation est ignorée. On aura alors comme sortie un score de confiance (la probablité que l'objet de la classe C se trouve dans la cellule) et les coordonnées de la bounding box. On obtiendra alors à la fin des convolutions, une multitudes de boites avec chacune sa localisation et la probabilité que la boite contienne la classe voulue. C'est la que l'operation de Non-Max-Suppression intervient pour en séléctionner qu'une seule boite. Cette opération consiste à comparer tous les scores de confiance des boites de chaque classe, la boite ayant le score de confiance le plus élevé sera séléctionnée. On obtiendra alors la localisation des deux mains avec leurs coordonnées respectives, ces coordonnées sont ensuite transferées a un modèle pré-trainé spécialisé dans la détection des landmarks pour avoir en sortie 21 landmarks pour chaque main comme l'illustre la figure au-dessous :

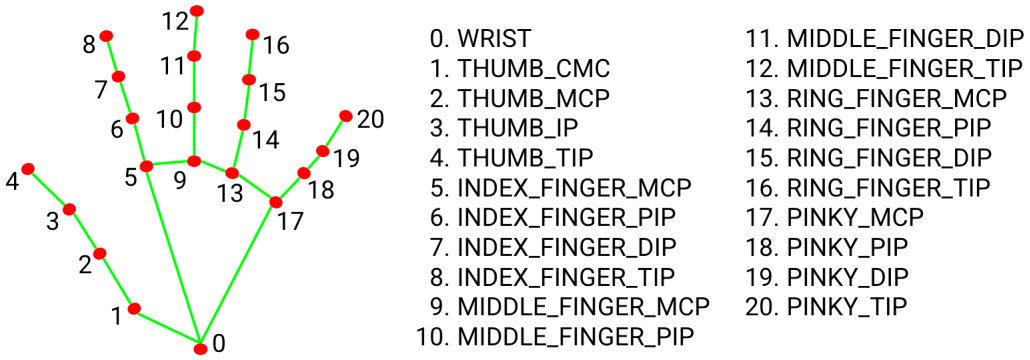

FIGURE 3.5 – Les 21 landmarks d'une main [28]

3.4.2 L'étape de la classification :

Les landmarks résultantes de l'étape de la détection seront stockées dans un tableau qui sera ensuite transferé au modèle qui correspond à un empilement de réseaux de neurones LSTM qui se chargera de classifier et de détecter les signes. Comme décrit précédement dans le chapitre 2, l'architecture LSTM contient des "blocs de mémoire". Ces blocs de mémoire contiennent des cellules de mémoire stockant les états du réseau au fils du temps (dans notre cas il stock les coordonnées des landmarks au fils du temps), ce qui rend ce modèle parfait pour la detection/reconnaissance d'action en temps réel contrairement aux autres modèles de reseaux de neurones existants. Ce programme permettra de derouler la reconnnaissance de la langue des signes francaise en temps réel fonctionnant sur le web.

Le modèle crée contient six couches incluant la couche d'entrée et la couche de sortie. Ce modèle est un empilement de 3 couches LSTMs avec 3 couches denses entiérement connectées à la fin de la classification, la classification se fait à travers le mouvement des landmarks en collectant a chaque fois 30 frames (par convention, on considère qu'un signe prend 1 seconde). Les couches ont un nombre différent d'unités :

— Couche entrée : 64 unités.

— 1ere couche cachée : 128 unités.

— 2eme couche cachée : 64 unités.

— 3ème couche cachée : 64 unités.

— 4ème couche cachée : 32 unités.

— Couche de sortie : le nombre de mots que le modèle peut prédire.

On a choisi l'optimizer Adam pour le modèle avec un taux d'apprentissage de 0.00001 : Adam est un algorithme d'optimisation qui peut être utilisé à la place de la procédure classique de descente de gradient pour mettre à jour les poids de réseau itératifs en fonction des données d'apprentissage. La valeur choisie pour le taux d'apprentissage est si bas car cela permet un apprentissage plus lent mais garantissant de meilleurs résultats.

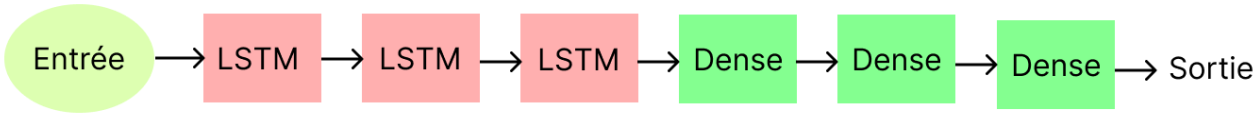

FIGURE 3.6 – Architecture de LSTM

On aura alors comme sortie une liste de probabilités qui correpondent chacune à un signe (la somme de ces probabilités est égale à 1) , la probablité la plus élevée est considérée comme le resulat finale de la prédiction c'est a dire le signe prédit .

3.5 Description du déroulement de l'application :

Notre application se présente sous forme d'une application web qui communique avec un serveur qui contient le modèle qui gère les prédictions.

Tout d'abord, L'application web commence par capturer le flux vidéo de la webcam, ensuite elle passe chaque frame au modèle Mediapipe Holistics qui retourne une liste de coordonnées qui indiquent les différentes landmarks du corps et des mains. De ce fait, Mediapipe retourne trois listes : une qui contient les landmarks du corps et une pour la main gauche et une pour la main droite.

On prend toutes ces landmarks de ces listes en les combinant dans une seule et unique liste qui représente donc la frame au total avec toutes les landmarks présentes. Les données de cette frame sont ensuite sauvegardées dans une liste de frames.

Quand cette liste de frames atteint 30 entrées, elle est envoyée au serveur et on réinitialise la liste (on considère que chaque signe correspond à 30 frames, donc a chaque collecte de 30 frames, une prédiction est faite et la liste est réinitialisée pour collecter les prochaines 30 frames). Le serveur reçoit les données, les passe à notre modèle de prédiction pour qu'il fasse la classification du signe.

Ensuite, on prend les prédictions de notre modèle et on construit une réponse qu'on envoie à l'application web qui ensuite l'affiche à utilisateur.

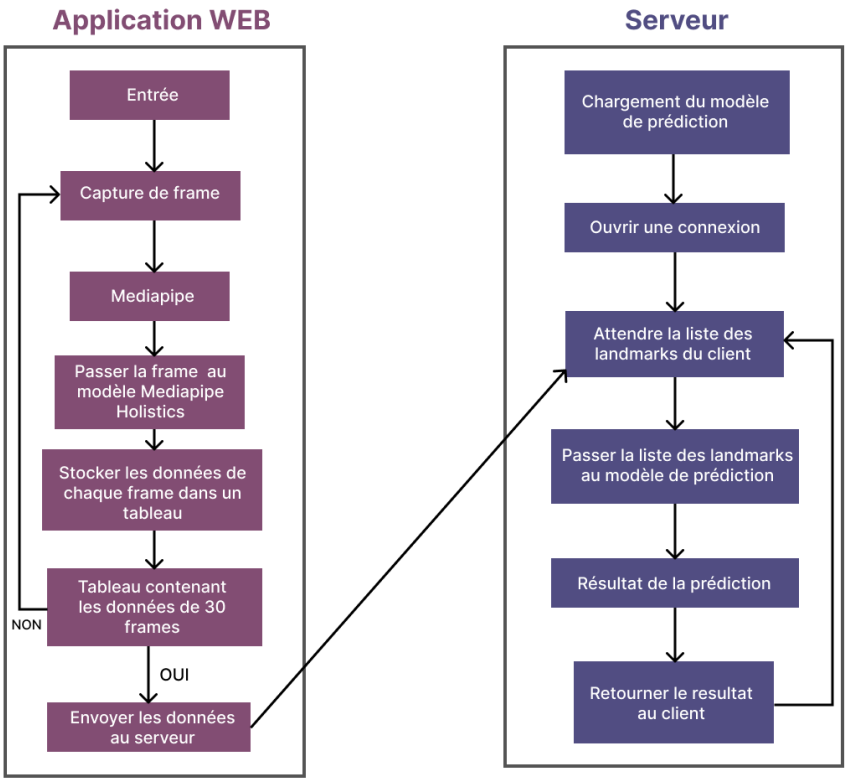

FIGURE 3.7 – Le fonctionnement de l'application Web

3.5.1 Les interfaces

La fenetre principale de l'application est :

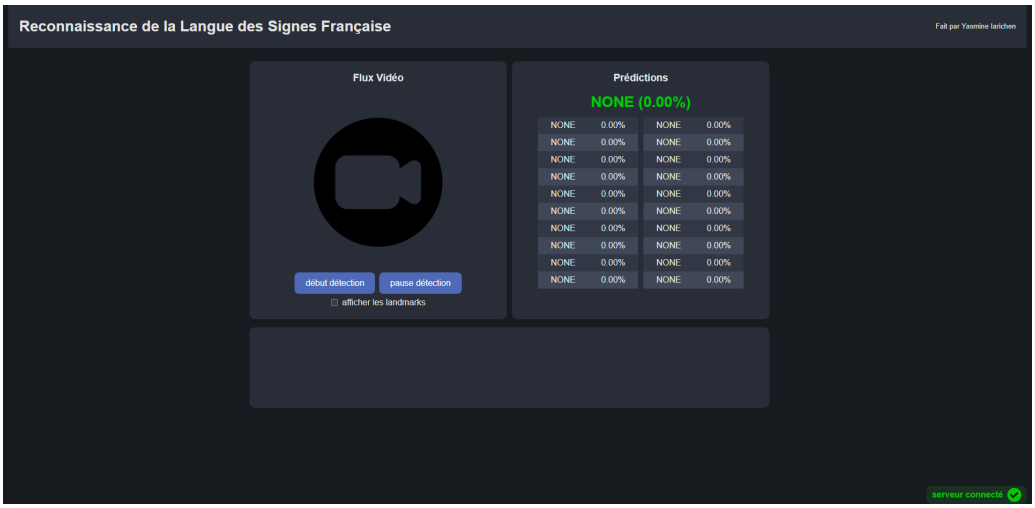

FIGURE 3.8 – Interface de l'application

Cliquer sur le bouton "début détection" pour allumer la fenetre "camera" et debuter la détection.

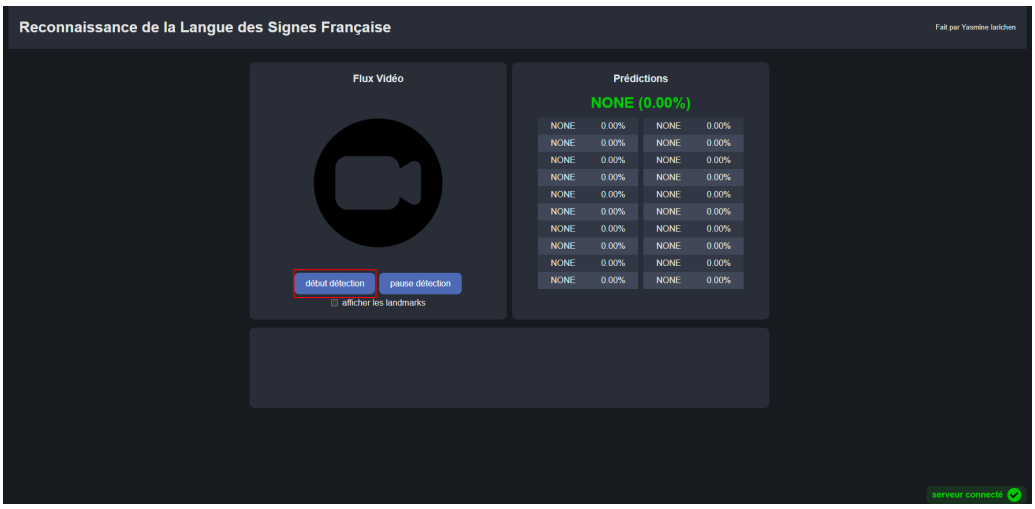

FIGURE 3.9 – Démarrer la détection

Cliquer sur le bouton "Pause détection" pour arreter la détection.

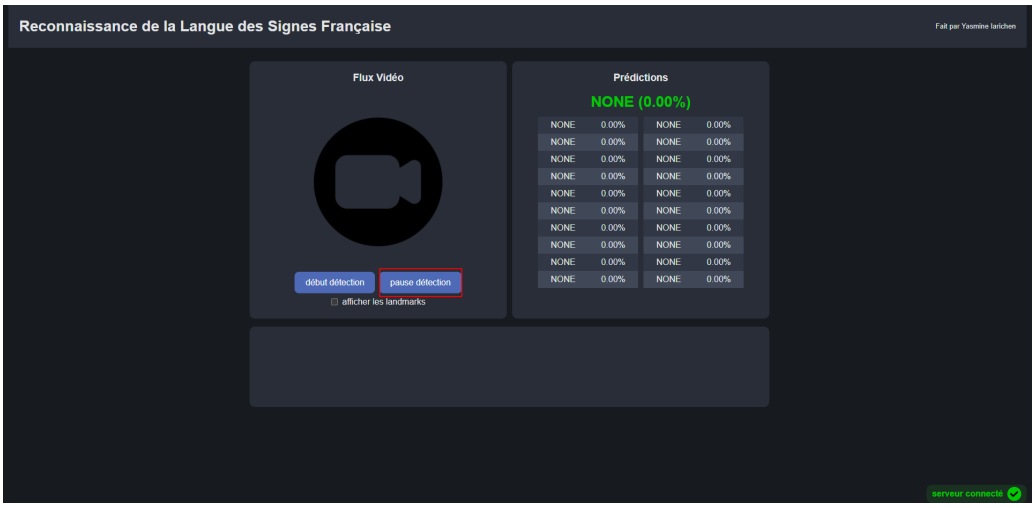

FIGURE 3.10 – Arrêter la détection

Le modèle retourne une liste de probabilités, chaque entrée dans la liste correspond à un mot. On prend la plus grande probabilité comme étant le résultat final de la prédiction.

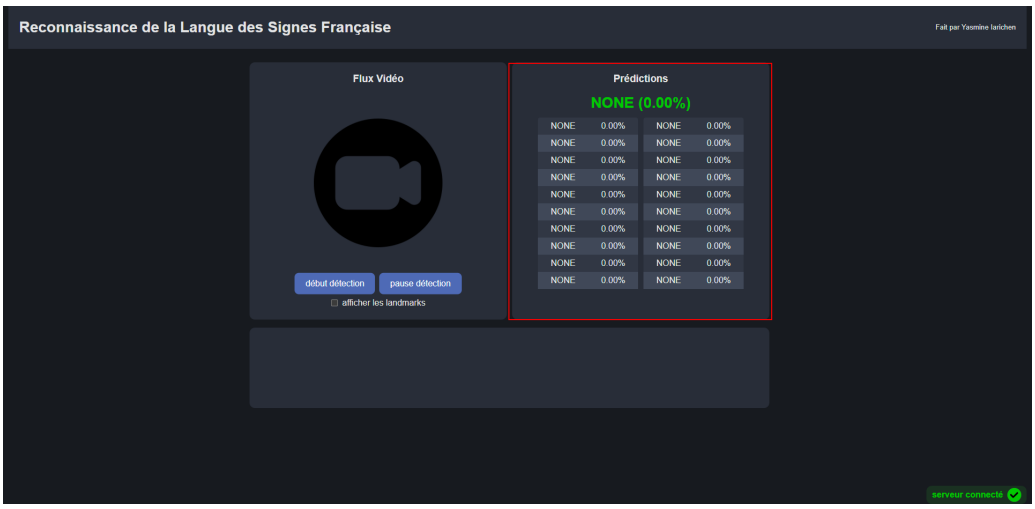

FIGURE 3.11 – La prédiction des signes

Le bloc au dessus montre les prédiction des signes.

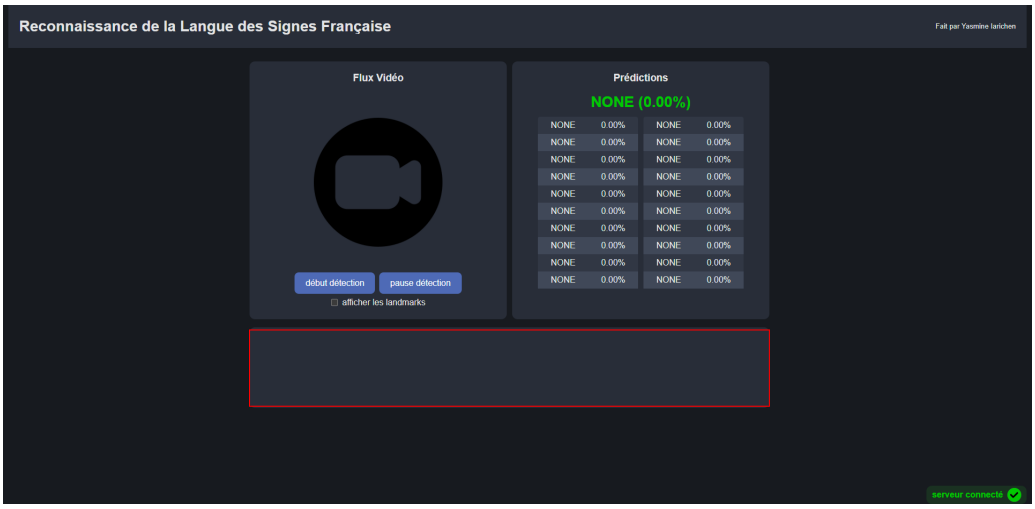

FIGURE 3.12 – L'affichage des mots correspondants aux signes

3.6 Les résultats obtenus

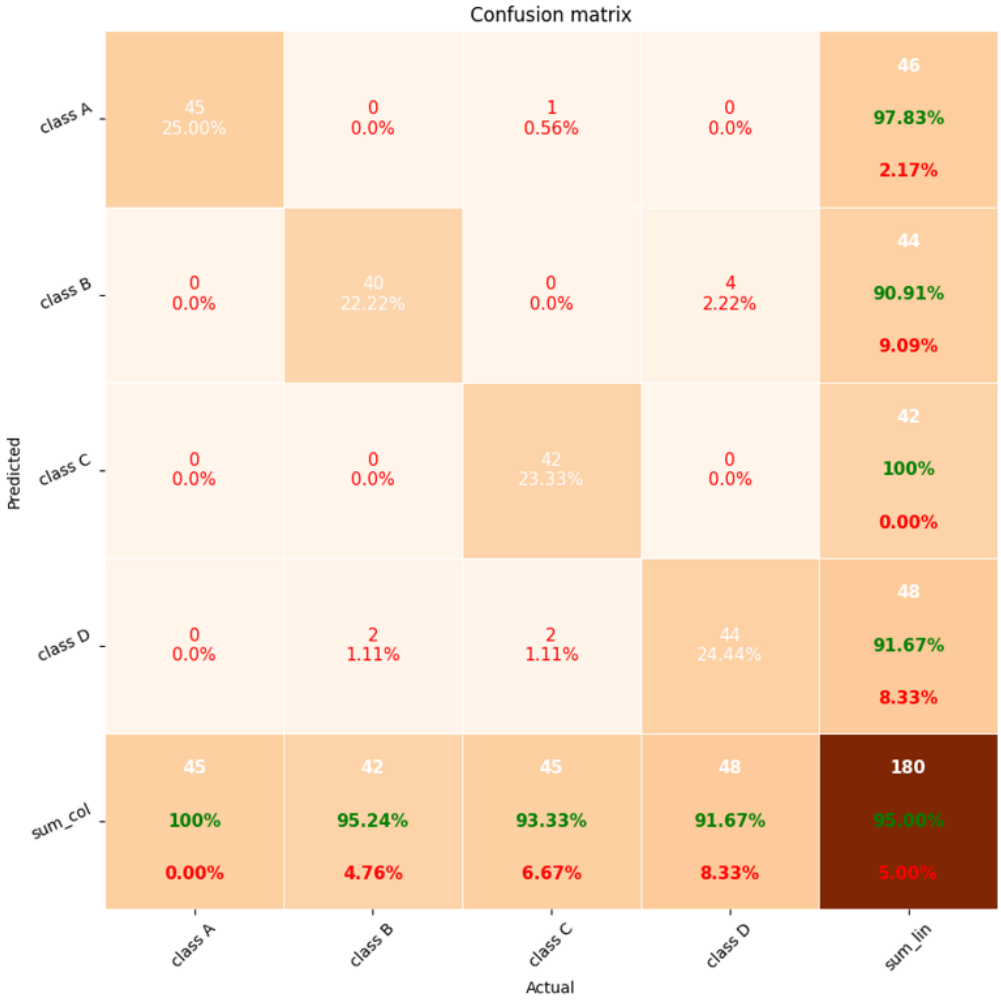

FIGURE 3.13 – Le taux de précision

On a reussi à attendre un taux de précision de 95% en utilisant l'optimizer Adam configuré avec un learning rate de 0,00001.

Pour éviter le problème de la classification déséquilibrée, nous avons séléctionner que les signes ayant un grands nombres de videos de qualité dans la base de données réparties en 4 classes : Classe A : AUSSI, Classe B : COMPRENDRE, Classe C : DANS, Classe D : OUI. .

3.7 Conclusion

Ce chapitre a abordé les outils nécessaires pour la réalisation de l'application. On a présenté aussi l'environnement de développement ainsi que le déroulement et le fonctionnement du modèle suivi de quelques captures d'ecran expliquant l'interface de l'application web ainsi que les résultats obtenus avec des détails sur les modèles utilisés et leur impact sur la reconaissance en temps réel.

Conclusion Générale

Dans ce projet nous avons abordé les notions fondamentales de la langue des signes, de la vision par ordinateur ainsi que les réseaux de neurones en générale et les réseaux de neurones convolutionnels et réccurents en particulier en expliquant leurs fonctionnements et leurs architectures.

La combinaison de ces deux architecture a permit d'obtenir des meilleurs résultats en terme de précision. De plus cette application est multi-platformes et ne demande pas un appareil performant pour qu'elle marche.

Quelques problèmes ont été rencontré durant l'implémentation, à savoir la base de données est assez pauvre en terme de variété des signes et de qualité des vidéos, de plus, l'écart entre le nombre de videos de chaque signes est relativment grand ce qui rend le modèle instable et créé le problème de données déséquilibrées. De plus, les problèmes de connexion empêche l'application de se dérouler fluidement, l'utilisation d'un CPU a aussi fait que le temps d'exécution était très couteux.

De ce fait la solution était de séléctionner que les signes ayant un grand nombre de videos (plus de 1000) pour améliorer la précision en sacrifiant la variété de signes que le modèle pourrait détécter. L'utilisation d'un GPU puissant (une carte graphique) accompagné avec un système puissant (CPU puissant, quantité elevée de RAM, grand stockage) est nécessaire pour un gain de temps et de performance.

Comme perspectives, on aimerai élargir la variété de signes en diminuant le temps de détéction, l'idéale est que l'application soit mobile pour qu'elle soit utilisable partout et à tout moment.

Bibliographie

[1] Y. A. Cornuéjols, L. Miclet. *Apprentissage Artificiel, Concepts et algorithmes*. Eyrolles, 2009.

[2] A. Manzanera. Les images numériques [en ligne] consulté le [mois de juin, 2022]. Disponible sur : `https://perso.ensta-paris.fr/~manzaner/Cours/Poly/Poly_Chap1_Intro.pdf`.

[3] A. Rohan. Convolutional Neural Network-Based Real-Time Object Detection and Tracking for Parrot AR Drone 2. *IEEE Access*, 2019.

[4] Adobe. Fichiers images [en ligne] consulté le [mois de juin, 2022]. Disponible sur : `https://www.adobe.com/ca_fr/creativecloud/file-types/image.html`.

[5] Arils. Que doit-on dire : Langue des signes ou langage des signes ? [en ligne] consulté le [mois de juillet, 2022]. Disponible sur : `http://arils.ch/langue-des-signes-ou-langage-des-signes/`.

[6] Belmerabet Sarra. Segmentation d'image. Mémoire de Master, Université Larbi Ben M'hidi Oum El Bouaghi, 2007.

[7] Bossard Bruno. Problèmes posés par la reconnaissance de gestes en langue des signes. Mémoire de Master, Université Paris XI, 2004.

[8] Boughaba Mohammed et Boukhris Brahim. L'apprentissage profond (deep learning) pour la classification et la recherche d'images par le contenu. Mémoire de Master, Université Kasdi Merbah Ouargla, 2017.

[9] Cypris. Langue des signes francaise [en ligne] consulté le [mois de juin, 2022]. Disponible sur : `https://cypris.fr/solidarite/LSF.pdf`.

[10] Datascientest. Convolutional neural network [en ligne] consulté le [mois de juillet, 2022]. Disponible sur : `https://datascientest.com/convolutional-neural-network`.

[11] Dominique Béréziat. Bases du traitement des images [en ligne] consulté le [mois de août, 2022]. Disponible sur : `https://www-master.ufr-info-p6.jussieu.fr/parcours/ima/bima/`.

[12] Fouad Boudjenouia. Restauration d'images avec critères orientés qualité. Thèse de Doctorat, Université d'Orléans, 2018.

[13] i2tutorials. What do you mean by vgg16 model ? and how do we use it for image classification ? [mois de septembre, 2022]. Disponible sur : `https://www.i2tutorials.com/what-do-you-mean-by-vgg16-model-and-how-do-we-use-it-for-image-classification/`.

[14] Ilias Papastratis, Christos Chatzikonstantinou , Dimitrios Konstantinidis. Artificial Intelligence Technologies for Sign Language. *Sensors*, 21, 2021.

[15] J. Schmidhuber. Long Short-term Memory. *Neural Computation*, 9(8) :1735–1780, 1997.

[16] James F. Peters. *Foundations of Computer Vision*. Springer Publishing, 2017.

[17] Jean-Marie John-Mathews. L'interprétabilité en apprentissage machine, un regard sur les réseaux de neurones artificiels profonds. Mémoire de Master, Université Paris 1 Paanthéon Sorbonne, 2018.

[18] Jedha. Qu'est-ce qu'un réseau de neurones en deep learningk [en ligne] consulté le [mois de juillet, 2022]. Disponible sur : `https://www.jedha.co`.

[19] Joseph Murray. World federation of the deaf [en ligne] consulté le [mois de mai, 2022]. Disponible sur : `https://wfdeaf.org/`.

[20] Laboiteasaussure. Histoire de la langue des signes française [en ligne] consulté le [mois de juin, 2022]. Disponible sur : `http://laboiteasaussure.fr/lsf_histoire.htm`.

[21] H. Laborit. Quelques signes pour communiquer avec les patients sourds [en ligne] consulté le [mois de juillet, 2022]. Disponible sur : `https://ch-laborit.fr/wp-content/uploads/1/2017/09/Quelques-signes-pour-communiquer-avec-les-patients-Sourds.pdf`.

[22] T. Laouadi Narimane. Segmentation d'image par région sur la base des contours des objets. Mémoire de Master, Université Larbi Ben M'hidi Oum El Bouaghi, 2017.

[23] Le Big Data. Understanding of convolutional neural network (cnn) — deep learningréseau de neurones artificiels : qu'est-ce que c'est et à quoi ça sert ? [en ligne] consulté le [mois de juillet, 2022]. Disponible sur : `https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition`.

[24] Le big data. Vision par ordinateur [en ligne] consulté le [mois de juin, 2022]. Disponible sur : `https://www.lebigdata.fr/computer-vision-definition`.

[25] Le DAP. Réseaux de neurones reccurents [en ligne] consulté le [mois de juillet, 2022]. Disponible sur : `https://dataanalyticspost.com/Lexique/reseaux-de-neurones-recurrents/`.

[26] Lorena P Vargas. Sign Language Recognition System using Neural Network for Digital Hardware Implementation . *Journal of Physics : Conference Series*, 2011.

[27] Mansour Mohamed Seghier. Langage et surdité. Mémoire de Master, Université d'Oran, 2007.

[28] Mediapipe. Mediapipe hands [en ligne] consulté le [mois de août, 2022]. Disponible sur : `https://google.github.io/mediapipe/solutions/hands.html`.

[29] Mediapipe. Mediapipe holistics [en ligne] consulté le [mois de septembre, 2022]. Disponible sur : `https://google.github.io/mediapipe/solutions/holistic.html`.

[30] Mediapipe. Mediapipe pose [en ligne] consulté le [mois de août, 2022]. Disponible sur : `https://google.github.io/mediapipe/solutions/pose.html`.

[31] Mohammed Khamadja et Said Benierbah. Traitement d'images [en ligne] consulté le [mois de juin, 2022]. Disponible sur : `https://fac.umc.edu.dz/fstech/cours/Electronique/`.

[32] Mokri Mohammed Zakaria. Classification des images avec les réseaux de neurones convolutionnels. Mémoire de Master, Université Abou Bakr Belkaid Tlemcen, 2017.

[33] Prabhu. Understanding of convolutional neural network (cnn) — deep learning [en ligne] consulté le [mois de juillet, 2022]. Disponible sur : https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[34] Ranjay Krishna. *Computer Vision : Foundations and Applications*. Stanford University, 2017.

[35] Razieh Rastgoo, Kourosh Kiani , Sergio Escalera . Sign Language Recognition : A Deep Survey. *Expert Systems with Applications*, 164, 2020.

[36] Schwager W et Zeshan U. Word classes in sign languages . [parts of speech : Descriptive tools, theoretical constructs]. 32(3) :509–545, 2008.

[37] Simon Bernard. Extraction de primitives structurelles pour la reconnaissance de symboles : Une approche robuste par transformée de hough. Mémoire de Master, Université de la Rochelle, 2007.

[38] The Echo Project. Sign language [en ligne] consulté le [mois de juin, 2022]. Disponible sur : http://sign-lang.ruhosting.nl/echo/docs/SignLanguages.pdf.

[39] Thomas Oberlin. Traitement d'images : Partie 4 [en ligne] consulté le [mois de juin, 2022]. Disponible sur : https://perso.ensta-paris.fr/~manzaner/Cours/Poly/Poly_Chap1_Intro.pdf.

[40] Tom Keldenich. Cnn et couche de convolution, qu'est-ce que c'est ? [en ligne] consulté le [mois de juillet, 2022]. Disponible sur : https://inside-machinelearning.com/cnn-couche-de-convolution/.

[41] Vipin Tyagi. *Understanding Digital Image Processing*. Taylor Francis, 2018.

[42] Wasseem Nahy Ibrahem. Image processing lecture 2 [en ligne] consulté le [mois de juin, 2022]. Disponible sur : https://www.academia.edu/37315655/Types_of_Digital_Images.

[43] Wikipedia. Langue des signes française [en ligne] consulté le [mois de juin, 2022]. Disponible sur : https://fr.wikipedia.org/wiki/Langue_des_signes.

[44] Wikipedia. Platon [en ligne] consulté le [mois de septembre, 2022]. Disponible sur : https://fr.wikipedia.org/wiki/Platon.

RÉSUMÉ

Ce mémoire aborde la conception et l'implementation d'un système qui fait la reconaissance de la langue des signes francaise en temps réel, son but est d'aider la communauté des malentendants de fondre dans la société. Pour cela, nous avons utilisé les réseaux de neurones réccurents et convolutifs implémentés dans des modèles recents à l'aide de differentes bibliothèques spécialisés dans le Machine Learning et le Deep Learning (Mediapipe, TensorFlow ..etc). Les résultats obtenus montre l'efficacité de la méthode qu'on a utilisée.

ABSTRACT

This thesis addresses the design and implementation of a system for real time recognition of the French sign, its goal is to help the deaf community to blend into society. For this, we used recurrent and convolutional neural networks implemented in recent models using different libraries specialized in Machine Learning and Deep Learning (Mediapipe, TensorFlow ..etc). The results obtained show the effectiveness of the method we used.