

Rapport du Projet de fin d'année

La Reconnaissance Du

Langage Signe



Réalisé par :

El-Bouchikhi Kawtar
Mikou Badr

Encadré par :

Pr. Hraoui Said

ENSAF-GSEII2 – 2020/2021

Remerciements

On profite par le biais de ce rapport à remercier toutes les personnes ayant contribué au succès de ce projet.

On remercie particulièrement **Mr. Said Hraoui**, notre encadreur académique à l'école nationale des sciences appliquées de Fès, pour leur patience, leur disponibilité et surtout leurs judicieux conseils, qui ont contribué à alimenter notre réflexion et pour leurs soutiens, leurs aides, qu'ils trouvent ici l'expression de nos profondes reconnaissances et nos profonds respects.

Aussi on remercie nos professeurs de la filière génie systèmes embarqués et informatique industrielle pour leurs temps, leur partage de connaissances et leurs efforts continus pour rendre notre formation meilleure.

Résumé

La perte de la capacité de parler ou d'entendre exerce des impacts psychologiques et sociaux sur les personnes affectées en raison du manque de communication appropriée. Les systèmes de reconnaissance de la langue des signes basés sur des gants sensoriels sont des innovations importantes qui visent à obtenir des données sur la forme ou le mouvement de la main humaine. La technologie innovante en la matière est principalement restreinte et dispersée. Les tendances et les lacunes disponibles devraient être explorées dans cette approche de recherche pour fournir des informations précieuses sur les environnements technologiques. Ainsi, une revue est menée pour créer une taxonomie cohérente pour décrire les dernières recherches divisées en quatre catégories principales : développement, cadre, reconnaissance des autres gestes de la main, et revues et sondages. Ensuite, nous concevons un système pratique qui soit utile aux personnes qui ont des difficultés auditives et en général qui utilisent une méthode très simple et efficace ; Un système de capture de mouvement pour la conversion de la langue des signes. Il capte les signes et dicte à l'écran comme une écriture.

Mots clés

Langue des signes, reconnaissance de la langue des signes, traduction, Traducteur signe en lettre, Gant à main électronique, Capteurs Flex, Microcontrôleur, LCD, Intelligence artificielle, Réseaux neuronaux, CNN, classification, Reconnaissance de la posture de la main, OpenCV.

Abstract

The loss of the ability to speak or hear has psychological and social impacts on those affected due to lack of appropriate communication. Sign language recognition systems based on sensory gloves are important innovations that aim to obtain data on the shape or movement of the human hand. The innovative technology in this area is mainly restricted and dispersed. The available trends and gaps should be explored in this research approach to provide valuable insight into technological environments. Thus, a review is being conducted to create a consistent taxonomy to describe the latest research divided into four main categories: development, framework, recognition of other hand gestures, and reviews and surveys. Then we design a practical system that is useful for people who have hearing difficulties and in general who use a very simple and effective method; A motion capture system for converting sign language. He picks up the signs and dictates on the screen like writing.

Keywords

sign language, sign language recognition, Sign to Letter Translator, Electronic Hand Glove, Flex sensors, Microcontroller, LCD, Artificial Intelligence, Neural networks, CNN, classification, hand posture recognition, OpenCV.



Table des matières

Remerciements.....	2
Résumé.....	3
Abstract	4
Table des matières.....	5
Liste de figures.....	7
Liste de tableaux	8
Liste des Abréviations	8
Introduction générale.....	9
Chapitre 1 : Contexte général du projet.....	11
1. Motivation.....	12
2. Méthodologie :	12
2.1 Reconnaissance de langage des signes (SLR) :	12
2.2. Matériels et méthodes	16
Chapitre 2 : un système de traduction signe-lettre à l'aide d'un gant à main	22
1. Introduction.....	23
2. Objectifs.....	23
3. Méthodologie	23
3.1 Architecture globale.....	24
3.2 Mise en Œuvre Matérielle du Système Proposé	25
4. Conclusion et Plans Futurs	31
Chapitre 3 : un système de traduction signe-lettre à l'aide des réseaux de neurones.....	32
1. Introduction	33
2. Enquête bibliographique	33
a. L'acquisition des données :	33
b. Prétraitement des données et extraction de caractéristiques:	34
c. Classement des gestes :	35
3. Mots clés et définitions.....	36
3.1 Extraction et représentation de caractéristiques :	36
3.2 Réseaux de neurones artificiels :	36



3.3 Réseau de neurones à convolution :	37
3.4 TensorFlow	40
3.5 Keras	40
3.6 OpenCV	40
4. Méthodologie	41
4.1 Génération d'ensembles de données.....	41
4.2 CLASSEMENT DES GESTES	42
5. Défis rencontrés	47
6. Résultats.....	47
7. Conclusion :	50
8. Portée future :	50
Conclusion.....	51
Les références	53
APPENDICE	54

Liste de figures

Figure 1. Scénario actuel	9
Figure 2. Langage signe américain	10
Figure 3. Les éléments essentiels liés à la formation des gestes en langue des signes	13
Figure 4. Approches de reconnaissance de la langue des signes	13
Figure 5. Un organigramme des étapes utilisées dans le système basé sur la vision pour SLR	14
Figure 6. Les phases de collecte et de reconnaissance des données de gestes SL à l'aide du système à base de gants.	15
Figure 7. Les principaux composants matériels du système à base de gants.	16
Figure 8. (a) Capteur de flexion, (b) niveaux de flexion et (c) circuit diviseur de tension [2]	17
Figure 9. L'ACC à 3 axes ADXL335 avec une broche analogique à trois sorties x, y et z.	18
Figure 10. (a) microcontrôleur ATmega, (b) microcontrôleur MSP430G2553, (c) carte Arduino Uno et (d) mini-ordinateur Odroid XU4.	19
Figure 11. Nombre d'articles sur chaque variété de gestes.	21
Figure 12. Schéma fonctionnel proposé	24
Figure 13. Schéma fonctionnel du S2L proposé	25
Figure 14. Capteurs flexibles	26
Figure 15. L'ensemble du système	28
Figure 16. Organigramme du programme S2L	29
Figure 17. Représentation en bits de la lettre A	29
Figure 18. Signe de la lettre A	30
Figure 19. Précision du système S2L	31
Figure 20. Réseaux de neurones artificiels	37
Figure 21. Réseaux de neurones à convolution	38
Figure 22. Types de mise en commun	39
Figure 23. Couche entièrement connectée	39
Figure 24. Image affichée par la webcam et la région d'intérêt	41
Figure 25. Image convertie en niveau de gris	42
Figure 26. Image après l'implication du flou gaussien	42

Figure 27. Matrice de confusion (Algo 1)	49
Figure 28. Matrice de confusion (Algo 1 + Algo 2)	49

Liste de tableaux

Tableau 1. Angles des plages de flexibilité pour différentes positions	26
Tableau 2. Binaire de chaque lettre pour chaque capteur flex	27
Tableau 3. Exemple de valeurs de la lettre "A"	30

Liste des Abréviations

ANN : Artificial Neural Networks

ASL : American sign language

CNN : Convolutional Neural Network

GPU : Graphical Processing Unit

LCD : Liquid Crystal Display

ReLU : Rectified Linear Unit

ROI : region of interest

RVB : Rouge vert bleu

SLR : Sign Language Recognition

Introduction générale

Aujourd'hui, il y a près de 2 millions de personnes classées comme sourds-muets. Ils ont de grandes difficultés à communiquer entre eux et avec d'autres individus car le seul moyen de communication est la langue des signes. Ils doivent apprendre cette langue des signes. Il est extrêmement difficile pour une personne qui ne connaît pas cette langue des signes de comprendre et de décoder ses actions.

Il est impossible d'identifier quoi que ce soit sans sa connaissance préalable. Même pour les ordinateurs, ils ont besoin d'avoir des informations dans leur mémoire pour identifier et fournir des données relatives à n'importe quel objet. Maintenant, un objet particulier peut différer d'un autre type d'objet similaire par sa forme, sa taille, son orientation ou même les effets visuels peuvent différer. Mais toutes les différentes formes d'un type d'objet doivent être classées dans la même catégorie.

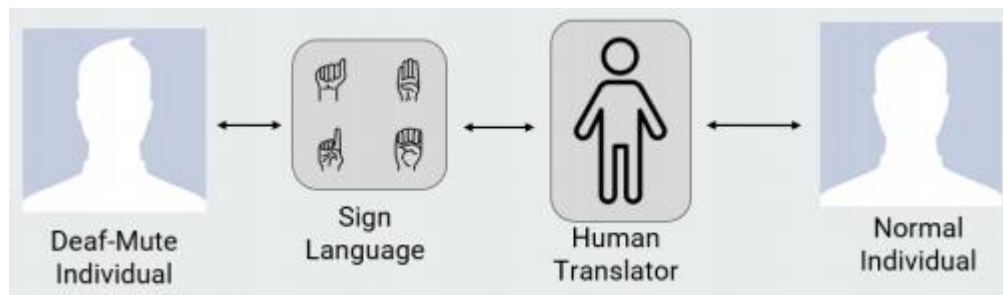


Figure 1. Scénario actuel

Il existe deux approches pour la reconnaissance des gestes. Dans l'approche non basée sur la vision, des capteurs tels que des capteurs flexibles, des capteurs de pression sont utilisés pour la reconnaissance des signes qui ne peuvent nécessiter aucun éclairage approprié. Dans une approche basée sur la vision, une image gestuelle en temps réel réalisée par des personnes sourdes-muettes

est utilisée comme entrée pour la reconnaissance des signes, ce qui nécessite un éclairage approprié pour des résultats précis.

Dans notre projet, nous nous concentrons sur la création un traducteur en langue des signes qui utilise un gant équipé de capteurs capables d'interpréter les 26 lettres, mots et phrases anglais en langue des signes américaine (ASL). Le gant portable utilise des capteurs et des accéléromètres pour collecter des données sur la position de chaque doigt et le mouvement de la main. L'alphabet signé est ensuite affiché sur l'écran LCD ainsi que prononcé par les haut-parleurs.

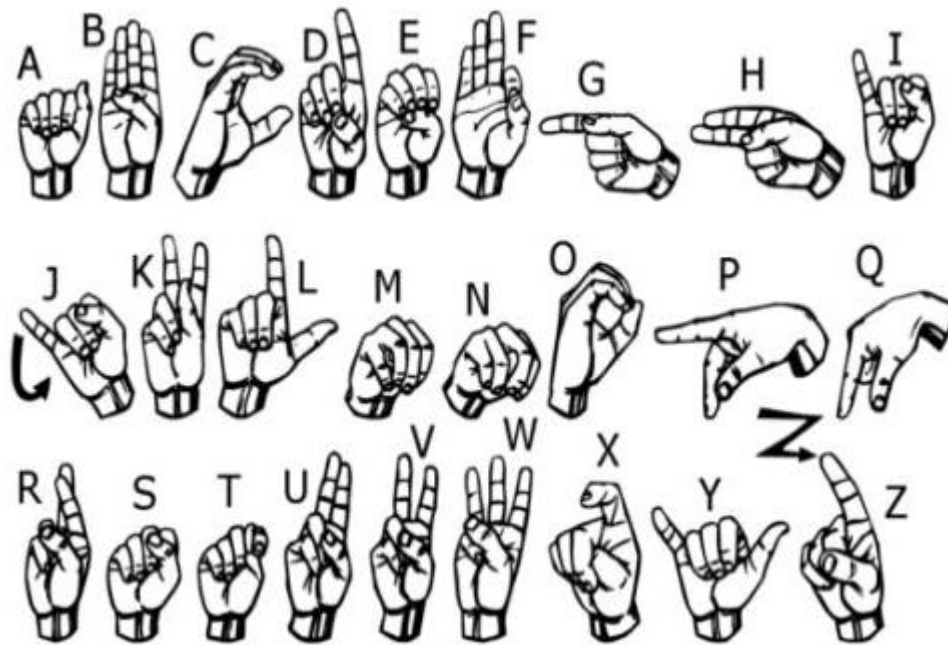


Figure 2. Langage signe américain

En gardant cela à l'esprit, nous visons également à produire un modèle capable de reconnaître les gestes de la main basés sur l'orthographe afin de former un mot complet en combinant chaque geste. Les gestes que nous visons à former sont ceux donnés dans l'image ci-dessus.



Chapitre 1 : Contexte général du projet

1. Motivation

Il est très difficile pour les personnes sourdes de communiquer avec la personne entendante et il n'y a pas beaucoup d'options disponibles pour les aider. Et toutes les alternatives ont des défauts majeurs. Les interprètes ne sont généralement pas disponibles et sont chers. Un stylo et du papier n'est pas non plus une bonne idée, c'est inconfortable, salissant et même chronophage, à la fois pour les sourds et les entendants. Avec l'évolution de l'IoT, tout s'automatise. La demande pour le Machine Learning et ses applications est très élevée. La précision et l'efficacité de tout algorithme et du modèle développé doivent être très élevées pour le rendre utile. La connaissance du Machine Learning devient donc très importante.

À l'ère de l'apprentissage automatique où tout devient automatisé, le besoin d'un interprète pour traduire la langue des signes en texte n'est qu'un gaspillage de ressources. La classification des objets, la détection d'objets et le traitement d'images jouent un rôle très important.

Ainsi, l'objectif principal est de combler le fossé entre les individus normaux et les sourds-muets en fournissant un système de traduction automatique.

2. Méthodologie :

2.1 Reconnaissance de langage des signes (SLR) :

Langage des signes est un langage visuel-spatial basé sur des composants positionnels et visuels, tels que la forme des doigts et des mains, l'emplacement et l'orientation des mains, et les mouvements des bras et du corps. Ces composants sont utilisés ensemble pour transmettre le sens d'une idée. La structure phonologique de SL comporte généralement cinq éléments (Figure 3). Chaque geste dans SL est une combinaison de cinq blocs de construction. Ces cinq blocs

représentent les éléments précieux de SL et peuvent être exploités par des systèmes intelligents automatisés pour la reconnaissance SL (SLR).

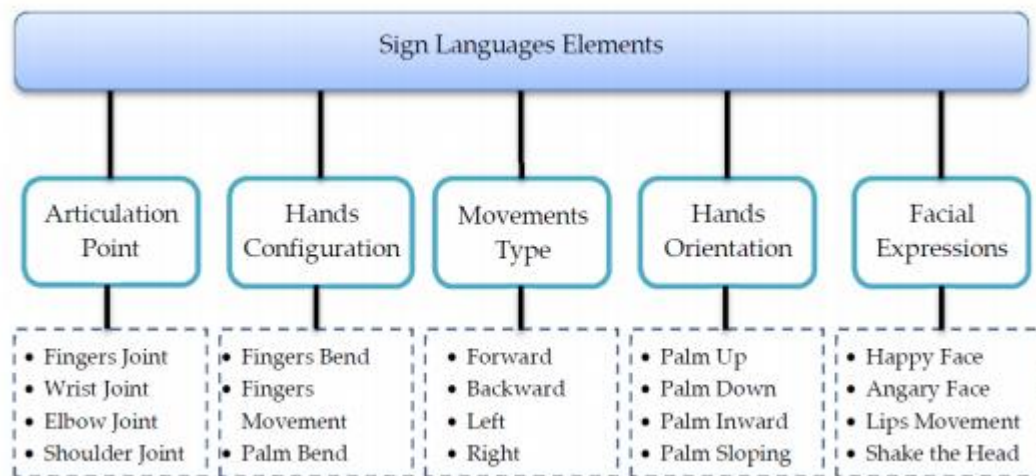


Figure 3. Les éléments essentiels liés à la formation des gestes en langue des signes

Les interventions savantes pour surmonter les difficultés liées au handicap sont multiples et systématiques et varient selon le contexte. L'une des interventions importantes est les systèmes SLR qui sont utilisés pour traduire les signes de SL en texte ou en parole afin d'établir une communication avec des personnes qui ne connaissent pas les signes. Les systèmes SLR basés sur le gant sensoriel sont parmi les efforts les plus importants visant à obtenir des données pour le mouvement des mains humaines. Trois approches (Figure 4), à savoir, basées sur la vision, basées sur des capteurs et une combinaison des deux, sont adoptées pour capturer les configurations de la main et reconnaître les significations correspondantes des gestes.

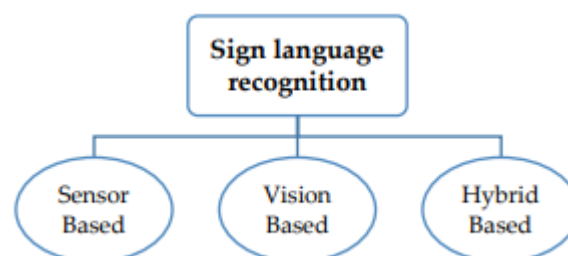


Figure 4. Approches de reconnaissance de la langue des signes

Les systèmes basés sur la vision utilisent des caméras comme outils principaux pour obtenir les données d'entrée nécessaires (Figure 5). Le principal avantage de l'utilisation d'une caméra est qu'elle supprime le besoin de capteurs dans les gants sensoriels et réduit les coûts de construction du système. Les caméras sont assez bonnes marché et la plupart des ordinateurs portables utilisent une caméra de haute spécification en raison du flou causé par une caméra Web. Cependant, malgré la caméra de haute spécification, que la plupart des smartphones possèdent, il existe divers problèmes tels que le champ de vision limité du dispositif de capture, les coûts de calcul élevés et le besoin de plusieurs caméras pour obtenir des caméras robustes. Ces problèmes sont inhérents à ce système et rendent l'ensemble du système futile pour le développement d'applications de reconnaissance en temps réel. Les meilleures performances du système atteignent une précision de 95%.

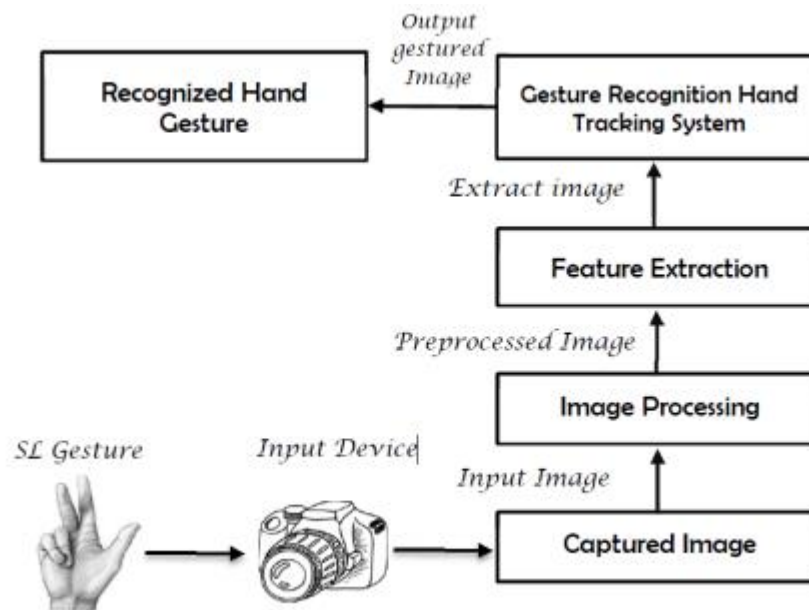


Figure 5. Un organigramme des étapes utilisées dans le système basé sur la vision pour SLR

L'utilisation d'un certain type de gants instrumentés équipés de divers capteurs, à savoir des capteurs de flexion (ou de courbure), des accéléromètres (ACC), des capteurs de proximité et des

capteurs d'abduction, est une approche alternative pour acquérir des données liées aux gestes (Figure 6). Ces capteurs sont utilisés pour mesurer les angles de flexion des doigts, l'abduction entre les doigts et l'orientation (roulis, tangage et lacet) du poignet. Les degrés de liberté réalisables à l'aide de tels gants varient de 5 à 22, selon le nombre de capteurs intégrés dans le gant. Un avantage majeur des systèmes basés sur des gants par rapport aux systèmes basés sur la vision est que les gants peuvent signaler directement les données pertinentes et requises (degré de courbure, inclinaison, etc.) en termes de valeurs de tension au dispositif informatique, éliminant ainsi le besoin pour traiter les données brutes en valeurs significatives. En revanche, les systèmes basés sur la vision doivent appliquer des algorithmes de suivi et d'extraction de caractéristiques spécifiques aux flux vidéo bruts, augmentant ainsi la surcharge de calcul.

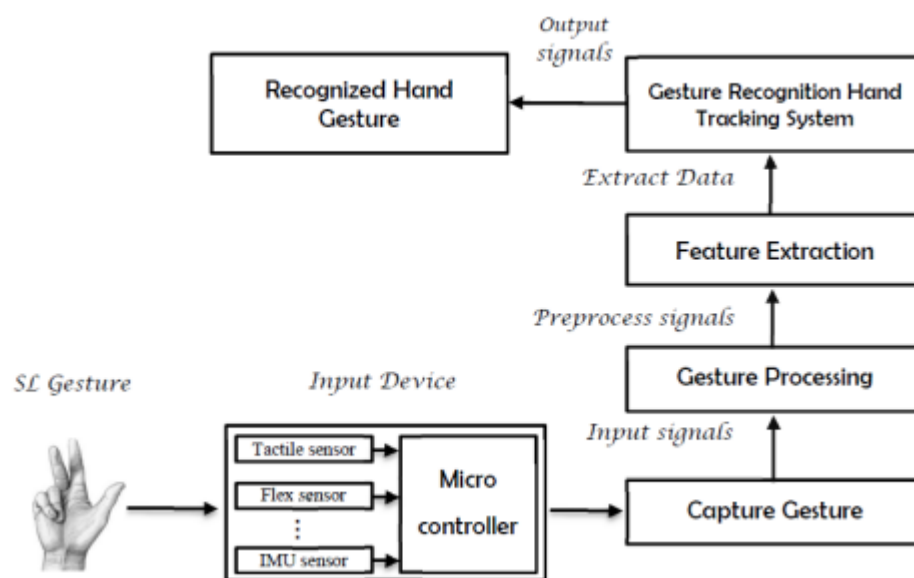


Figure 6. Les phases de collecte et de reconnaissance des données de gestes SL à l'aide du système à base de gants.

La troisième méthode de collecte de données gestuelles brutes utilise une approche hybride qui combine des systèmes basés sur des gants et des caméras. Cette approche utilise l'élimination des erreurs mutuelles pour améliorer l'exactitude et la précision globales. Cependant, peu de travaux ont été effectués dans cette direction en raison du coût et des frais généraux de calcul de l'ensemble de l'installation. Néanmoins, les systèmes de réalité augmentée produisent des résultats prometteurs lorsqu'ils sont utilisés avec une méthodologie de suivi hybride.

2.2. Matériels et méthodes

Les efforts minutieux et intensifs pour trouver des solutions réalistes et réalisables pour surmonter les obstacles à la communication sont des défis majeurs rencontrés par les sourds-muets. Par conséquent, l'accent a été mis sur la mise en œuvre d'un système qui identifie SL dans ses branches logicielles et matérielles.

2.2.1. Concernant les matériaux du système

En ce qui concerne les composants matériels, le système de reconnaissance à base de gants est composé de trois unités principales (Figure 7) : l'entrée, le traitement et la sortie.

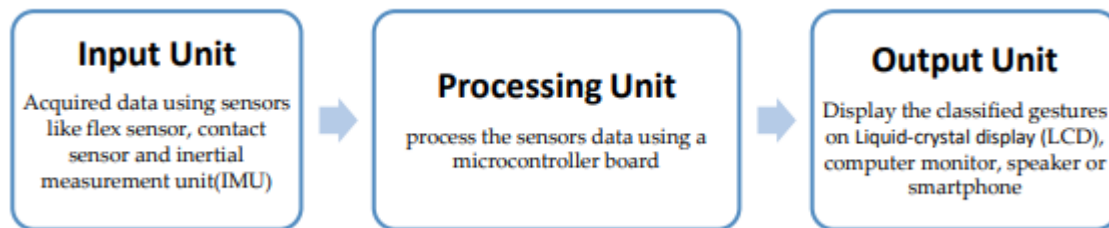


Figure 7. Les principaux composants matériels du système à base de gants.

2.2.1.1. L'unité d'entrée

En raison des développements scientifiques et techniques dans le domaine des circuits électroniques, les capteurs ont attiré l'attention, ce qui a entraîné des investissements dans des fonctionnalités de capteur dans de nombreuses petites et grandes applications. Le capteur est le principal acteur de la mesure des données de la main en termes de flexion (forme), de mouvement, de rotation et de position de la main.

a. Capteurs utilisés pour détecter la flexion des doigts

Le mouvement le plus important qui peut être effectué par les quatre doigts (petit doigt, anneau, majeur et index) est de se pencher vers la paume puis de revenir à la position initiale. Le pouce a

des avantages uniques sur les autres doigts, lui permettant ainsi de se déplacer librement dans six degrés de liberté. En général, le mouvement prédominant en SL lié aux doigts est la flexion.

L'inclinaison des doigts peut être détectée en utilisant différentes méthodes, comme le montre la littérature. Le capteur Flex (Figure 8), qui détermine la quantité de courbure du doigt utilisée par un grand nombre de chercheurs et de développeurs. La technologie du capteur Flex est basée sur des éléments résistifs en carbone. Lorsque le substrat est plié, le capteur produit une sortie de résistance corrélée au rayon de pliage : plus le rayon est petit, plus la valeur de résistance est élevée. Ainsi, la résistance du capteur de flexion augmente à mesure que le corps du composant se plie. Le capteur de flexion est très fin et léger, il est donc également très confortable.

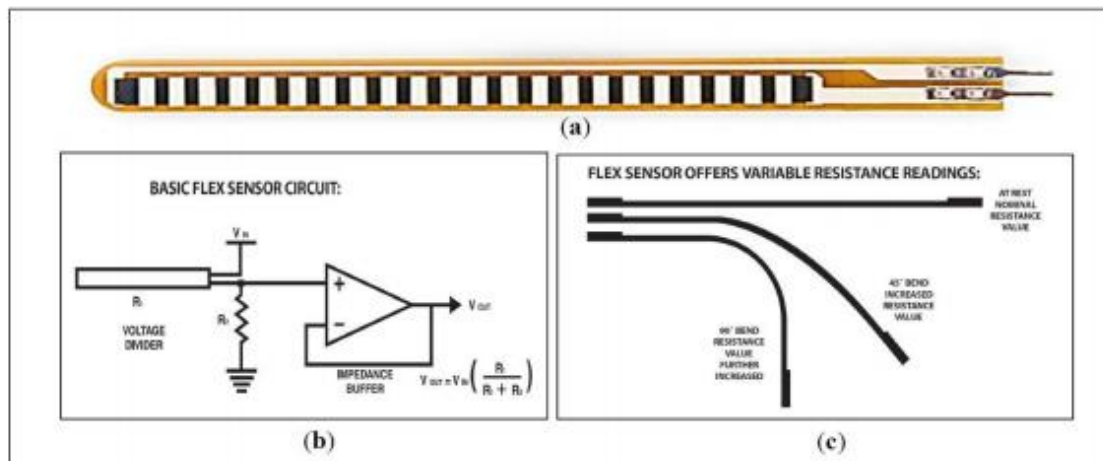


Figure 8. (a) Capteur de flexion, (b) niveaux de flexion et (c) circuit diviseur de tension [2].

b. Capteurs utilisés pour détecter le mouvement et l'orientation de la main

Considérant que les postures SL sont constituées de mouvements de la main et du poignet, elles doivent être prises en compte. Malgré les avantages de l'utilisation de capteurs pour déterminer la forme du doigt, les mouvements de la main ne peuvent pas être distingués par ces capteurs. Les caractéristiques du capteur ACC lui permettent de distinguer le mouvement et la rotation du poignet comme un autre facteur en plus de sa capacité à déterminer la forme du doigt. Par conséquent, l'ACC à trois axes qui fournit les différences d'accélération le long de chaque axe est utilisé pour capturer l'orientation et le mouvement du poignet afin de représenter correctement la fonction importante

d'un gant à capteur. L'ADXL335 (Figure 9) est un accéléromètre 3 axes mince, petit, de faible puissance, avec des sorties de tension conditionnées par signal. Il mesure l'accélération avec une plage de pleine échelle minimale de 3g. Cet appareil mesure l'accélération statique de la gravité dans les applications de détection d'inclinaison et l'accélération dynamique résultant du mouvement, des chocs ou des vibrations. L'ADXL335 contient une structure micro-usinée de surface en polysilicium construite sur une plaquette de silicium. Des ressorts en polysilicium suspendent la structure sur la surface de la plaquette et offrent une résistance aux forces d'accélération. Un condensateur différentiel, constitué de plaques fixes indépendantes fixées à la masse en mouvement, mesure la déflexion de la structure. L'accélération déséquilibre le condensateur, ce qui, à son tour, génère une sortie de capteur avec une amplitude proportionnelle à l'accélération subie.



Figure 9. L'ACC à 3 axes ADXL335 avec une broche analogique à trois sorties x, y et z.

2.2.1.2. Unité de traitement

Le microcontrôleur est l'esprit du système qui est chargé de collecter les données des capteurs fournis par le gant et d'effectuer le traitement requis de ces données pour reconnaître et transférer le signe vers le port de sortie pour être présenté à l'étape finale. Un microcontrôleur hautes performances doté d'une puce avec des microcontrôleurs AVR 8 bits basés sur un ordinateur à jeu d'instructions réduit (RISC), qui combine une mémoire flash de programmation intégrée (ISP) de 32 Ko avec des capacités de lecture-écriture appelées ATmega (Figure 10a). De plus, comme on le trouve dans la littérature, une plate-forme électronique open source appelée Arduino a été utilisée. Plusieurs cartes Arduino sont disponibles sur le marché telles que Arduino Nano, Uno, Mega, etc.

Par exemple, Arduino Uno (Arduino, Italie), (Figure 10c) est basé sur le microcontrôleur ATmega328P et possède 14 entrées/sorties numériques, 6 entrées analogiques, un cristal de quartz 16 MHz et une connexion USB.

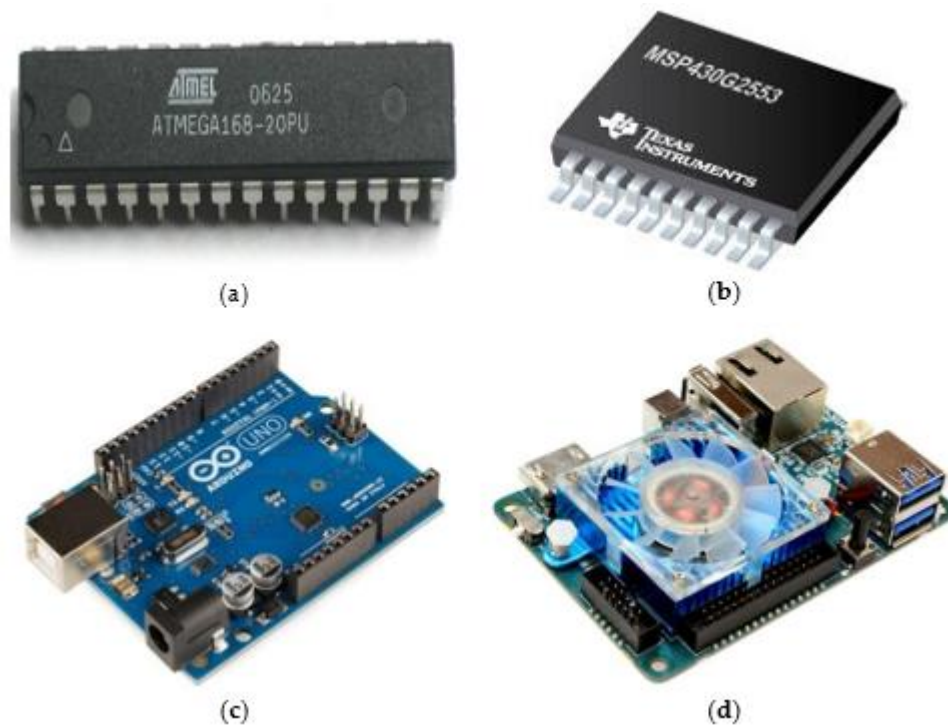


Figure 10. (a) microcontrôleur ATmega, (b) microcontrôleur MSP430G2553, (c) carte Arduino Uno et (d) mini-ordinateur Odroid XU4.

2.2.1.3. Unité de sortie

Habituellement, un utilisateur interagit avec l'appareil via des appareils de sortie, de sorte que les appareils de sortie jouent un rôle important dans l'obtention des meilleures performances des appareils mis en œuvre dans le domaine des reflex. Le principal appareil adopté par les chercheurs comme sortie était l'écran d'ordinateur.

D'autres dispositifs attirant l'attention des chercheurs sont l'écran à cristaux liquides (LCD), le haut-parleur, ou les deux. En fin de compte, le smartphone est une autre alternative choisie pour la sortie du système.



2.2.2 Méthodes d'apprentissage des gestes

Le logiciel, qui est un composant essentiel de tout système, joue un rôle important dans le traitement des données en plus de la possibilité d'améliorer les sorties du système. Le développement de logiciels pour les systèmes SLR est lié aux méthodes utilisées dans le processus de classification pour reconnaître les gestes. L'une des méthodes directes courantes pour effectuer une reconnaissance de posture statique est l'appariement de prototypes (également connu sous le nom d'appariement de modèles statistiques), qui fonctionne sur la base de statistiques pour déterminer la correspondance la plus proche des valeurs d'informations acquises avec des échantillons d'entraînement prédéfinis appelés « modèles ». En fait, cette méthode se caractérise par l'absence de processus d'entraînement compliqués ou d'étalonnage large, augmentant ainsi sa vitesse. Du point de vue de la reconnaissance de formes, le réseau de neurones artificiels (ANN) est la méthode la plus populaire utilisée pour l'apprentissage automatique dans le domaine de la reconnaissance. Par conséquent, il est possible d'entraîner cette technique pour distinguer les gestes statiques et dynamiques, ainsi que la classification de la posture, sur la base des données obtenues à partir du gant de données.

2.2.3. Ensembles de données d'entraînement

L'attention portée au gant sensoriel est due à sa capacité à capturer les données nécessaires pour photographier la forme et le mouvement de la main dans le but de reconnaître les gestes de SL. Malgré le fait que SL a beaucoup de postures, de nombreuses études se sont concentrées sur un ensemble sélectif de postures qui peuvent être aussi petites que certaines lettres de l'alphabet ; ou les postures des mots les plus fréquemment utilisés ; ou une combinaison de l'alphabet, des nombres et des mots pour effectuer leurs expériences et développer un système SLR. D'autres ont contribué à l'élargissement de la base de données des gestes alloués dans le but de développer un système pour les distinguer et les faire inclure soit des alphabets entiers, des nombres dans la plage de 0 à 9 , ou les deux, dans le même système. En outre, d'autres ont contribué à leur effort pour distinguer certains

mots et expressions, voire des phrases, choisis pour couvrir un large éventail de circonstances de la vie réelle, telles que la famille, les achats, l'éducation, le sport, etc.

La figure 10 illustre le nombre d'articles dans chaque variété des gestes susmentionnés.

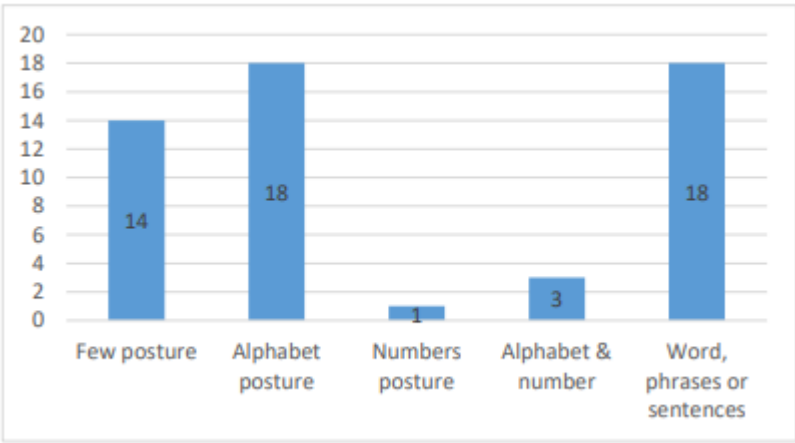


Figure 11. Nombre d'articles sur chaque variété de gestes.



Chapitre 2 : un système de traduction signe-lettre à l'aide d'un gant à main

1. Introduction

L'objectif est de développer et de mettre en œuvre un système de traduction en langue des signes pour aider les personnes malentendantes et malentendantes à communiquer en temps réel en utilisant la langue des signes indienne. C'est pourquoi nous avons construit un traducteur en langue des signes qui utilise un gant équipé de capteurs capables d'interpréter les 26 lettres anglaises en langue des signes américaine (ASL). Le gant portable utilise des capteurs et des accéléromètres pour collecter des données sur la position de chaque doigt et le mouvement de la main. L'alphabet signé est ensuite affiché sur l'écran LCD ainsi que prononcé par les haut-parleurs.

2. Objectifs

- Sélection de capteurs flexibles, de capteurs de contact et d'accéléromètres adaptés.
- Conversion des signaux analogiques détectés en signaux numériques à l'aide de circuits intégrés appropriés.
- Interfaçage des capteurs, accéléromètres et émetteur radio avec le microcontrôleur de l'unité de détection.
- Interfaçage du microcontrôleur de la station de base avec le récepteur, l'écran LCD et les haut-parleurs.

3. Méthodologie

Le gant utilise des capteurs de flexion, des capteurs de contact et des accéléromètres en trois dimensions pour collecter des données sur la position de chaque doigt et le mouvement de la main pour différencier les lettres. Les sorties des capteurs et de l'accéléromètre sont transmises à l'unité de détection placée sur le bras.

L'unité de détection se compose d'un ADC (convertisseur analogique-numérique) pour numériser la sortie détectée qui est ensuite transmise au microcontrôleur basé sur l'AVR. La traduction est transmise à l'écran LCD et aux haut-parleurs pour afficher et prononcer respectivement la lettre.

3.1 Architecture globale

La figure 11 montre le système proposé pour le signe à la lettre Traducteur (S2L). Le système se compose d'un gant tenant flex capteurs, composants discrets, un microcontrôleur et un écran LCD.

Le gant est composé de cinq capteurs de flexion, de fines bandes qui détectent les changements de résistance indiquant quand un doigt est plié.

Au dos de la main se trouve un circuit micro-contrôleur, cœur du système qui analyse tous ces signaux entrants et les transmet à un Mini-LCD afin d'afficher les lettres résultantes.

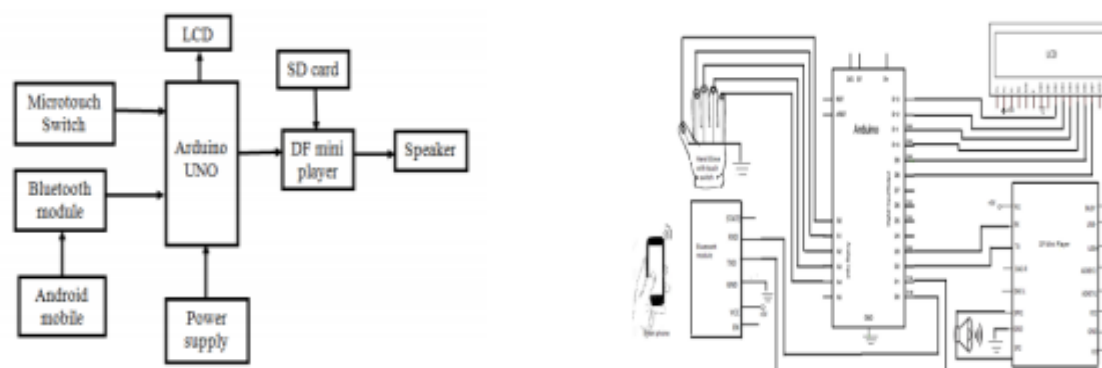


Figure 12. Schéma fonctionnel proposé

Parce que le gant doit être placé dans la main du sourd ; la conception doit répondre aux exigences suivantes :

- Le système fournit une collecte et une sortie de données précises.
- Le système est portable.
- Le système est facile à installer.
- Le système est facile à utiliser.

- Le système est sûr à utiliser.
- Le système est durable.

3.2 Mise en Œuvre Matérielle du Système Proposé

Le schéma fonctionnel du système (S2L) proposé est illustré à la figure 12. Il est principalement composé des résistances flexibles, d'une alimentation 9 volts et d'un circuit matériel. Les entrées sont les signes des capteurs flexibles qui sont connectés au circuit matériel et à l'alimentation. Le circuit matériel comprend un microcontrôleur, des composants discrets et un écran LCD.

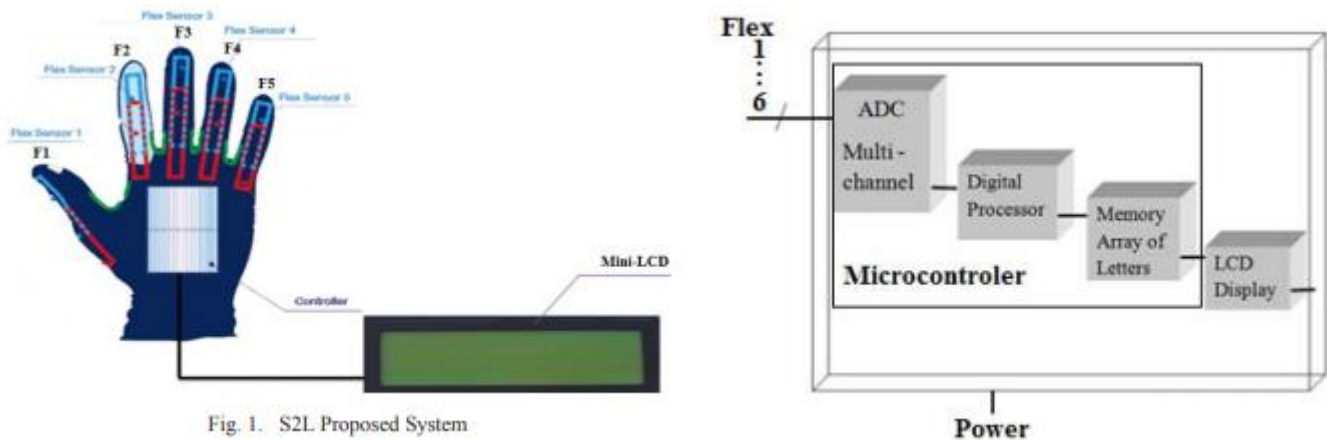


Figure 13. Schéma fonctionnel du S2L proposé

Les résistances de flexion illustrées à la figure 13 sont des éléments essentiels qui jouent le rôle d'obtenir les signes qui sont les entrées de l'utilisateur tout en détectant le changement de la quantité de courbure appliquée sur le flex. Ils convertissent le changement de courbure en résistance électrique - plus la courbure est importante, plus la valeur de résistance est élevée. Ils sont utilisés dans un circuit diviseur de tension pour fournir les données (signal analogique) au microcontrôleur après avoir été numérisées en 10 bits de précision pour être traitées.



Figure 14. Capteurs flexibles

Les formules utilisées pour obtenir les tensions et les 10 bits sont représentés respectivement dans les équations (1) et (2).

$$V = \frac{R}{(R+5)} \times 5 \quad (1)$$

Où R est la résistance mesurée.

$$X = \text{Decimal value}(10 \text{ bit} - \text{equivalent}) = V \times \left(\frac{1024}{5}\right) \quad (2)$$

Les portées prises étaient approximativement proches pour les différents doigts et la même position. Les différentes plages étaient intitulées comme suit : OPEN, MID1, MID2 et CLOSE. Le tableau I illustre les différentes valeurs pour chaque flex.

Flex	OPEN (00)	MID1 (01)	MID2 (10)	CLOSE (11)
F1	X<670	669<X<715	714<X<738	X>737
F2	X<675	674<X<760	759<X<808	X>807
F3	X<662	661<X<700	699<X<740	X>739
F4	X<740	739<X<795	794<X<840	X>839
F5	X<700	699<X<750	749<X<790	X>789
F6	X>600			X<600

Tableau 1. Angles des plages de flexibilité pour différentes positions

Où X est les tensions d'entrée numérisées pour les flexions F1 à F5 connectées à 5 doigts, et la flexion F6 placée sur le poignet pour détecter son mouvement (s'il est droit ou plié).

Sur la base des données du tableau I et de la liste des signes de chaque lettre, le tableau II a été rempli de zéros et d'uns correspondant à chaque lettre pour les cinq capteurs de flexion.

	F1	F2	F3	F4	F5
A	00	10	10	10	10
B	00	00	00	00	00
C	01	01	01	01	01
D	01	00	01	01	01
E	10	10	10	10	10
F	01	01	00	00	00
G	00	01	10	10	10
H	10	00	00	10	10
I	01	10	10	10	00
J	00	10	10	10	00
K	00	00	00	10	10
L	00	00	10	10	10
M	10	01	01	01	10
N	10	01	01	10	10
O	10	01	01	01	01
P	00	00	00	11	11
Q	00	00	11	11	11
R	01	01	00	10	10
S	10	11	11	11	11
T	01	01	10	10	10
U	01	00	00	10	10
V	01	00	00	11	11
W	01	00	00	00	01
X	10	01	11	11	11
Y	00	10	10	10	00
Z	10	00	11	11	11

Tableau 2. Binaire de chaque lettre pour chaque capteur flex

Le microcontrôleur est l'unité centrale. Il fonctionne avec une horloge de 4 MHz pour fournir suffisamment de données à ses composants connectés tout en recevant des données précises. Il possède de nombreuses fonctionnalités spéciales dont une unité centrale de traitement et un convertisseur A/N 10 bits et 8 canaux. En plus du microcontrôleur, le PCB contient des composants discrets, un oscillateur et un régulateur de tension qui convertit les 9 volts fournis par la batterie en 5 volts nécessaires au fonctionnement du circuit du microcontrôleur.

La figure 15 montre l'ensemble du système, y compris le gant.

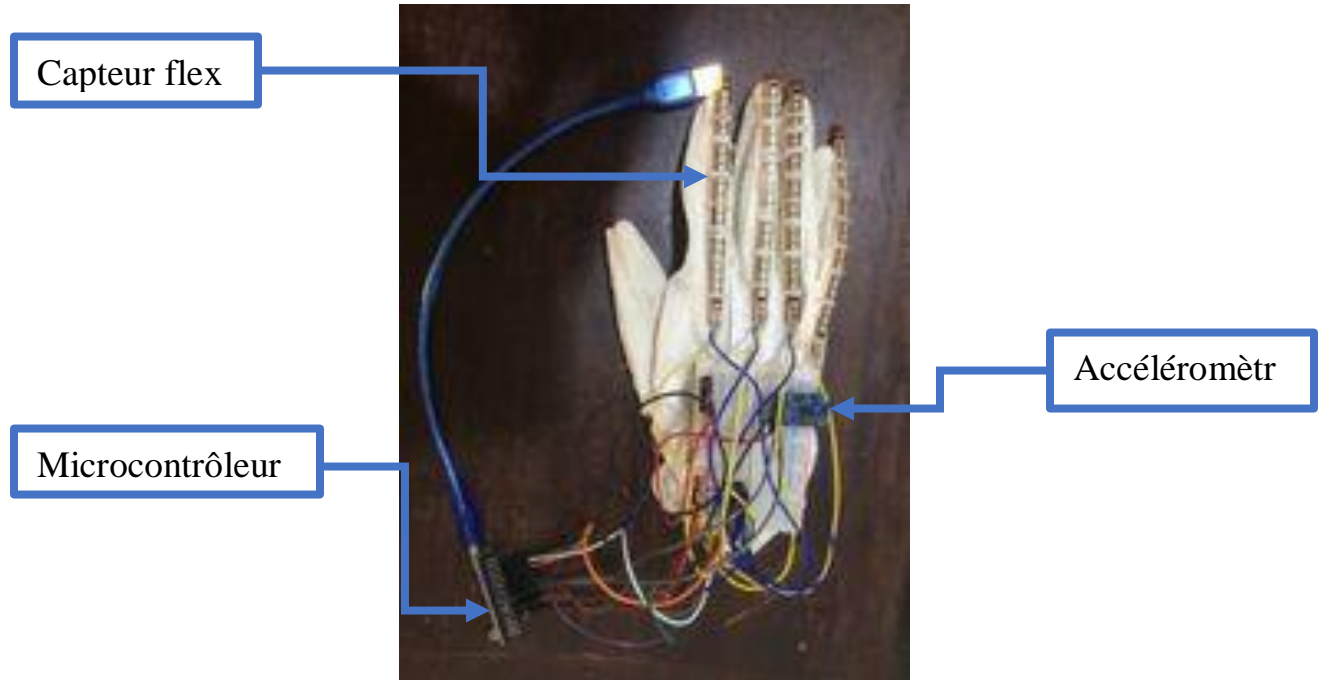


Figure 15. L'ensemble du système

Après avoir construit le matériel, le logiciel est développé pour afficher les lettres de sortie correspondant aux informations d'entrée provenant des capteurs. L'organigramme présenté à la figure 15 illustre les étapes du programme. Dans la phase d'initialisation, les tables de codes ont été définies. Le programme commence par lire les entrées analogiques d'origine des capteurs flex (F1 F6) et en réservant deux positions de bit pour chaque valeur flex.

Le code est construit en concaténant les représentations binaires de tous les flex (F1 à F2) comme le montre la figure 16. Ensuite, le code subit des conditions if puis appelle une fonction afin d'atteindre la sortie finale, la lettre adéquate après laquelle les quelques les dernières étapes seront répétées à travers une boucle pour obtenir les nouvelles données.

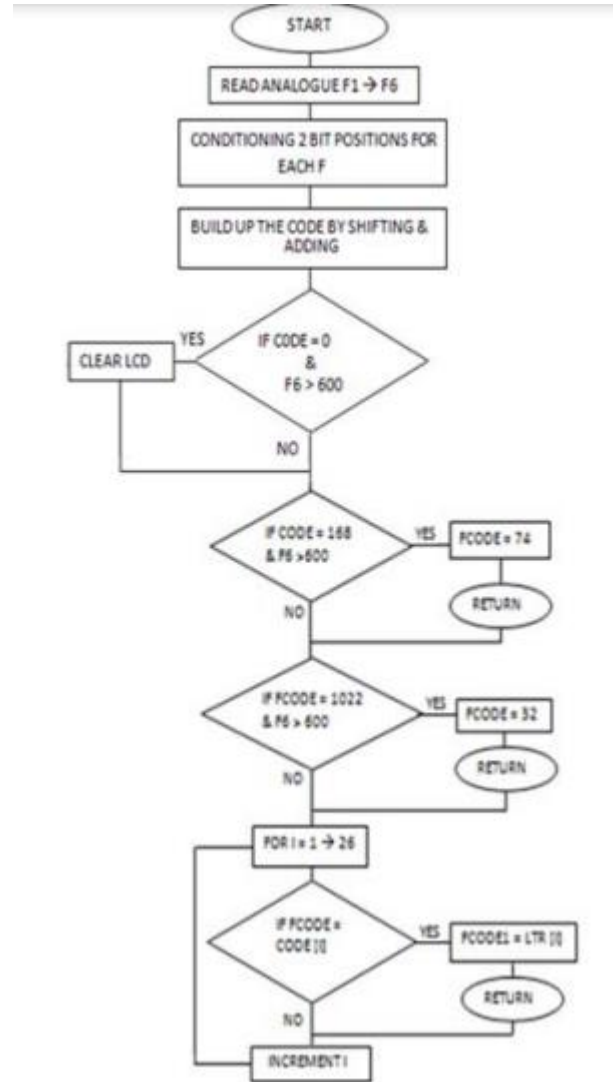


Figure 16. Organigramme du programme S2L

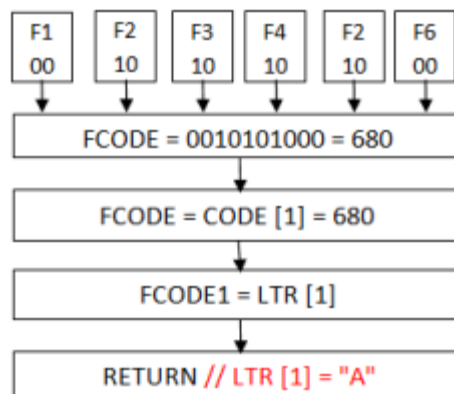


Figure 17. Représentation en bits de la lettre A

L'utilisateur portant le gant peut utiliser la langue des signes en déplaçant simplement les doigts appropriés de sa main. Étant donné que chaque capteur flexible a une gamme de résistances, chaque capteur a été testé séparément et ses valeurs de tension ont été prises pour les vingt-six lettres. Lors des tests, les valeurs ont changé, le problème a donc été résolu en prenant différentes plages pour chaque position de flexion. La figure 17 ci-dessous montre la représentation du signe pour la lettre "A" et le tableau III illustre les différentes résistances, tensions testées pour les six capteurs flexibles pour cette lettre, représentation 10 bits, positions des doigts (soit ouvert, milieu1, milieu2 ou fermé), et la représentation binaire de ces positions.



Figure 18. *Signe de la lettre A*

	Resistance (kΩ)	Voltage (V)	Decimal 10-bits	Finger position	Binary
F1	9.9	3.3221	680.38	Mid1	01
F2	15.3	3.7685	771.78	mid2	10
F3	13.8	3.6702	751.66	mid2	10
F4	10.9	3.428	701.99	mid2	10
F5	14.5	3.718	761.44	mid2	10
F6	8.57	3.158	646.7	open	00

Tableau 3. *Exemple de valeurs de la lettre "A"*

Ce système a atteint une précision moyenne de 94% en essayant d'afficher toutes les lettres. La figure 18 montre les valeurs moyennes de précision pour chaque lettre.




Figure 19. Précision du système S2L

4. Conclusion et Plans Futurs

L'objectif du système Sign to Letter Translator est de détecter tout changement concernant le geste de la main qui fournit suffisamment de données ou d'informations pour permettre au système de traduire le signe en lettre. Ce système s'est avéré suffisamment précis, fiable et abordable, ce qui permet à toute personne souffrant de déficience auditive de l'utiliser chaque fois qu'elle a besoin de communiquer avec d'autres personnes. Des systèmes, similaires à cette conception qui utilise des technologies de communication et des capteurs avancés, ouvrent de nouveaux horizons pour améliorer nos modes de vie.

Des plans futurs sont en cours d'élaboration afin d'améliorer le système proposé. Premièrement, des capteurs flexibles ou des gyroscopes et/ou des accéléromètres de meilleure qualité seront mis en œuvre dans ce système afin d'améliorer ses performances et de réduire les erreurs. Deuxièmement, le système deviendra sans fil pour transmettre des données à un téléphone intelligent, ce qui devrait éliminer l'utilisation d'un écran LCD et tirer parti des applications texte-voix disponibles dans les téléphones intelligents.



Chapitre 3 : un système de traduction signe-lettre à l'aide des réseaux de neurones

1. Introduction

La langue des signes est l'une des formes de langue de communication les plus anciennes et les plus naturelles, mais comme la plupart des gens ne connaissent pas la langue des signes et que les interprètes sont très difficiles à trouver, nous avons mis au point une méthode en temps réel utilisant des réseaux de neurones pour l'orthographe digitale basée sur le Langue de signe américain. Dans notre méthode, la main est d'abord passée à travers un filtre et après l'application du filtre, la main est passée à travers un classificateur qui prédit la classe des gestes de la main. Notre méthode fournit une précision de 95,7 % pour les 26 lettres de l'alphabet.

2. Enquête bibliographique

Ces dernières années, des recherches considérables ont été menées sur la reconnaissance des gestes de la main.

À l'aide d'une étude bibliographique effectuée, nous avons réalisé que les étapes de base de la reconnaissance des gestes de la main sont :

- L'acquisition des données
- Prétraitement des données
- Extraction de caractéristiques
- Classement des gestes

a. L'acquisition des données :

Les différentes approches pour acquérir des données sur le geste de la main peuvent se faire des manières suivantes :



i. Utilisation d'appareils sensoriels

Il utilise des dispositifs électromécaniques pour fournir une configuration et une position exactes de la main. Différentes approches basées sur des gants peuvent être utilisées pour extraire des informations. Mais elles sont coûteuses et peu conviviales.

ii. Approche basée sur la vision

En vue méthodes basées sur la caméra d'ordinateur est le périphérique d'entrée pour observer les informations des mains ou des doigts. Les méthodes basées sur la vision ne nécessitent qu'une caméra, réalisant ainsi une interaction naturelle entre les humains et les ordinateurs sans l'utilisation d'appareils supplémentaires. Ces systèmes tendent à compléter la vision biologique en décrivant des systèmes de vision artificielle qui sont implémentés dans des logiciels et/ou du matériel.

Le principal défi de la détection de la main basée sur la vision est de faire face à la grande variabilité de l'apparence de la main humaine due à un grand nombre de mouvements de la main, aux différentes possibilités de couleur de peau ainsi qu'aux variations des points de vue, des échelles et de la vitesse de la caméra capturant la scène.

b. Prétraitement des données et extraction de caractéristiques:

- L'approche de la détection des mains combine la détection des couleurs basée sur un seuil avec la soustraction de l'arrière-plan. Nous pouvons utiliser le détecteur de visage Adaboost pour différencier les visages et les mains car les deux impliquent une couleur de peau similaire.
- Nous pouvons également extraire l'image nécessaire qui doit être formée en appliquant un filtre appelé flou gaussien. Le filtre peut être facilement appliqué en utilisant la vision par ordinateur ouverte également connue sous le nom d'OpenCV.
- Pour extraire l'image nécessaire qui doit être entraînée, nous pouvons utiliser des gants instrumentés. Cela permet de réduire le temps de calcul pour le prétraitement et peut nous fournir des données plus concises et précises par rapport à l'application de filtres sur les données reçues de l'extraction vidéo.



• Nous avons essayé de segmenter manuellement une image à l'aide de techniques de segmentation des couleurs, mais comme mentionné dans le document de recherche, la couleur et le ton de la peau dépendent fortement des conditions d'éclairage en raison du résultat obtenu pour la segmentation que nous avons essayé de faire. De plus, nous avons un grand nombre de symboles à entraîner pour notre projet, dont beaucoup se ressemblent, comme le geste pour le symbole « V » et le chiffre « 2 », nous avons donc décidé qu'afin de produire de meilleures précisions pour notre grand nombre de symboles, plutôt que en segmentant la main sur un arrière-plan aléatoire, nous gardons l'arrière-plan de la main d'une seule couleur stable afin que nous n'ayons pas besoin de le segmenter sur la base de la couleur de la peau. Cela nous aiderait à obtenir de meilleurs résultats.

c. Classement des gestes :

• Hidden Markov Models (HMM) est utilisé pour la classification des gestes. Ce modèle traite des aspects dynamiques des gestes. Les gestes sont extraits d'une séquence d'images vidéo en suivant les taches de couleur de peau correspondant à la main dans un corps– espace du visage centré sur le visage de l'utilisateur. L'objectif est de reconnaître deux classes de gestes : déictique et symbolique. L'image est filtrée à l'aide d'une table d'indexation rapide. Après filtrage, les pixels de couleur de peau sont regroupés en blobs. Les blobs sont des objets statistiques basés sur la localisation (x,y) et la colorimétrie (Y,U,V) des pixels de couleur de peau afin de déterminer des zones homogènes.

• Naïve Bayes Classifier est utilisé, qui est une méthode efficace et rapide pour la reconnaissance statique des gestes de la main. Elle repose sur la classification des différents gestes selon des invariants à base géométrique qui sont obtenus à partir de données d'images après segmentation. Ainsi, contrairement à de nombreuses autres méthodes de reconnaissance, cette méthode n'est pas dépendante de la couleur de peau. Les gestes sont extraits de chaque image de la vidéo, avec un fond statique. La première étape consiste à segmenter et étiqueter les objets d'intérêt et à en extraire des invariants géométriques. La prochaine étape est la classification des gestes en utilisant un algorithme de voisin le plus proche K assisté d'un algorithme de pondération de distance (KNNDW) pour fournir des données appropriées pour un Naïve Bayes pondéré localement' classificateur.



- Selon un article sur la « Reconnaissance des gestes de la main humaine à l'aide d'un réseau de neurones à convolution » par Hsien-I Lin, Ming- Hsiang Hsu et Wei-Kai Chen diplômés de l'Institute of Automation Technology National Taipei University of Technology Taipei, Taiwan, ils construisent une peau modèle pour extraire la main d'une image, puis appliquer un seuil binaire à l'ensemble de l'image. Après avoir obtenu l'image seuil ils la calibrent autour de l'axe principal afin de centrer l'image autour de celui-ci. Ils entrent cette image dans un modèle de réseau neuronal convolutif afin d'entraîner et de prédire les sorties. Ils ont entraîné leur modèle sur 7 gestes de la main et en utilisant leur modèle, ils produisent une précision d'environ 95% pour ces 7 gestes.

3. Mots clés et définitions

3.1 Extraction et représentation de caractéristiques :

La représentation d'une image sous forme de matrice 3D ayant une dimension comme hauteur et largeur de l'image et la valeur de chaque pixel comme profondeur (1 en cas de niveaux de gris et 3 en cas de RVB). De plus, ces valeurs de pixels sont utilisées pour extraire des caractéristiques utiles à l'aide de CNN.

3.2 Réseaux de neurones artificiels :

Le réseau de neurones artificiels est une connexion de neurones, reproduisant la structure du cerveau humain. Chaque connexion de neurone transfère des informations à un autre neurone. Les entrées sont introduites dans la première couche de neurones qui les traite et les transfère à une autre couche de neurones appelée couches cachées. Après le traitement des informations à travers plusieurs couches de couches cachées, les informations sont transmises à la couche de sortie finale.

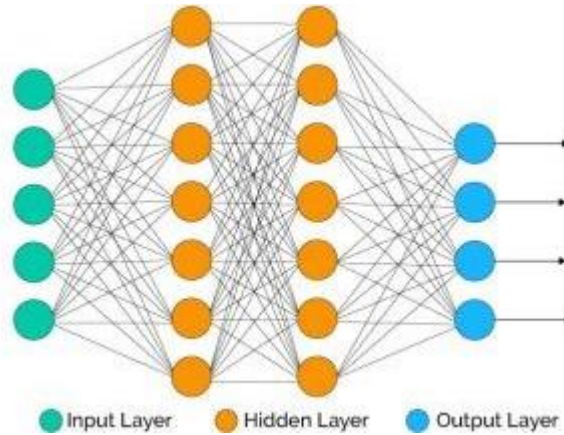


Figure 20. Réseaux de neurones artificiels

Ils sont capables d'apprendre et ils doivent être formés. Il existe différentes stratégies d'apprentissage :

1. Apprentissage non supervisé
2. Enseignement supervisé
3. Apprentissage par renforcement

3.3 Réseau de neurones à convolution :

Contrairement aux réseaux de neurones ordinaires, dans les couches de CNN, les neurones sont disposés en 3 dimensions : largeur, hauteur, profondeur. Les neurones d'une couche ne seront connectés qu'à une petite région de la couche (taille de la fenêtre) qui la précède, au lieu de tous les neurones de manière entièrement connectée. De plus, la couche de sortie finale aurait des dimensions (nombre de classes), car à la fin de l'architecture CNN, nous réduirons l'image complète en un seul vecteur de scores de classe.

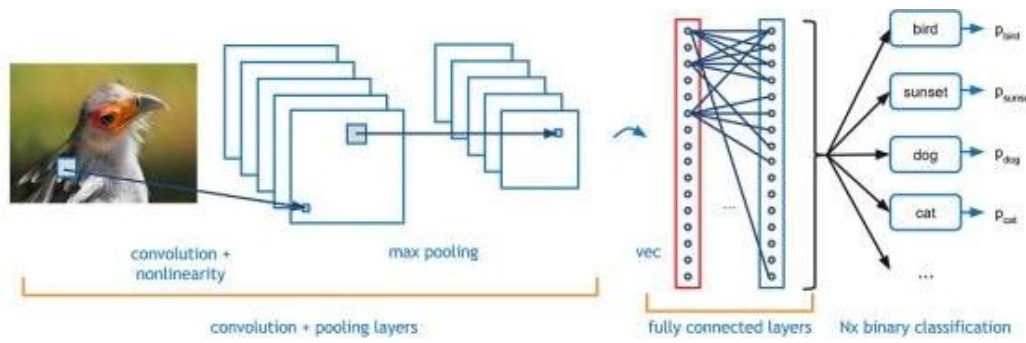


Figure 21. Réseaux de neurones à convolution

3.3.1 Couche de convolution :

Dans la couche de convolution, nous prenons une petite taille de fenêtre [généralement de longueur 5×5] qui s'étend jusqu'à la profondeur de la matrice d'entrée. La couche se compose de filtres apprenables de la taille de la fenêtre. Au cours de chaque itération, nous avons fait glisser la fenêtre par taille de foulée [généralement 1] et calculons le produit scalaire des entrées de filtre et des valeurs d'entrée à une position donnée. Au fur et à mesure que nous poursuivons ce processus, créez une matrice d'activation bidimensionnelle qui donne la réponse de cette matrice à chaque position spatiale. C'est-à-dire que le réseau apprendra des filtres qui s'activent lorsqu'ils voient un certain type de caractéristique visuelle telle qu'un bord d'une certaine orientation ou une tache d'une certaine couleur.

3.3.2 Couche de mise en commun :

Nous utilisons la couche de mise en commun pour diminuer la taille de la matrice d'activation et finalement réduire les paramètres d'apprentissage. Il existe deux types de mutualisation :

a) Mise en commun maximale : Dans max pooling, nous prenons une taille de fenêtre [par exemple une fenêtre de taille 2×2], et ne prenons que le maximum de 4 valeurs. Couvrez bien cette fenêtre et continuez ce processus, alors obtenez enfin une matrice d'activation la moitié de sa taille d'origine.

b) Mise en commun moyenne : Dans la mise en commun moyenne, nous prenons la moyenne de toutes les valeurs dans une fenêtre.

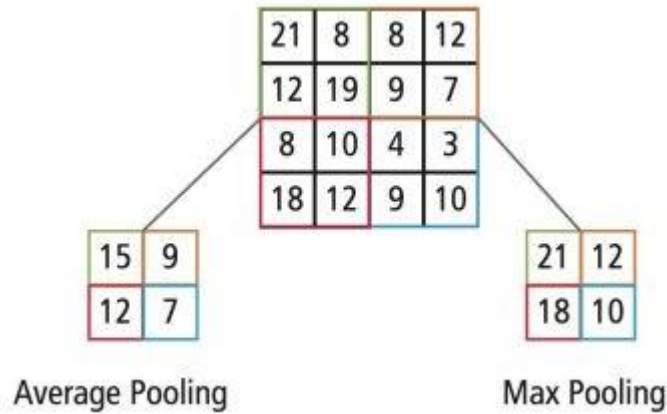


Figure 22. Types de mise en commun

3.3.3 Couche entièrement connectée :

Dans la couche de convolution, les neurones ne sont connectés qu'à une région locale, tandis que dans une région entièrement connectée, connectez bien toutes les entrées aux neurones.

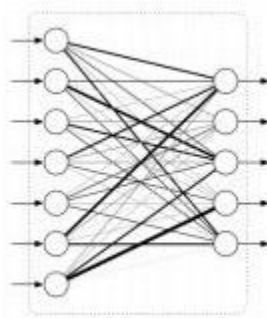


Figure 23. Couche entièrement connectée

3.3.4 Couche de sortie finale :

Après avoir obtenu les valeurs de la couche entièrement connectée, connectez-les bien à la couche finale de neurones [ayant un nombre égal au nombre total de classes], qui prédira la probabilité que chaque image appartienne à des classes différentes.

3.4 TensorFlow

Tensorflow est une bibliothèque logicielle open source pour le calcul numérique. On définit d'abord les nœuds du graphe de calcul, puis à l'intérieur d'une session, le calcul réel a lieu. TensorFlow est largement utilisé dans le MachineLearning.

3.5 Keras

Keras est une bibliothèque de réseaux neuronaux de haut niveau écrite en python qui fonctionne comme un wrapper pour TensorFlow. Il est utilisé dans les cas où nous voulons construire et tester rapidement le réseau de neurones avec un minimum de lignes de code. Il contient des implémentations d'éléments de réseau neuronal couramment utilisés tels que des couches, des objectifs, des fonctions d'activation, des optimiseurs et des outils pour faciliter le travail avec des images et des données textuelles.

3.6 OpenCV

OpenCV (Open Source Computer Vision) est une bibliothèque open source de fonctions de programmation utilisées pour la vision par ordinateur en temps réel. Il est principalement utilisé pour le traitement d'images, la capture vidéo et l'analyse de fonctionnalités telles que la reconnaissance des visages et des objets. Il est écrit en C++ qui est son interface principale, cependant des liaisons sont disponibles pour Python, Java, MATLAB/ OCTAVE.

4. Méthodologie

Le système est une approche basée sur la vision. Tous les signes sont représentés à mains nues, ce qui élimine le problème de l'utilisation de dispositifs artificiels pour l'interaction.

4.1 Génération d'ensembles de données

Pour le projet, nous avons essayé de trouver des jeux de données déjà créés, mais nous n'avons pas pu trouver de jeu de données sous forme d'images brutes correspondant à nos besoins. Tout ce que nous avons pu trouver, ce sont les ensembles de données sous forme de valeurs RVB. Nous avons donc décidé de créer notre propre ensemble de données. Les étapes que nous avons suivies pour créer notre ensemble de données sont les suivantes.

Nous avons utilisé la bibliothèque Open Computer Vision (OpenCV) afin de produire notre ensemble de données. Premièrement, nous avons capturé environ 800 images de chacun des symboles en ASL à des fins de formation et environ 200 images par symbole à des fins de test.

D'abord, nous capturons chaque image affichée par la webcam de notre machine. Dans chaque cadre, nous définissons une région d'intérêt (ROI) qui est indiquée par un carré vert délimité, comme indiqué dans l'image ci-dessous.

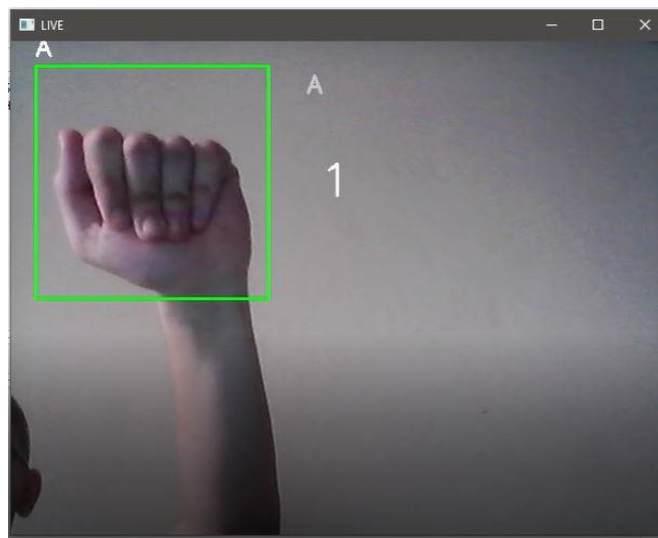


Figure 24. Image affichée par la webcam et la région d'intérêt

De toute cette image, nous extrayons notre retour sur investissement qui est RVB et le convertissons en image en niveaux de gris comme indiqué ci-dessous.



Figure 25. Image convertie en niveau de gris

Enfin, nous appliquons notre filtre de flou gaussien à notre image, ce qui nous aide à extraire diverses caractéristiques de notre image. L'image après application du flou gaussien ressemble à ci-dessous.



Figure 26. Image après l'implication du flou gaussien

4.2 CLASSEMENT DES GESTES

L'approche que nous avons utilisée pour ce projet est :

Notre approche utilise deux couches d'algorithme pour prédire le symbole final de l'utilisateur.

Algorithme Couche 1 :

1. Application d'un filtre de flou gaussien et un seuil à l'image prise avec openCV pour obtenir l'image traitée après l'extraction des caractéristiques.
2. Cette image traitée est transmise au modèle CNN pour la prédiction et si une lettre est détectée pendant plus de 50 images, la lettre est imprimée et prise en compte pour former le mot.
3. L'espace entre les mots est pris en compte à l'aide du symbole blanc.

Algorithme Couche 2 :

1. Nous détectons différents ensembles de symboles qui montrent des résultats similaires lors de la détection.
2. Nous classons ensuite entre ces ensembles à l'aide de classificateurs conçus uniquement pour ces ensembles.

4.2.1 Couche 1

4.2.1.1 Modèle CNN :

1ère couche de convolution : L'image d'entrée a une résolution de 128x128 pixels.

Il est d'abord traité dans la première couche convolutive à l'aide de 32 poids de filtre (3x3 pixels chacun). Cela donnera une image de 126X126 pixels, une pour chaque poids de filtre.

1ère couche de mutualisation : Les images sont sous-échantillonnées en utilisant une mise en commun maximale de 2x2, c'est-à-dire que nous gardons la valeur la plus élevée dans le carré 2x2 du tableau. Par conséquent, notre image est sous-échantillonnée à 63x63 pixels.

2ème couche de convolution : Désormais, ces 63 x 63 provenant de la sortie de la première couche de mise en commun servent d'entrée à la deuxième couche convolutive. Ils sont traités dans la deuxième couche convolutive à l'aide de 32 poids de filtre (3 x 3 pixels chacun).

2ème couche de mutualisation : Les images résultantes sont à nouveau sous-échantillonnées en utilisant un pool maximum de 2x2 et sont réduites à une résolution d'images de 30 x 30.



1ère couche densément connectée : Maintenant, ces images sont utilisées comme entrée d'une couche entièrement connectée avec 128 neurones et la sortie de la deuxième couche convolutive est remodelée en un tableau de $30 \times 30 \times 32 = 28800$ valeurs. L'entrée de cette couche est un tableau de 28 800 valeurs. La sortie de ces couches est transmise à la 2e couche densément connectée. Nous utilisons une couche de décrochage de valeur 0,5 pour éviter le surapprentissage.

2ème couche densément connectée : Maintenant, la sortie de la 1ère couche densément connectée est utilisée comme entrée vers une couche entièrement connectée avec 96 neurones.

Couche finale : La sortie de la 2ème couche densément connectée sert d'entrée pour la couche finale qui aura le nombre de neurones comme nombre de classes que nous classons (alphabets + symbole vide).

4.2.1.2 Fonction d'activation :

Nous avons utilisé ReLu (Unité linéaire rectifiée) dans chacune des couches (neurones convolutifs et neurones entièrement connectés). ReLu calcule $\max(x, 0)$ pour chaque pixel d'entrée. Cela ajoute de la non-linéarité à la formule et aide à apprendre des fonctionnalités plus compliquées. Cela aide à éliminer le problème de gradient de fuite et à accélérer la formation en réduisant le temps de calcul.

4.2.1.3 Couche de mise en commun :

Nous appliquons **Max** pooling à l'image d'entrée avec une taille de pool de (2, 2) avec fonction d'activation relu. Cela réduit la quantité de paramètres diminuant ainsi le coût de calcul et réduit le surapprentissage.

4.2.1.4 Calques de suppression :

Le problème du surapprentissage, où après l'entraînement, les poids du réseau sont tellement adaptés aux exemples d'entraînement qu'ils sont donnés que le réseau ne fonctionne pas bien lorsqu'on lui donne de nouveaux exemples. Cette couche "supprime" un ensemble aléatoire d'activations dans cette couche en les mettant à zéro. Le réseau devrait être capable de fournir la bonne classification ou

sortie pour un exemple spécifique même si certaines des activations sont abandonnées.

4.2.1.5 Optimiseur :

Nous avons utilisé l'optimiseur Adam pour mettre à jour le modèle en réponse à la sortie de la fonction de perte. Adam combine les avantages de deux extensions de deux algorithmes de descente de gradient stochastique à savoir algorithme de gradient adaptatif (ADA) et root mean square propagation (RMSProp).

4.2.2 Couche 2

Nous utilisons deux couches d'algorithmes pour vérifier et prédire les symboles qui sont plus similaires les uns aux autres afin que nous puissions nous rapprocher car nous pouvons détecter le symbole affiché. Lors de nos tests, nous avons constaté que les symboles suivants nes'affichaient pas correctement et en donnaient également d'autres :

1. Pour D : R et U
2. Pour U : D et R
3. Pour I : T, D, K et I
4. Pour S : M et N

Donc, pour gérer les cas ci-dessus, nous avons créé trois classificateurs différents pour classer ces ensembles :

1. {D,R,U}
2. {T,K,D,I}
3. {S,M,N}

4.2.2.1 Formation de phrases d'orthographe au doigt

a. Mise en œuvre :

1. Chaque fois que le nombre d'une lettre détectée dépasse une valeur spécifique et aucune autre lettre n'est proche d'elle par un seuil, nous imprimons la lettre et l'ajoutons à la chaîne actuelle (dans notre

code, nous avons conservé la valeur à 50 et le seuil de différence à 20).

2. Sinon, nous effaçons le dictionnaire actuel qui a le nombre de détections de symbole présent pour éviter la probabilité qu'une mauvaise lettre soit prédite.

3. Chaque fois que le nombre d'un blanc (fond uni) détecté dépasse une valeur spécifique et si le tampon actuel est vide, aucun espace n'est détecté.

4. Dans un autre cas, il prédit la fin du mot en imprimant un espace et le courant est ajouté à la phrase ci-dessous.

b. Fonction de correction automatique :

Une bibliothèque python **Hunspell_suggest** est utilisée pour suggérer des alternatives correctes pour chaque mot d'entrée (incorrect) et nous affichons un ensemble de mots correspondant au mot actuel dans lequel l'utilisateur peut sélectionner un mot pour l'ajouter à la phrase actuelle. Cela aide à réduire les erreurs commises dans l'orthographe et aide pour prédire des mots complexes.

4.2.2.2 Formation et tests :

Nous convertissons nos images d'entrée (RVB) en niveaux de gris et appliquons un flou gaussien pour supprimer le bruit inutile. Nous appliquons un seuil adaptatif pour extraire notre main de l'arrière-plan et redimensionner nos images à 128 x 128.

Nous alimentons les images d'entrée après prétraitement à notre modèle pour l'entraînement et les tests après avoir appliqué toutes les opérations mentionnées ci-dessus.

La couche de prédiction estime la probabilité que l'image appartienne à l'une des classes. Ainsi, la sortie est normalisée entre 0 et 1 et telle que la somme de chaque valeur de chaque classe est égale à 1. Nous y sommes parvenus en utilisant la fonction softmax.

Au début, la sortie de la couche de prédiction sera quelque peu éloignée de la valeur réelle. Pour l'améliorer, nous avons entraîné les réseaux à l'aide de données étiquetées. L'entropie croisée est une mesure de performance utilisée dans la classification. C'est une fonction continue qui est positive à des



valeurs qui ne sont pas identiques à la valeur étiquetée et qui est zéro exactement lorsqu'elle est égale à la valeur étiquetée. Par conséquent, nous avons optimisé l'entropie croisée en la minimisant au plus près de zéro. Pour ce faire, dans notre couche réseau, nous ajustons les poids de nos réseaux de neurones. TensorFlow a une fonction intégrée pour calculer l'entropie croisée.

Comme nous avons découvert la fonction d'entropie croisée, nous l'avons optimisée à l'aide de Gradient Descent. En fait, le meilleur optimiseur de descente de gradient s'appelle Adam Optimizer.

5. Défis rencontrés

Nous avons été confrontés à de nombreux défis au cours du projet. Le tout premier problème auquel nous avons été confrontés concernait l'ensemble de données.

Nous voulions traiter des images brutes et des images trop carrées comme CNN à Keras car il était beaucoup plus pratique de travailler avec uniquement des images carrées. Nous n'avons pu trouver aucun ensemble de données existant pour cela, nous avons donc décidé de créer notre propre ensemble de données. Le deuxième problème consistait à sélectionner un filtre que nous pouvions appliquer sur nos images afin que les caractéristiques appropriées des images puissent être obtenues et, par conséquent, nous puissions fournir cette image comme entrée pour le modèle CNN. Nous avons essayé divers filtres, notamment le seuil binaire, la détection des contours astucieux, le flou gaussien, etc., mais nous avons finalement opté pour le filtre de flou gaussien.

6. Résultats

Nous avons atteint une précision de 95,8% dans notre modèle en utilisant uniquement la couche 1 de notre algorithme, et en utilisant la combinaison de la couche 1 et de la couche 2, nous obtenons une précision de 98,0%, ce qui est une meilleure précision que la plupart des documents de recherche actuels sur langue des signes américaine. La plupart des articles de recherche se concentrent sur l'utilisation d'appareils tels que kinect pour la détection des mains. Dans [7], ils construisent un système de reconnaissance de la langue des signes flamande utilisant des réseaux de neurones convolutifs et kinect et atteignent un taux d'erreur de 2,5%.



Dans [8], un modèle de reconnaissance est construit en utilisant un classificateur de modèle de Markov caché et un vocabulaire de 30 mots et ils atteignent un taux d'erreur de 10,90 %. Dans [9], ils atteignent une précision moyenne de 86 % pour 41 gestes statiques en langue des signes japonaise. L'utilisation de la carte des capteurs de profondeur [10] a atteint une précision de 99,99 % pour les signataires observés et de 83,58 % et 85,49 % pour les nouveaux signataires. Ils ont également utilisé CNN pour leur système de reconnaissance. Il convient de noter que notre modèle n'utilise aucun algorithme de soustraction d'arrière-plan alors que certains des modèles présents ci-dessus le font. Ainsi, une fois que nous essayons d'implémenter la soustraction de fond dans notre projet, les précisions peuvent varier. D'autre part, la plupart des projets de ce qui précède utilisent des appareils kinect, mais notre objectif principal était de créer un projet pouvant être utilisé avec des ressources facilement disponibles. Un capteur comme kinect non seulement n'est pas facilement disponible, mais il est également coûteux à acheter pour la plupart du public et notre modèle utilise une webcam normale de l'ordinateur portable, c'est donc un avantage considérable. Vous trouverez ci-dessous les matrices de confusion pour nos résultats.



		A	B	C	D	P	r	e	d	i	c	t	e	d		V	a	I	u	e	s					
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	145	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	1	0	0	2	10	0	0	0
C o r r e c t	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1
	I	0	0	0	33	0	0	0	0	108	0	2	0	0	0	0	0	0	0	0	7	1	0	0	0	0
	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
V a l u e s	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
	S	0	0	0	0	1	0	0	0	0	0	0	0	0	1	10	0	0	0	132	0	0	0	0	8	0
	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	151	0	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	115	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 27. Matrice de confusion (Algo 1)

					P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	A	147	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	4	0	0	0	0	0	0	0	0	3	10	0	0	0
C o r r e c t	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	H	1	0	0	0	0	0	7	143	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
	I	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	J	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	K	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	L	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V a l u e s	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0
	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
	S	0	0	0	0	1	0	0	0	0	0	0	0	0	10	0	0	0	133	0	0	0	0	8	0
	T	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0	
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	148	0	
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 28. Matrice de confusion (Algo 1 + Algo 2)

7. Conclusion :

Dans ce rapport, une reconnaissance fonctionnelle de la langue des signes américaine basée sur la vision en temps réel pour les personnes D&M a été développée pour les alphabets ASL. Nous avons atteint une précision finale de 98,0% sur notre ensemble de données. Nous sommes en mesure d'améliorer notre prédiction après avoir implémenté deux couches d'algorithmes dans lesquelles nous vérifions et prédisons des symboles qui se ressemblent davantage.

De cette façon, nous pouvons détecter presque tous les symboles à condition qu'ils soient affichés correctement, qu'il n'y ait pas de bruit de fond et que l'éclairage soit adéquat.

8. Portée future :

Nous prévoyons d'atteindre une plus grande précision même en cas d'arrière-plans complexes en essayant divers algorithmes de soustraction d'arrière-plan. Nous pensons également à améliorer le prétraitement pour prédire les gestes dans des conditions de faible luminosité avec une plus grande précision.

Conclusion

Le développement d'un système de traduction automatique de SL basé sur une machine qui transforme SL en parole et en texte ou vice versa est particulièrement utile pour améliorer l'intercommunication. Les progrès de la reconnaissance des formes offrent la promesse de systèmes de traduction automatique, mais de nombreux problèmes difficiles doivent être résolus avant qu'ils ne deviennent une réalité. Plusieurs aspects liés à la technologie SLR, en particulier SLR qui utilise une approche de capteur de gants, ont été explorés et étudiés. De nombreuses recommandations ont été suggérées par les chercheurs pour résoudre les défis existants et anticipés qui offrent de nombreuses opportunités de recherche dans ce domaine. Nous espérons que les chercheurs continueront à adopter de nouvelles technologies pour établir un système réaliste pouvant aider les personnes ayant des troubles de l'audition et de la parole à améliorer leur capacité à s'intégrer dans la société et à réduire le fossé de la communication. Le principal avantage d'une approche sensorielle est que les gants peuvent acquérir directement des données en termes de valeurs de tension de l'appareil informatique, éliminant ainsi le besoin de traiter les données brutes en données significatives. De plus, cette approche n'est pas soumise aux influences environnementales, par exemple, l'emplacement de l'individu ou les conditions de fond et les effets de lumière ; ainsi, les données générées sont exactes.

Cependant, la reconnaissance des gestes à base de gants nécessite que l'utilisateur porte un gant de données encombrant pour capturer les mouvements de la main et des doigts. Cela nuit à la commodité et au naturel de l'interaction homme-ordinateur. La limitation rencontrée par cette approche est l'incapacité d'obtenir des données significatives complémentaires aux gestes pour donner tout le sens de la conversation, telles que les expressions faciales, les mouvements des yeux et la lecture des lèvres. De plus, des efforts devraient être faits pour améliorer la robustesse du système pour permettre une personnalisation sans effort et étendre les méthodes actuelles à d'autres types d'applications, par exemple, aux interfaces mobiles basées sur les gestes. De plus, les langues des signes ont certaines règles et une certaine grammaire pour la formation de leurs phrases. Ces règles doivent être prises en compte lors de la traduction d'une langue des signes vers une langue parlée. Enfin, il serait utile de développer un système de traduction capable d'interpréter différentes langues des signes.

Les références

- [1] T. Yang, Y. Xu et « A. », Hidden Markov Model for Gesture Recognition », CMU-RI-TR-94 10, Robotics Institute, CarnegieMellon Univ., Pittsburgh, PA, mai 1994.
- [2] Pujan Ziaie, Thomas Müller, Mary Ellen Foster et Alois Knoll "A NaïveBayes Munich, Dépt. of Informatics VI, Robotique et systèmes embarqués, Boltzmannstr. 3, DE-85748 Garching, Allemagne.
- [3] https://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html
- [4] Mohammed Waleed Kalous, Reconnaissance automatique des signes d'Auslan à l'aide de PowerGloves : Vers une reconnaissance à grand lexique de la langue des signes.
- [5] aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2
- [6] <http://www-i6.informatik.rwth-aachen.de/~dreuw/database.php>
- [7] Pigou L., Dieleman S., Kindermans P.J., Schrauwen B. (2015) Reconnaissance de la langue des signes à l'aide de réseaux de neurones convolutifs. Dans : Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Notes de cours en informatique, vol 8925. Springer, Cham
- [8] Zaki, MM, Shaheen, SI : reconnaissance de la langue des signes à l'aide d'une combinaison de nouvelles fonctionnalités basées sur la vision. Lettres de reconnaissance de modèle 32(4), 572-577 (2011)
- [9] N. Mukai, N. Harada et Y. Chang, « Reconnaissance de l'orthographe japonaise basée sur l'arbre de classification et l'apprentissage automatique » 2017 *Nicograph International (NicoInt)*, Kyoto, Japon, 2017, p. 19-24. doi:10.1109/NICOInt.2017.9
- [10] Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen "Reconnaissance de l'orthographe en langage des signes en temps réel à l'aide de réseaux de neurones convolutifs à partir d'une carte de profondeur" 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)

APPENDICE

OpenCV

OpenCV (Open Source Computer Vision Library) est publié sous une licence BSD et est donc gratuit pour un usage académique et commercial. Il possède des interfaces C++, Python et Java et prend en charge Windows, Linux, Mac OS, iOS et Android. OpenCV a été conçu pour l'efficacité de calcul et avec un fort accent sur les applications en temps réel. Écrit en C/C++ optimisé, la bibliothèque peut tirer parti du traitement multicœur. Activé avec OpenCL, il peut tirer parti de l'accélération matérielle de la plate-forme de calcul hétérogène sous-jacente.

Adopté dans le monde entier, OpenCV compte plus de 47 000 utilisateurs et un nombre estimé de téléchargements dépassant les 14 millions. L'utilisation va de l'art interactif à l'inspection des mines, en passant par l'assemblage de cartes sur le Web ou par la robotique avancée.

Réseau de neurones à convolution

Les CNN utilisent une variante de perceptrons multicouches conçue pour nécessiter un prétraitement minimal. Ils sont également connus sous le nom de réseaux de neurones artificiels invariants par décalage ou invariants dans l'espace (SIANN), en raison de leur architecture à poids partagés et de leurs caractéristiques d'invariance de traduction.

Les réseaux convolutifs ont été inspirés par des processus biologiques dans la mesure où le modèle de connectivité entre les neurones ressemble à l'organisation du cortex visuel animal. Les neurones corticaux individuels ne répondent aux stimuli que dans une région restreinte du champ visuel connue sous le nom de champ récepteur. Les champs récepteurs des différents neurones se chevauchent partiellement de sorte qu'ils couvrent tout le champ visuel.

Les CNN utilisent relativement peu de prétraitement par rapport aux autres algorithmes de classification d'images. Cela signifie que le réseau apprend les filtres qui, dans les algorithmes traditionnels, ont été conçus à la main. Cette indépendance par rapport aux connaissances préalables et à l'effort humain dans la conception des fonctionnalités est un avantage majeur.

Ils ont des applications dans la reconnaissance d'images et de vidéos, les systèmes de recommandation, la classification d'images, l'analyse d'images médicales et le traitement du langage naturel.

Tensorflow

TensorFlow est une bibliothèque logicielle open source pour la programmation de flux de données dans une gamme de tâches. Il s'agit d'une bibliothèque mathématique symbolique, également utilisée pour les applications d'apprentissage automatique telles que les réseaux de neurones. Il est utilisé à la fois pour la recherche et la production chez Google.

TensorFlow a été développé par l'équipe Google Brain pour une utilisation interne à Google. Il a été publié sous la bibliothèque open source Apache 2.0 en novembre 9, 2015.

TensorFlow est le système de deuxième génération de Google Brain. La version 1.0.0 a été publiée le 11 février 2017. Alors que l'implémentation de référence s'exécute sur un seul appareil, TensorFlow peut s'exécuter sur plusieurs processeurs et GPU (avec des extensions CUDA et SYCL en option pour le calcul à usage général sur des unités de traitement graphique). TensorFlow est disponible sur Linux 64 bits, macOS, Windows et les plates- formes informatiques mobiles, y compris Android et iOS.

Son architecture flexible permet un déploiement facile du calcul sur une variété de plates- formes (CPU, GPU, TPU), et des ordinateurs de bureau aux clusters de serveurs en passant par les appareils mobiles et périphériques.