REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Kheider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : SIOD_Startup_04/M2/2024

Mémoire

pour l'obtention du diplôme Master académique en Informatique

Spécialité : Système d'Information Optimisation et Décision (SIOD)

Appareil intelligent pour les étudiants aveugles : Cas d'examen

---

Présenté Par :

Mammeri oumaima

Meghazi larafi aya

Soutenu le 23 juin 2024, devant le jury composé de :

Guerrouf Fayçal	MCB	Président
Zerarka Nourelhouda	MCB	Rapporteur
Moussaoui Manel	MAA	Examinateur

Année Universitaire 2023/2024

❖Remerciments ❖

Avant tout, nous tenons à remercier Allah Tout-Puissant, qui nous a donné la force et la patience pour accomplir ce travail. Sa grâce et Sa bienveillance ont été les raisons principales de l'achèvement de cette recherche.

Nous exprimons notre profonde gratitude à notre encadrante, Dr Zerarka Nourelhouda, pour ses précieux conseils et son soutien constant tout au long du projet. Elle a été pour nous une guide et un soutien, ne ménageant ni son temps ni ses connaissances. Merci du fond du cœur.

Nous remercions également tous les membres du jury pour avoir bien voulu évaluer notre travail. Nous apprécions énormément leurs efforts et le temps précieux consacré à la révision et à l'évaluation de cette recherche.

Nos remerciements sincères vont également à tous les enseignants du département d'informatique qui ont contribué à notre formation tout au long de ces années. Ils nous ont généreusement transmis leur savoir et leur expertise, et ont toujours été présents pour nous soutenir dans notre parcours académique.

Nous remercions chaleureusement nos parents, qui ont toujours été notre pilier et notre soutien durant toute la durée de cette recherche. Ils ont été à nos côtés, offrant un soutien moral et matériel, et nous encourageant à persévérer et à nous dépasser. Merci pour tout.

Enfin, craignant d'avoir oublié quelqu'un, nous adressons nos plus sincères remerciements et notre profonde gratitude à tous ceux et celles à qui nous sommes redevables. Nous demandons à Dieu de les récompenser de la meilleure manière et de leur accorder santé et bonheur. Merci à vous tous du fond du cœur.

❖Dedication ❖

À Allah avant tout, louange à Toi comme il se doit pour la majesté de Ton visage et la grandeur de Ton pouvoir.

Le chemin n'a pas été court et ne devait pas l'être, et celui qui persévère y parvient malgré tout.

À mon cher père, mon premier enseignant, ma fierté et mon honneur dans cette vie.

À ma chère mère, qui a toujours été ma première source de soutien et d'encouragement dans la vie, que Dieu te protège.

À mes frères Mohamed Amine, Moatassem Billah, Yahia et Firas, pour leur loyauté et leur fraternité sans faille.

À mon binôme Oumeima, pour son dévouement et son travail acharné dans la réussite de ce projet, qui a toujours été comme une sœur pour moi .

À notre encadrante, pour ses efforts inlassables, ses précieux conseils et son soutien constant du début à la fin du projet.

À mes tantes, mes oncles et tous mes amis chers, ainsi qu'à tous ceux qui ont contribué à mon succès.

Que cette thèse soit un témoignage de ma profonde gratitude envers chacun de vous. Vous avez tous joué un rôle essentiel dans cette réalisation, et je vous en suis infiniment reconnaissant.

Aya

❖Dedication ❖

avec tous mes sentiments de respect , avec l'experience de ma reconnaissance, je dédie ma remise de diplome et ma joie.

À ma mère, a mon paradis , à la prunelle de mes yeux ,à la source de ma joie et mon bonheur, ma lune et le fil d'espoir qui allumer mon chemin

À mon père, pour son soutien indéfectible, ses encouragements constants, et sa foi en mes capacités.

À mes frères Amine et Ilyes pour leur soutien et leurs mots d'encouragement.

À ma tante maternelle, pour sa présence chaleureuse, ses conseils précieux, et son soutien infaillible.

À l'encadrante dr zerarka norelhouda , pour ses conseils avisés, sa patience, et son accompagnement précieux tout au long de ce projet.

À mon binôme Aya, pour sa camaraderie,son soutien indéfectible et son esprit de collaboration.

À mes meilleurs amis, pour leur amitié sincère, leur soutien constant.

Je consacre ce travail à vous tous, en signe de ma plus sincère reconnaissance pour votre dévouement sans faille, votre amour et votre soutien constant. Merci d'être des piliers essentiels dans ma vie et d'embellir mon parcours par votre présence.

Oumaima 

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

IoT : Internet des Objets

OCR : Reconnaissance Optique de Caractères

gTTS : Google Text-to-Speech

NLP : Traitement du Langage Naturel (Natural Language Processing)

LSTM : Long Short-Term Memory (Mémoire à Long Court Terme)

CNN : Convolutional Neural Network (Réseau de Neurones Convolutifs)

RNN : Recurrent Neural Network (Réseau de Neurones Récurrents)

GPIO : General-Purpose Input/Output (Entrée/Sortie à Usage Général)

MFCC : Mel-Frequency Cepstral Coefficients (Coefficients Cepstraux en Fréquence Mel)

RPI : Raspberry Pi (Micro-ordinateur à Carte Unique)

Résumé

Les étudiants aveugles font face à des défis majeurs, surtout durant les examens, où les méthodes traditionnelles d'assistance humaine révèlent souvent leurs limites en termes d'accessibilité et d'équité. Pour répondre à ces enjeux, nous avons conçu un appareil intelligent intégrant des technologies de reconnaissance optique de caractères et de synthèse vocale , ainsi que des modèles d'apprentissage profond . Cet appareil permet aux étudiants aveugles de passer leurs examens de manière autonome, supprimant ainsi le besoin d'un assistant humain pour lire les questions et écrire les réponses. Nos recherches ont requis l'utilisation de l'Internet des objets pour une communication en temps réel efficace et l'intégration de techniques avancées d'intelligence artificielle afin de garantir une reconnaissance vocale et textuelle de haute précision. La conception de l'appareil a également pris en compte l'ergonomie et l'accessibilité, assurant une utilisation facile pour les étudiants aveugles tout en garantissant la sécurité et la confidentialité des données personnelles. En améliorant continuellement la précision de nos modèles et en élargissant le support linguistique et fonctionnel, notre travail pave la voie à de futures innovations dans le domaine des technologies assistives. Cela contribue ainsi à une meilleure inclusion et à une égalité des chances accrue pour les personnes aveugles dans le système éducatif et au-delà.

Mots clés :Étudiants Aveugles,Intelligence Artificielle, Internet des Objets, Apprentissage Profond, Reconnaissance Optique de Caractères, Google Text-to-Speech.

Abstract

Blind students face major challenges, especially during exams, where traditional human assistance methods often reveal their limitations in terms of accessibility and equity. To address these issues, we have designed an intelligent device integrating Optical Character Recognition and Google Text-to-Speech technologies, as well as deep learning models . This device enables blind students to take their exams independently, eliminating the need for a human assistant to read questions and write answers. Our research required the use of the Internet of Things for effective real-time communication and the integration of advanced Artificial Intelligence techniques to ensure high precision in voice and text recognition. The device's design also took into account ergonomics and accessibility, ensuring easy use for blind students while guaranteeing the security and confidentiality of personal data. By continuously improving the accuracy of our models and expanding linguistic and functional support, our work paves the way for future innovations in assistive technologies. This contributes to better inclusion and increased equality of opportunity for blind people in the education system and beyond.

Keywords: Blind Students, Artificial Intelligence, Internet of Things, Deep Learning, Optical Character Recognition, Google Text-to-Speech.

ملخص

يواجه الطلاب المكفوفون تحديات كبيرة،خاصة اثناء الامتحانات، حيث تكشف الطرق التقليدية للمساعدة البشرية غالباًا عن حدودها من حيث إمكانية الوصول والعدالة. للتعامل مع هذه القضايا، قمنا بتصميم جهاز ذكي يدمج تقنيات التعرف الضوئي على الحروف والنص إلى كلام من جوجل ، بالإضافة إلى نماذج التعلم العميق . يمكن لهذا الجهاز أن يمكن الطلاب المكفوفين من إجراء امتحاناتهم بشكل مستقل، مما يلغي الحاجة إلى مساعد بشري لقراءة الأسئلة وكتابة الإجابات. تطلبت أبحاثنا استخدام إنترنت الأشياء للاتصال الفعال في الوقت الفعلي ودمج تقنيات الذكاء الاصطناعي المتقدمة لضمان دقة عالية في التعرف الصوتي والنصي. كما أخذ تصميم الجهاز في الاعتبار سهولة الاستخدام وإمكانية الوصول، مما يضمن سهولة الاستخدام للطلاب المكفوفين مع ضمان أمان وسرية البيانات الشخصية. من خلال تحسين دقة نماذجنا باستمرار وتوسيع الدعم اللغوي والوظيفي، يمهد عملنا الطريق للابتكارات المستقبلية في مجال التقنيات المساعدة. يساهم هذا في تحسين الشمولية وزيادة تكافؤ الفرص للأشخاص المكفوفين في النظام التعليمي وما بعده.

الكلمات المفتاحية:الطلاب المكفوفين،الذكاء الاصطناعي ،إنترنت الأشياء، التعلم العميق ،التعرف الضوئي على الحروف، تحويل النص إلى كلام

Introduction

La vision joue un rôle crucial dans la perception de l'environnement, ce qui facilite la navigation, la communication et l'acquisition de connaissances. Ceux qui en profitent ont la possibilité de réaliser des tâches quotidiennes et d'accéder rapidement à de nombreuses informations visuelles, ce qui joue un rôle essentiel dans leur développement personnel et d'emploi.

Les individus aveugles font face à des difficultés majeures, restreignant l'accès à l'information visuelle et rendant les tâches quotidiennes difficiles. Ces obstacles sont surmontés par des technologies assistives telles que les lecteurs d'écran et les dispositifs braille, mais elles rencontrent toujours des problèmes de mobilité, d'éducation, d'emploi et de participation sociale. Les élèves aveugles sont confrontés à des difficultés académiques encore plus importantes. Les formats inaccessibles empêchent souvent l'accès aux ressources éducatives. Pendant les examens, ils ont souvent recours à des assistants humains pour lire les questions et rédiger les réponses, ce qui restreint leur indépendance et peut avoir un impact sur leurs résultats scolaires.[24]

Les difficultés particulières auxquelles font face les étudiants aveugles lors des examens sont au cœur de cette étude. Les techniques actuelles, basées sur l'utilisation d'assistants humains pour lire les questions et transcrire les réponses, ne sont pas toujours exactes et peuvent engendrer des disparités et des erreurs. En outre, l'inégalité d'accès aux ressources et aux technologies assistives renforce les inégalités académiques entre les étudiants aveugles et leurs camarades voyants. Les questions essentielles sur l'équité et l'accessibilité dans le système éducatif sont soulevées par cette situation.

L'objectif de ce travail consiste à concevoir un dispositif intelligent qui exploite des technologies de reconnaissance optique de caractères (OCR) et de synthèse vocale (gTTS),

ainsi que des modèles d'apprentissage profond, afin d'aider les étudiants aveugles lors des examens. Les étudiants peuvent utiliser cet appareil pour capturer des photos des questions d'examen, les convertir en audio et saisir leurs réponses par la voix sans avoir besoin de l'aide d'un enseignant ou d'un assistant humain. En automatisant cette procédure, notre objectif est de proposer une solution plus juste et autonome aux étudiants aveugles, ce qui leur permettra d'améliorer leur expérience académique et leurs performances sans qu'ils ne se sentent inférieurs.

Ce mémoire est structuré en quatre chapitres principaux :

1. **Chapitre 1 : Concepts de base :**Ce chapitre introduit les défis quotidiens et académiques des personnes aveugles et explore comment l'intelligence artificielle (IA) peut offrir des solutions innovantes. Il présente également les concepts d'apprentissage profond et d'apprentissage automatique appliqués pour répondre aux besoins des personnes aveugles.

2. **Chapitre 2 : Intégration de NLP dans l'IoT :** Ce chapitre examine les techniques de traitement du langage naturel (NLP) et les technologies de reconnaissance de texte et vocale. Il explore les méthodes modernes et leur intégration dans les systèmes actuels, ainsi que les applications de l'Internet des objets (IoT) pour les personnes aveugles.

3. **Chapitre 3 : Conception et réalisation :** Ce chapitre décrit la contribution spécifique de notre travail, en détaillant la partie matérielle, incluant les composants et la conception 3D, et la partie logicielle, avec l'implémentation des modèles d'apprentissage profond pour la reconnaissance vocale et OCR, ainsi que les optimisations apportées.

4. **Chapitre 4 : Implémentation et Résultats Obtenus :** Ce dernier chapitre présente l'implémentation pratique du système, incluant la mise en place et l'intégration des composants matériels et logiciels. Il évalue les performances des modèles, analyse les données recueillies, et propose un tableau de comparaison des différentes approches utilisées, tout en discutant les retours des utilisateurs.

Chapitre 1

Concepts de base

Introduction

Dans ce chapitre, nous nous pencherons sur la vie des personnes aveugles, en examinant les différents types de cécité et les nombreux défis qu'elles rencontrent au quotidien. Nous explorerons comment l'intelligence artificielle (IA) a émergé comme une technologie révolutionnaire, offrant des solutions innovantes pour améliorer leur qualité de vie. Nous discuterons des réussites significatives de l'IA pour les personnes aveugles, en mettant l'accent sur l'apprentissage automatique et l'apprentissage profond (deep learning). Ces technologies ont permis de développer des outils et des applications qui facilitent la navigation, la communication et l'accès à l'information pour les personnes aveugles, ouvrant ainsi de nouvelles perspectives pour leur autonomie et inclusion .

1.1 Les personnes aveugles

Les aveugles sont définis comme des personnes codées "aveugles complets", "totalement aveugles", "partiellement aveugles" ou "malvoyants". La plupart ont déclaré être totalement ou partiellement aveugles, c'est-à-dire ayant une vision résiduelle limitée à la distinction de silhouettes." Cette définition précise que les personnes aveugles sont catégorisées selon différents degrés de cécité :

- "Aveugles complets" = cécité totale
- "Totalement aveugles" = cécité totale

- "Partiellement aveugles" = vision très limitée
- "Malvoyants" = vision réduite mais supérieure à la cécité partielle

Et que la majorité de ces personnes souffrent soit d'une cécité totale, soit d'une cécité partielle ne leur permettant que de distinguer vaguement les formes/silhouettes [25].

1.2 Types de Cécité

Dire qu'une personne est aveugle ne signifie pas qu'elle ne peut rien voir. La cécité a différents types :

- **Cécité totale :** La cécité totale, qui prive le sujet de toute perception lumineuse et de toute information visuelle. Donc une cécité totale est définie comme une absence complète de perception visuelle, que ce soit de la lumière, des formes, des couleurs ou toute autre information visuelle. C'est la forme la plus sévère de déficience visuelle où la personne n'a absolument aucune vision résiduelle [26].
- **Cécité légale :** La cécité légale est définie comme une acuité visuelle corrigée inférieure à 20/200 dans le meilleur œil, ou un champ visuel réduit à un angle de 20 degrés ou moins. Cela signifie que même avec des lunettes, des lentilles de contact ou une intervention chirurgicale, une personne atteinte de cécité légale ne peut voir à 20 pieds (environ 6 mètres) ce qu'une personne ayant une vision normale peut voir à 200 pieds (environ 61 mètres). Bien que les personnes atteintes de cécité légale aient une vision résiduelle, celle-ci est extrêmement limitée et peut considérablement entraver leurs activités quotidiennes et leur mobilité sans aides appropriées. La cécité légale peut résulter de diverses conditions, telles que la dégénérescence maculaire liée à l'âge, le glaucome, le diabète ou les lésions oculaires [27] .
- **Cécité congénitale :** La cécité congénitale est une perte totale ou quasi-totale de la vision existant dès la naissance ou apparaissant peu après. Elle peut être causée par diverses anomalies génétiques héréditaires ou être la conséquence de complications pré ou périnatales comme une prématurité, un traumatisme obstétrical, une infection congénitale, etc. La cécité est dite congénitale lorsqu'elle est présente avant l'âge de 2-3 ans, avant le développement complet des voies visuelles. Les causes les

plus fréquentes sont les malformations oculaires, les lésions du nerf optique, les anomalies cérébrales ou les pathologies rétiniennes d'origine génétique. La cécité congénitale entraîne des déficits majeurs dans l'acquisition des capacités visuelles et visuospatiales qui doivent être compensés précocement [28].

- **Cécité acquise :** La cécité acquise survient après la naissance, résultant de diverses causes telles que les traumatismes, les maladies oculaires dégénératives, les maladies systémiques affectant la vision ou le vieillissement normal. Les personnes atteintes de cécité acquise ont déjà bénéficié d'une expérience visuelle antérieure, ce qui peut faciliter leur adaptation aux défis de la cécité. Cependant, la perte de la vision peut également entraîner des difficultés émotionnelles et des défis d'adaptation à un mode de vie non visuel. Le degré d'impact dépend de l'âge auquel la cécité est survenue et de la rapidité de la perte de vision. Une rééducation et des aides techniques appropriées sont essentielles pour permettre à ces personnes de maintenir leur indépendance et leur qualité de vie [29].

- **Cécité corticale :** La cécité corticale est une perte de sensation visuelle liée à des lésions atteignant les voies optiques en arrière du corps genouillé latéral et plus particulièrement les cortex visuels primaires." Donc la cécité corticale est spécifiquement une perte de la vision causée par des lésions ou dommages au niveau des aires visuelles du cortex cérébral, en particulier le cortex visuel primaire. Contrairement à d'autres types de cécité, les voies optiques jusqu'au corps genouillé latéral (avant le cortex) sont intactes, mais c'est le traitement de l'information visuelle au niveau cortical qui est atteint [30].

- **Cécité nocturne (Nyctalopie) :** La cécité nocturne congénitale est un trouble visuel héréditaire caractérisé par une dysfonction spécifique de la vision nocturne (scotopique). Elle est due à un défaut de transmission du signal visuel entre les photorécepteurs (cônes et bâtonnets) et les cellules bipolaires de la rétine, qui sont respectivement les premier et deuxième neurones impliqués dans le traitement de l'information visuelle. Ce déficit synaptique rétinien, présent dès la naissance, altère la capacité à voir dans des conditions de faible luminosité sans affecter la vision de jour [28].

1.3 Le monde des personnes aveugles

Le monde des personnes aveugles est un univers où les sens autres que la vision prennent toute leur importance. Sans les repères visuels, c'est l'ouïe, le toucher, l'odorat et les perceptions corporelles qui guident leur appréhension de l'environnement. Chaque mouvement, chaque déplacement devient un défi de mémorisation des textures, des sons et des odeurs qui jalonnent leur quotidien. Se nourrir, s'habiller, se déplacer nécessitent l'acquisition de techniques particulières qui mobilisent pleinement les sens restants. Si la perte de la vue peut sembler un handicap, elle développe également des facultés sensorielles et cognitives remarquables. Une acuité auditive, un sens du toucher et une mémoire spatiale hors normes permettent aux personnes aveugles de se représenter leur environnement et de s'y mouvoir avec une étonnante aisance. Leur monde, loin d'être un monde de ténèbres, est riche en paysages sonores, en reliefs tactiles et en effluves riches en informations [31].

1.4 Les défis de la vie des personnes aveugles

Les personnes aveugles sont confrontées à de nombreux défis dans leur vie quotidienne sous de nombreux aspects, notamment :

- **Défis environnementaux :** Les personnes aveugles ou malvoyantes font face à de nombreux défis lorsqu'elles se déplacent dans un environnement urbain. Les obstacles physiques comme les trottoirs encombrés, le bruit de la circulation et les odeurs fortes peuvent rendre la navigation difficile. De plus, le manque d'accès à l'information visuelle sur la signalisation et les repères spatiaux complique l'orientation. L'aménagement urbain inadapté crée des barrières importantes pour leur mobilité et leur sécurité, soulignant la nécessité d'une conception universelle prenant en compte leurs besoins spécifiques. Relever ces défis environnementaux est essentiel pour favoriser une meilleure inclusion de cette population [32].

- **Défis sociaux :** Les personnes aveugles ou malvoyantes doivent également faire face à de nombreux défis sociaux lorsqu'elles se déplacent en ville. Elles peuvent rencontrer des attitudes négatives, des préjugés ou un manque de sensibilisation de la part du grand public, ce qui peut entraîner de la discrimination ou de l'exclusion sociale. De plus, l'accès limité à l'information et aux services peut restreindre leur participa-

tion à la vie communautaire. Les barrières de communication avec les autres citadins constituent également un défi majeur. Surmonter ces obstacles sociaux est crucial pour promouvoir l'inclusion et l'égalité des chances, en sensibilisant davantage la population aux besoins spécifiques des personnes atteintes de déficience visuelle en milieu urbain [32].

- **Défis techniques :** Les personnes aveugles ou malvoyantes font face à des défis techniques importants dans les villes. L'accès limité aux technologies d'aide à la navigation et à l'orientation entrave leur mobilité. Les infrastructures urbaines intelligentes ne sont souvent pas compatibles avec leurs dispositifs d'assistance. Le manque de normes d'accessibilité universelle dans la conception urbaine pose également problème. Relever ces défis techniques est essentiel pour permettre leur autonomie en milieu urbain [32].

- **Défis psychologiques :** Au-delà des aspects environnementaux, sociaux et techniques, les personnes aveugles ou malvoyantes doivent également faire face à des défis psychologiques importants lors de leurs déplacements en ville. Le stress, l'anxiété et le sentiment d'insécurité peuvent être omniprésents, entraînant une perte de confiance et d'autonomie. La peur constante des obstacles, du bruit et des imprévus peut avoir un impact négatif sur leur bien-être mental et émotionnel. De plus, le sentiment d'exclusion sociale et les attitudes discriminatoires auxquelles elles sont confrontées peuvent affecter leur estime de soi. Relever ces défis psychologiques est crucial pour préserver leur santé mentale et leur permettre de se déplacer sereinement dans les espaces urbains [32].

- **Défis financiers :** Les personnes aveugles ou malvoyantes rencontrent des défis financiers majeurs en milieu urbain. L'acquisition de technologies d'assistance et l'accès à des services adaptés représentent des coûts importants. Les aménagements de leur environnement pour plus d'accessibilité nécessitent également des investissements conséquents. Ces contraintes financières limitent considérablement leur autonomie et leur mobilité en ville [32].

1.5 L'intelligence artificielle

L'intelligence artificielle (IA) fait référence à la capacité des systèmes informatiques à simuler l'intelligence humaine, telles que l'apprentissage, le raisonnement, la planification et la résolution de problèmes. L'IA moderne utilise des techniques avancées comme l'apprentissage automatique et les réseaux de neurones artificiels pour analyser d'immenses quantités de données et détecter des modèles complexes. Bien que l'IA soulève des interrogations éthiques et soulève des inquiétudes quant à son impact sur l'emploi, elle offre également d'incroyables opportunités dans de nombreux domaines. En médecine, l'IA aide au diagnostic précoce des maladies et permet de personnaliser les traitements. Dans les transports, les voitures autonomes utilisant l'IA promettent d'améliorer la sécurité routière. L'IA révolutionne également les assistants vocaux, la traduction automatique, la reconnaissance faciale et bien d'autres applications. Malgré ses progrès fulgurants, l'IA reste limitée dans certaines tâches cognitives complexes que les humains maîtrisent naturellement [33] [34].

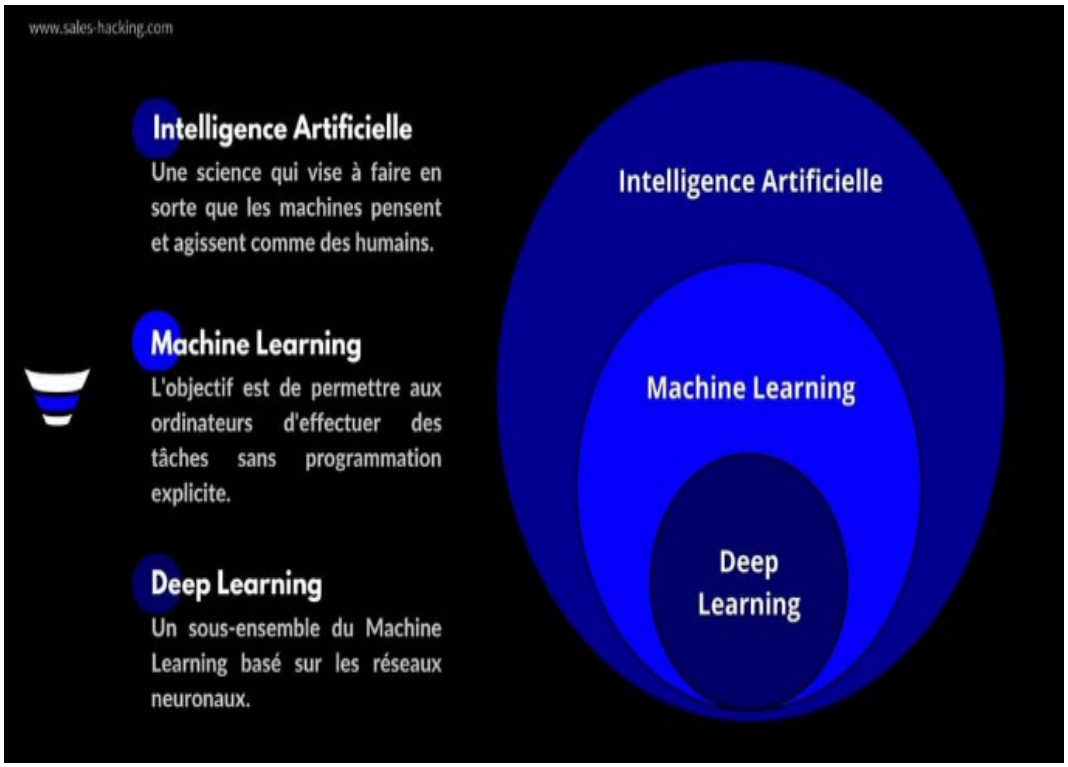


Figure 1.1 : Intelligence Artificielle vs. Machine Learning vs. Deep Learning.[1]

1.5.1 L'intelligence artificielle et les aveugles

L'intelligence artificielle ouvre de nouvelles perspectives prometteuses pour les personnes aveugles ou malvoyantes. Les progrès récents dans les domaines de la reconnaissance vocale, de la vision par ordinateur et du traitement du langage naturel permettent de développer des outils technologiques innovants visant à faciliter leur autonomie et leur inclusion. Les assistants vocaux comme Alexa, Siri ou Google Assistant deviennent des aides précieuses pour accomplir des tâches du quotidien, de la consultation météorologique à l'achat de billets, en passant par le contrôle de la domotique. Parallèlement, des applications comme Microsoft Seeing AI ou les lunettes intelligentes OrCam My Eye exploitent les capacités de reconnaissance visuelle de l'IA pour décrire leur environnement aux déficients visuels. Si des défis techniques, financiers et d'accessibilité subsistent, l'intelligence artificielle semble promise à révolutionner la vie de cette population en leur offrant une autonomie et une indépendance accrues. [35].

1.5.2 Les principaux appareils technologiques d'assistance pour les personnes aveugles

Pour les personnes aveugles ou malvoyantes, de nombreux dispositifs d'assistance technologiques existent afin de faciliter leur quotidien et leur autonomie. On distingue d'un côté les technologies générales comme les ordinateurs, smartphones ou GPS, et de l'autre les technologies d'assistance spécifiquement conçues pour eux. Dans cette dernière catégorie, on trouve les lecteurs d'écran et les loupes d'écran pour utiliser un ordinateur, les loupes vidéo et aides à la lecture/écriture, mais aussi des aides plus simples comme les montres et imprimantes braille. Si ces outils ne remplacent pas la canne blanche, ils permettent une meilleure accessibilité de l'information et un gain d'autonomie précieux au quotidien. Bien choisir et savoir utiliser ces dispositifs high-tech est donc essentiel pour les personnes déficientes visuelles [36].

1.6 L'apprentissage automatique

L'apprentissage automatique est une branche de l'intelligence artificielle qui permet aux systèmes informatiques d'apprendre et de s'améliorer de façon autonome à partir de

données, sans être explicitement programmés. Au lieu de suivre des instructions codées d'avance, les algorithmes d'apprentissage automatique détectent eux-mêmes les motifs et les tendances dans les données d'entraînement pour construire des modèles prédictifs ou de classification. Ils sont ensuite capables de généraliser ces modèles à de nouvelles données jamais rencontrées auparavant.

Les principales techniques d'apprentissage automatique comprennent l'apprentissage supervisé (à partir de données étiquetées), l'apprentissage non supervisé (à partir de données non étiquetées) et l'apprentissage par renforcement basé sur la récompense. L'apprentissage profond ou deep learning, qui utilise des réseaux de neurones artificiels multicouches, est particulièrement efficace pour les tâches complexes comme la reconnaissance d'images ou de la parole. L'apprentissage automatique est aujourd'hui omniprésent, des moteurs de recommandation au diagnostic médical en passant par la conduite autonome et la détection de fraudes [37, 38].

1.6.1 Les neurones

Les neurones sont des cellules excitables électriquement qui transmettent les signaux dans l'organisme. Chaque neurone est constitué d'un corps cellulaire, de dendrites qui reçoivent les signaux entrants, et d'un axone qui propage les signaux sortants. Les neurones utilisent des mécanismes électriques et chimiques pour transmettre l'information à d'autres neurones ou à des cellules effectrices comme les muscles. Ils sont les unités fondamentales du système nerveux, permettant les fonctions sensorielles, motrices et cognitives [39].

1.6.2 La définition d'un neurone artificiel

Un neurone artificiel est l'unité de base d'un réseau de neurones artificiels. Il reçoit plusieurs entrées pondérées, en fait la somme, puis applique une fonction d'activation à cette somme pour produire une sortie qui sera transmise ou non aux neurones suivants. Son fonctionnement modélise de façon simplifiée celui d'un neurone biologique [40].

1.6.3 Les réseaux de neurones

Un neurone est avant tout un opérateur mathématique, dont la valeur numérique se calcule par quelques lignes d'un programme. Un neurone réalise une somme pondérée suivie d'une fonction non linéaire f. Cette fonction f doit être bornée, continue et dérivable. Elle peut aussi avoir la forme d'une fonction seuil si le résultat recherché est de type booléen (soit 0 ou 1). Les fonctions les plus fréquemment utilisées sont les fonctions sigmoïdes." Un neurone artificiel dans un réseau de neurones effectue une somme pondérée de ses entrées, puis applique une fonction non-linéaire bornée, continue et dérivable (typiquement une fonction sigmoïde) au résultat de cette somme pour produire sa sortie. C'est un opérateur mathématique qui calcule une fonction non-linéaire de ses entrées pondérées [41].

1.6.4 Fonctionnement des réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des modèles d'apprentissage machine composés de multiples couches de neurones artificiels connectés, qui apprennent à modéliser des relations complexes entre les entrées et les sorties à travers : [38, 42] :

- Architecture du réseau : Un réseau de neurones est constitué de multiples neurones artificiels organisés en couches successives : une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Chaque neurone d'une couche est connecté aux neurones de la couche suivante.

- Propagation avant (forward propagation) : Lors de l'inférence, les données d'entrée sont propagées à travers le réseau, des couches d'entrée vers les couches de sortie. Chaque neurone reçoit les sorties pondérées des neurones de la couche précédente, applique sa fonction d'activation et transmet son résultat à la couche suivante.

- Fonctions d'activation : Les fonctions d'activation non linéaires (sigmoid, ReLU, etc.) introduisent de la non-linéarité, permettant au réseau de modéliser des relations complexes entre les entrées et les sorties.

- Rétropropagation du gradient (backpropagation) : Pendant l'entraînement supervisé, l'erreur entre la sortie prédite et la sortie désirée est calculée. Cette erreur se propage ensuite en sens inverse à travers le réseau pour ajuster les poids des connexions selon un algorithme d'optimisation (descente de gradient).

- Apprentissage : Itérativement, les poids sont mis à jour pour minimiser l'erreur globale du réseau sur l'ensemble des données d'entraînement. Le réseau "apprend" ainsi à modéliser la relation entrée-sortie souhaitée.

- Couches spécialisées : Différents types de couches (convolutionnelles, récurrentes, pooling, etc.) permettent d'exploiter la structure des données (images, séquences, etc.) pour des tâches spécifiques.

- Généralisation : Un bon réseau doit à la fois bien modéliser les données d'entraînement, mais aussi généraliser à de nouvelles données inconnues lors de l'inférence. Les réseaux de neurones profonds avec de multiples couches cachées sont particulièrement puissants pour l'apprentissage automatique de caractéristiques à partir de données brutes complexes.

1.7 Les techniques d'apprentissage automatique

Les technologies et techniques d'apprentissage automatique offrent une variété d'approches puissantes comme :

1.7.1 Algorithmes d'apprentissage supervisé :

Les algorithmes d'apprentissage supervisé incluent la régression logistique pour la classification binaire, les machines à vecteurs de support (SVM) pour la classification, ainsi que les arbres de décision et les forêts aléatoires pour la classification et la régression.[43] La figure 1.2 décrit le fonctionnement de l'algorithme d'apprentissage supervisé.

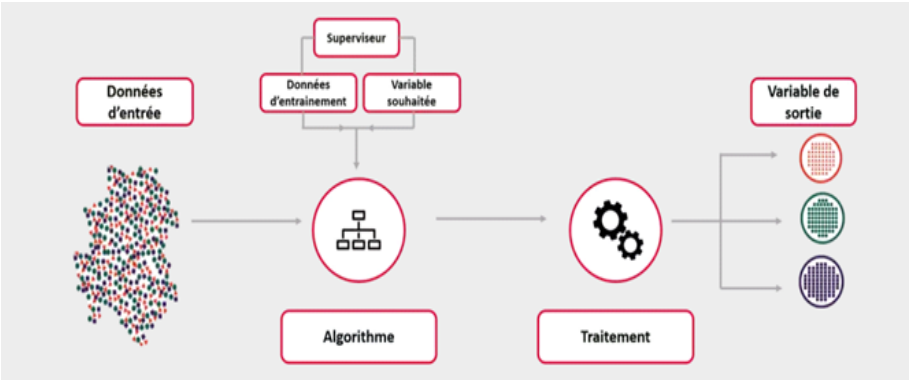


Figure 1.2 : L'apprentissage supervisé [2]

1.7.2 Algorithmes d'apprentissage non supervisé :

Les techniques d'apprentissage non supervisé comprennent le clustering par k-moyennes et k-médoïdes pour la segmentation, la réduction de dimension par ACP (Analyse en Composantes Principales), et les modèles de mélange gaussien pour la modélisation de densité.[3] Le fonctionnement de l'algorithme d'apprentissage non supervisé est illustré dans la figure 1.3.

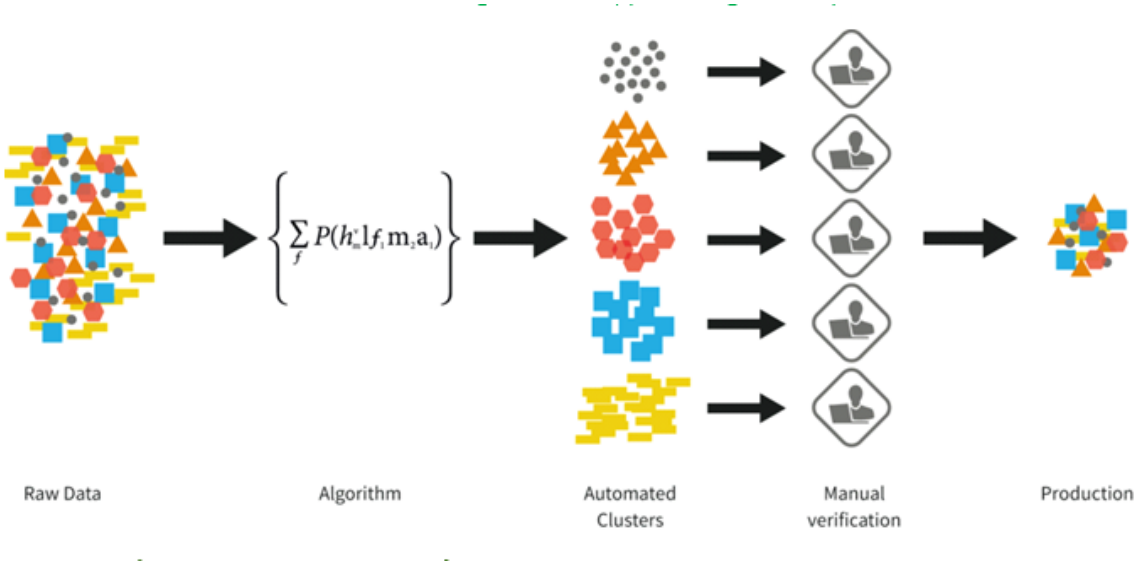


Figure 1.3 : L'apprentissage non supervisé [3]

1.7.3 Apprentissage par renforcement :

Les techniques d'apprentissage par renforcement incluent le Q-learning et les Deep Q-Networks (DQN) pour l'optimisation séquentielle, ainsi que les algorithmes de policy gradient pour l'apprentissage d'agents.[44] la figure 1.4 montre comment fonctionne un algorithme d'apprentissage par renforcement.

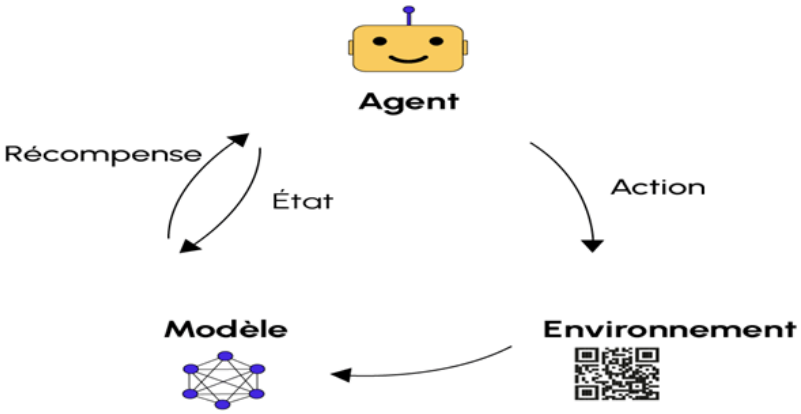


Figure 1.4 : Apprentissage par renforcement [4]

1.7.4 Apprentissage semi-supervisé :

L'apprentissage semi-supervisé combine des données étiquetées (supervisé) et non étiquetées (non supervisé), exploite peu de données étiquetées et beaucoup de données non étiquetées, et améliore les performances des modèles en utilisant davantage de données.[45] [46]

La figure (1.5) d éemontre le processus d'un algorithme d'apprentissage semi-supervis ée.

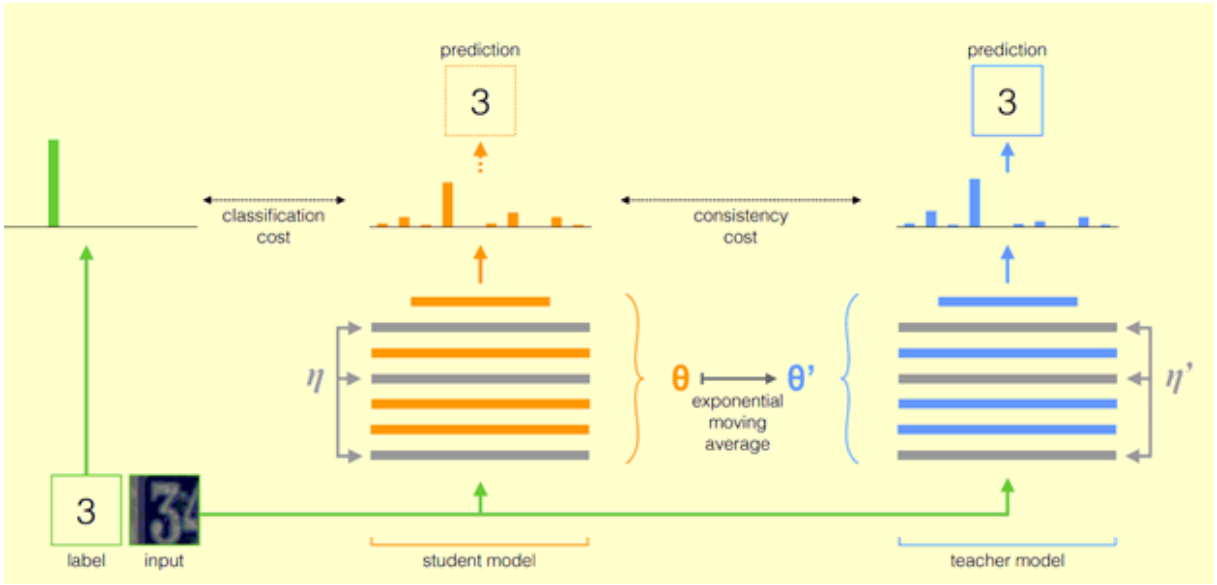


Figure 1.5 : Apprentissage semi-supervisé [4]

1.7.5 l'apprentissage par transfert (transfer learning)

Le transfert d'apprentissage consiste à réutiliser les connaissances acquises par un modèle sur une tâche source pour une nouvelle tâche cible connexe, permettant d'accélérer l'entraînement et d'obtenir de meilleures performances sur la tâche cible, surtout quand les données étiquetées sont limitées, et est très utilisé en apprentissage profond avec les réseaux de neurones convolutifs et récurrents.[47]

La représentation visuelle 1.6 détaille le processus de

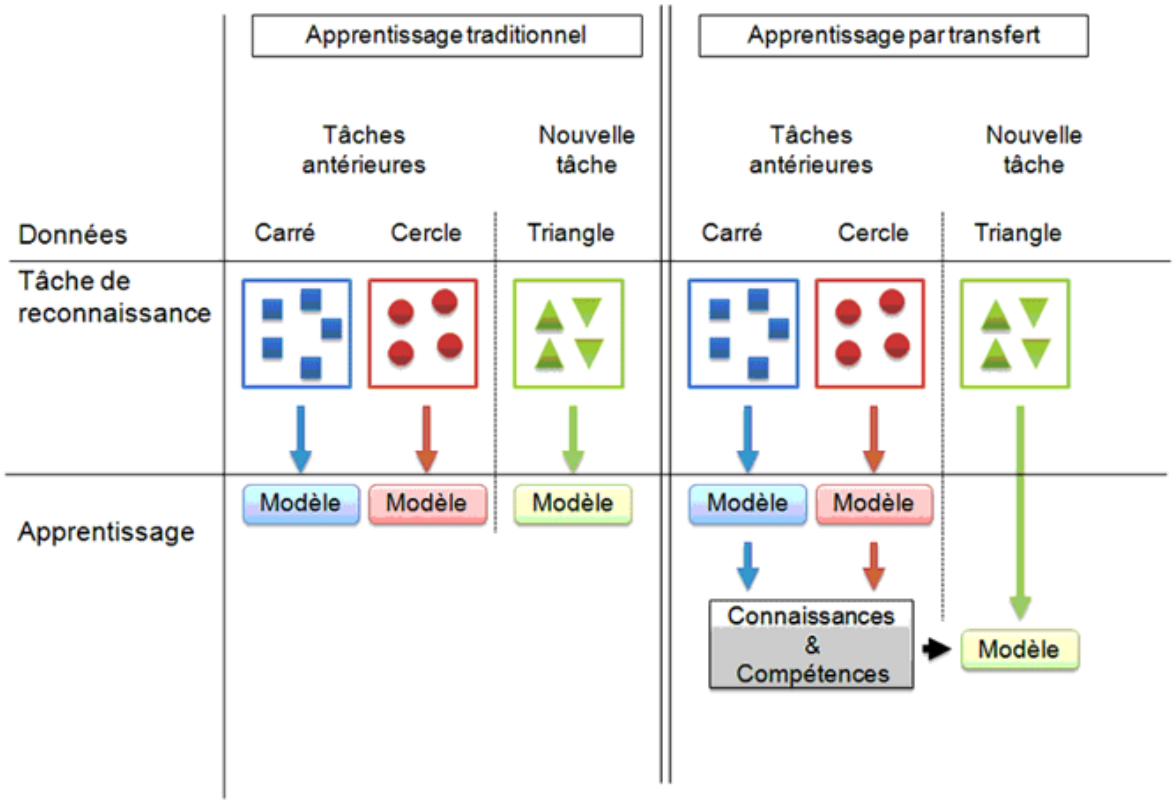


Figure 1.6 : l'apprentissage par transfert (transfer learning)[5]

1.8 L'apprentissage profond (deep learning)

L'apprentissage profond est une forme d'apprentissage automatique utilisant des réseaux de neurones artificiels à nombreuses couches. Ces architectures profondes permettent d'apprendre automatiquement des représentations hiérarchiques de données brutes comme les images ou les textes. Les couches successives extraient des caractéristiques de plus en plus abstraites, capturant ainsi des modèles complexes dans les données. [48]

1.8.1 La Différence entre l'apprentissage profond et l'apprentissage automatique

L'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning) sont deux approches liées mais distinctes dans le domaine de l'intelligence artificielle. L'apprentissage automatique englobe un large éventail de techniques visant à permettre aux systèmes d'apprendre et de s'améliorer à partir de données, sans être explicitement programmés. L'apprentissage profond est un sous-ensemble de l'apprentissage automatique qui utilise des réseaux de neurones artificiels profonds pour modéliser des données d'entrée complexes de manière hiérarchique. Contrairement aux méthodes d'apprentissage automatique traditionnelles, l'apprentissage profond ne nécessite pas une extraction manuelle des caractéristiques, mais apprend directement les représentations pertinentes à partir des données brutes. L'illustration 1.7 explique les différences entre l'apprentissage automatique et l'apprentissage profond, en illustrant comment chacun de ces domaines se distingue par ses applications et ses techniques [49].

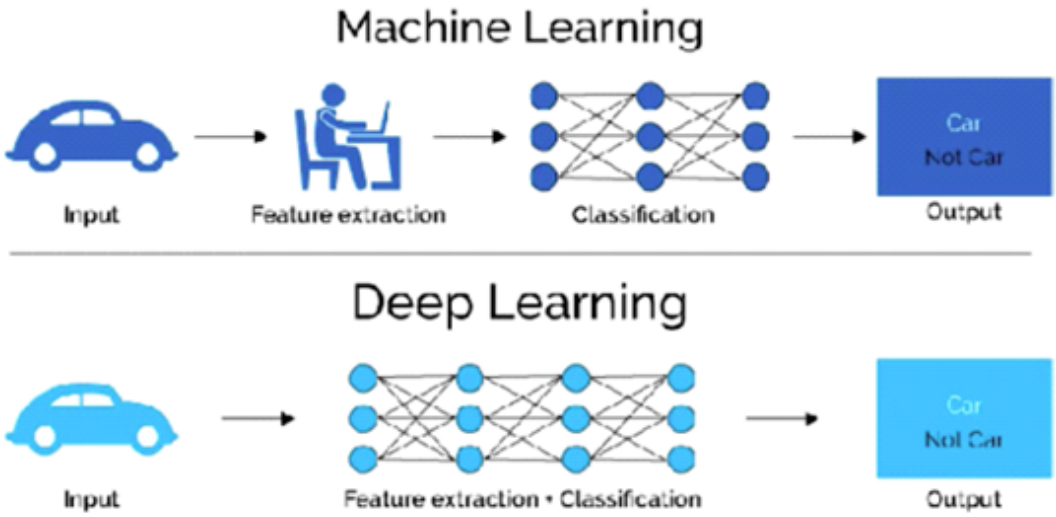


Figure 1.7 : Machine Learning vs Deep Learning[6]

1.8.2 Architectures de l'apprentissage profond

Voici une description des principales architectures utilisées en apprentissage profond

1- Réseaux de neurones convolutionnels (CNN) : Très performants pour les tâches de vision par ordinateur comme la classification d'images. Ils utilisent des opérations de

convolution pour extraire automatiquement les caractéristiques visuelles pertinentes.[38]

2- Réseaux de neurones récurrents (RNN) : Conçus pour le traitement de données séquentielles comme le texte ou la parole. Ils incorporent des boucles leur permettant de mémoriser des informations contextuelles.[38]

3- Réseaux de neurones récurrents à longue mémoire court-terme (LSTM) : Une variante améliorée des RNN capable de mieux capturer les dépendances à long terme dans les séquences.[38]

4- Réseaux de neurones de croyance profonde (DBN) : Réseaux non supervisés entraînés couche par couche pour apprendre des représentations profondes à partir de données non étiquetées.[38]

5- Réseaux antagonistes génératifs (GAN) : Combinant deux réseaux, un générateur et un discriminateur, pour générer de nouvelles données synthétiques réalistes.[38]

6- Réseaux Transformers : Architecture révolutionnaire dans le NLP utilisant l'attention pour modéliser les dépendances contextuelles. Le modèle BERT en est un exemple célèbre.[38]

Ces différentes architectures neuronales profondes sont adaptées à différents types de données et de tâches, permettant une modélisation puissante de patterns complexes dans les données.[38]

7- Auto-Encodeurs Variationnels(VAE) : Les VAE sont une architecture d'apprentissage profond non supervisé qui combine les auto-encodeurs et l'inférence variationnelle. Ils encodent les données d'entrée en une représentation latente suivant une distribution normale, dont on peut ensuite échantillonner pour reconstruire les données via un décodeur. Pendant l'entraînement, le VAE minimise à la fois l'erreur de reconstruction et la divergence entre la distribution latente et une gaussienne. Cela permet d'obtenir une représentation latente lisse, continue et interpolable, utile pour générer de nouveaux exemples plausibles. [50]

.

Conclusion

En conclusion, ce chapitre a mis en lumière les différents aspects de la vie des personnes aveugles, notamment les types de cécité et les défis qu'elles affrontent. Nous avons vu comment l'intelligence artificielle, à travers ses avancées en apprentissage automatique et apprentissage profond, a joué un rôle crucial dans l'amélioration de leur quotidien. Les réussites de l'IA dans ce domaine montrent un potentiel énorme pour continuer à développer des technologies qui offrent une plus grande autonomie et inclusion pour les personnes aveugles. L'innovation continue dans ces domaines promet de transformer encore davantage la manière dont les personnes aveugles interagissent avec le monde, rendant leur vie plus accessible et enrichissante.

Chapitre 2

Intégration de NLP dans l'IoT

Introduction

Dans ce chapitre, nous explorerons le traitement du langage naturel (NLP) et ses divers domaines d'application, en particulier la reconnaissance de texte et la reconnaissance vocale. Le NLP a révolutionné la manière dont les machines comprennent et interprètent le langage humain, ouvrant de nouvelles possibilités dans plusieurs secteurs. Nous examinerons également les travaux connexes dans les domaines de la reconnaissance de texte et vocale, mettant en lumière les avancées récentes et les défis persistants. En outre, nous aborderons l'Internet des objets (IoT), en montrant comment cette technologie peut être utilisée pour améliorer la vie des personnes aveugles. Nous discuterons de l'IoT en tant que logiciel et matériel, et comment ces composants interagissent pour créer des solutions innovantes et accessibles.

2.1 Traitement du langage naturel (NLP)

2.1.1 Définition

Le NLP est un domaine de l'intelligence artificielle qui se concentre sur l'analyse, la compréhension et la génération du langage naturel par les ordinateurs (Hirschberg et Manning, 2015). Il vise à permettre une communication naturelle entre les humains et les machines en utilisant des techniques issues de la linguistique, de l'informatique et de l'IA [51].

Cette illustration détaille le processus du traitement du langage naturel (NLP) :

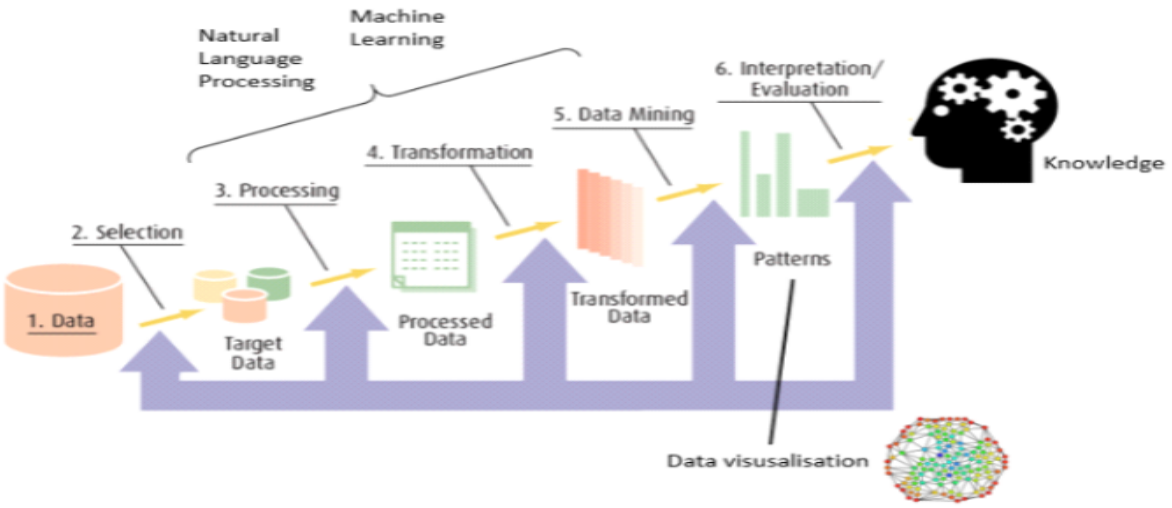


Figure 2.1 : Natural Language Processing [7].

2.1.2 Les domaines d'application du traitement du langage naturel

Le NLP trouve de nombreuses applications dans divers domaines. Il est largement utilisé pour la traduction automatique, permettant de traduire du texte ou de la parole d'une langue à une autre. Les moteurs de recherche web utilisent le NLP pour l'extraction d'informations et le résumé automatique de documents. Dans le domaine du marketing et de l'analyse des médias sociaux, le NLP permet l'analyse de sentiments pour comprendre les opinions des consommateurs. Les assistants virtuels conversationnels comme Siri ou Alexa reposent sur des techniques de NLP pour la reconnaissance vocale et la compréhension du langage naturel. Le NLP est également appliqué en santé pour l'exploration de dossiers médicaux et en cybersécurité pour la détection de menaces dans les communications électroniques. Avec les progrès de l'intelligence artificielle, les applications du NLP ne cessent de se multiplier[52].

Voici quelques exemples d'applications du traitement du langage naturel (NLP) :

- Traduction automatique :

  - Google Translate

- Moteurs de recherche web :

  - L'algorithme de recherche de Google utilise le NLP pour comprendre les requêtes

- Marketing et médias sociaux :

  - Outils d'analyse de sentiments pour comprendre les opinions des clients sur les produits/marques

- Santé :

  - Systèmes de diagnostic assisté par ordinateur

- Cybersécurité :

  - Détection d'activités suspectes dans les emails/messages

Ces exemples illustrent comment le NLP permet d'automatiser et d'améliorer de nombreuses tâches impliquant la compréhension du langage naturel. Cette représentation graphique dépeint le flux du processus de traduction automatique en traitement du langage naturel (NLP).[52]

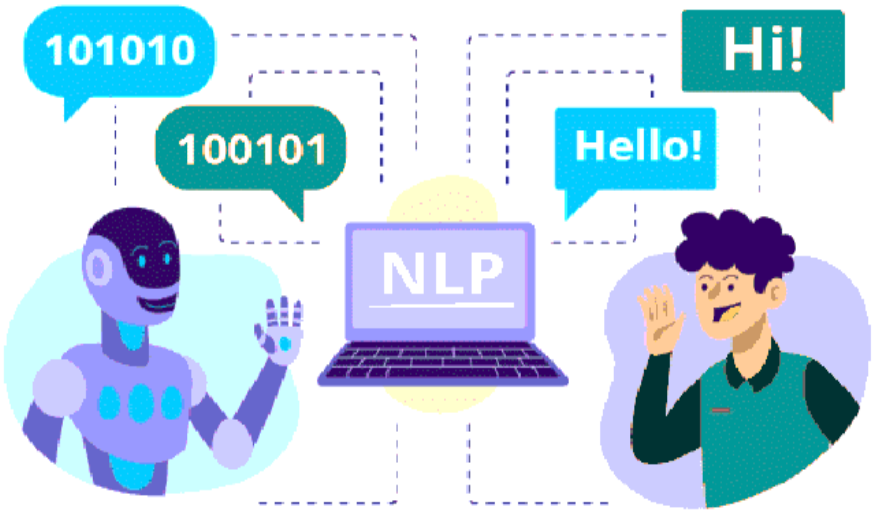


Figure 2.2 : Interaction Homme-Machine en NLP [8].

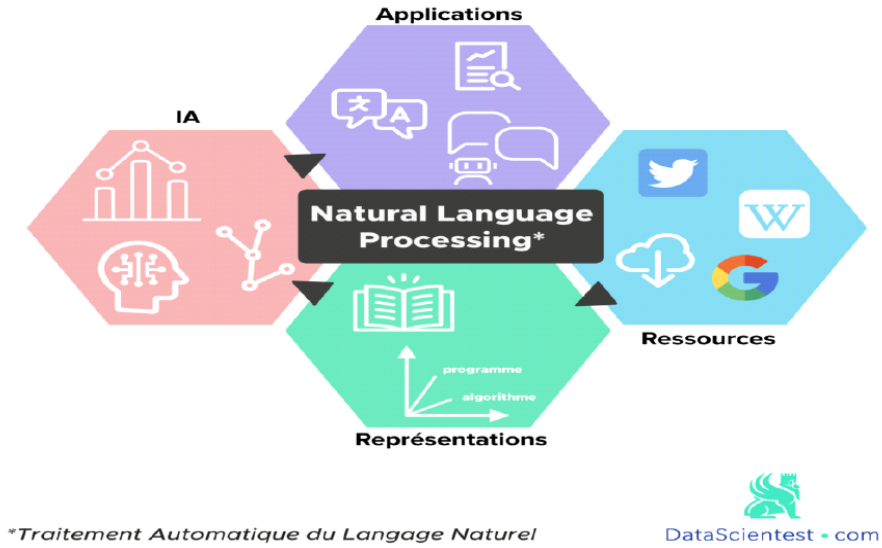


Figure 2.3 : Les Fondements du NLP [7].

2.1.3 L'importance de traitement du langage naturel

Le NLP est crucial car il permet de faire le lien entre les langues humaines et les langues formelles des ordinateurs. En rendant possible l'analyse et la génération du langage naturel, il ouvre la voie à une interaction plus naturelle entre les humains et les machines. C'est ce qui rend possible les assistants virtuels, la traduction automatique, les moteurs de recherche intelligents et de nombreuses autres applications impliquant la compréhension du langage. Au cœur des progrès en intelligence artificielle, le NLP est essentiel pour rapprocher l'informatique du langage humain et permettre une véritable communication entre les êtres humains et les systèmes intelligents [51].

2.1.4 Les principales méthodes utilisées en NLP

Le NLP combine des méthodes symboliques basées sur des règles linguistiques et des approches d'apprentissage automatique sur de grands corpus textuels. Les techniques d'apprentissage profond, notamment les réseaux de neurones comme les architectures Transformer (BERT), permettent de capturer automatiquement les régularités du langage. Les approches hybrides combinant règles et apprentissage sont aussi utilisées. Le NLP moderne s'appuie grandement sur les puissants modèles neuronaux entraînés sur d'immenses quantités de données textuelles, tout en s'inspirant des avancées linguistiques [53].

2.2 La reconnaissance de texte

2.2.1 Définition

La reconnaissance de texte fait référence aux processus logiciels utilisés par les systèmes OCR pour identifier et convertir le texte présent dans les images en données textuelles lisibles par une machine. Les deux principaux types d'algorithmes pour la reconnaissance de texte sont la correspondance de motifs et l'extraction de caractéristiques. La correspondance de motifs compare les images de caractères avec des glyphes stockés de polices et styles similaires. L'extraction de caractéristiques décompose les glyphes en traits caractéristiques comme les lignes, les boucles et les intersections, puis trouve la meilleure correspondance dans une base de données. Grâce à la reconnaissance de texte, le contenu textuel présent dans les images peut être converti en données textuelles exploitables par d'autres logiciels et applications [54].

2.2.2 Différentes techniques de reconnaissance de texte

Les différentes techniques de reconnaissance de texte [55] :

1. Détection de densité de coins présents dans l'image .
2. Méthodes d'apprentissage pour la détection du texte .
3. Mesure du gradient directionnel cumulé pour la détection du texte .
4. Pour la reconnaissance des caractères après détection du texte :
    - Segmentation des composantes d'un bloc de texte
    - Reconnaissance basée sur les caractéristiques des caractères ou sur des méthodes d'apprentissage comme les K-plus proches voisins, les modèles de Markov cachés, les perceptrons multicouches, les cartes auto-organisatrices de Kohonen ou les machines à vecteurs de support.
    - Correction des erreurs via la distance d'édition de Levenshtein et des post-traitements lexicaux et linguistiques.
5. Utilisation d'un logiciel OCR classique comme Tesseract de Google pour la reconnaissance des caractères à partir des images de texte détectées (approche choisie par

l'auteur).

2.3 Les travaux connexes de la reconnaissance de texte

De nombreuses recherches ont été menées dans ce domaine, mais nous avons sélectionné les études les plus récentes pour notre analyse.

Référence	Année	Titre d'article	Accuracy	Notes (contributions) ou résumé	Limitations dans la reconnaissance de texte
[56]	2019	What is wrong with scene text recognition model comparisons ? dataset and model analysis.	94,4%	Cette étude analyse en profondeur les défis liés à la comparaison des modèles de reconnaissance de texte dans des scènes naturelles. Les auteurs soulignent les problèmes liés aux datasets et aux métriques d'évaluation utilisées. Ils proposent des pistes d'amélioration pour une évaluation plus rigoureuse et significative.	Certaines limitations des modèles actuels sont mises en évidence, mais peu de solutions concrètes sont proposées.
[57]	2019	Scene text recognition from two-dimensional perspective.	94,5%	Cette étude propose une nouvelle approche pour la reconnaissance de texte dans des scènes naturelles, en utilisant une perspective bidimensionnelle. Les auteurs introduisent un modèle de réseau de neurones convolutionnel 2D capable de capturer à la fois les informations spatiales et sémantiques du texte.	L'approche proposée nécessite une étape de segmentation des lignes de texte, ce qui peut être complexe dans certains cas
[58]	2021	Read like humans : Autonomous, bidirectional and iterative language modeling for scene text recognition.	97.9% sur le jeu de données ICDAR 2015	Introduction d'un modèle de langage bidirectionnel et itératif qui imite la façon dont les humains lisent, améliorant la reconnaissance dans des scènes complexes.	Temps d'inférence plus long en raison de la nature itérative du modèle.

Tableu 2.1 : Les travaux connexes de la reconnaissance de texte

2.4 La reconnaissance vocale

2.4.1 Définition

La reconnaissance vocale est la capacité d'un système informatique à convertir la parole humaine en texte en analysant les composantes du signal audio. Ce processus implique

l'acquisition du signal vocal, son prétraitement, le décodage acoustique pour convertir les caractéristiques spectrales en unités linguistiques comme les phonèmes, et enfin le décodage linguistique pour déterminer la séquence de mots la plus probable [59].
Les systèmes modernes de reconnaissance vocale reposent principalement sur des techniques d'apprentissage profond telles que les réseaux de neurones récurrents et convolutifs, entraînés sur d'importants corpus de données vocales annotées. Cette technologie est utilisée dans de nombreuses applications comme la dictée vocale, les assistants vocaux intelligents, le contrôle par la voix ou encore la transcription automatique, bien que des défis subsistent en termes de robustesse face au bruit, aux accents, au vocabulaire spécifique et aux domaines d'application particuliers .[60]

2.4.2 Fonctionnement des systèmes de reconnaissance vocale

La technologie de la reconnaissance vocale est captivante et complexe, car elle permet de transformer la parole humaine en texte écrit. Il y a plusieurs étapes essentielles dans ce processus afin de garantir une transcription précise et fiable des paroles.[61]

1. **Appréhender le Signal Vocal :**Le processus de reconnaissance vocale commence par l'enregistrement de la voix de l'utilisateur. Les ondes sonores sont capturées par un microphone qui les transforme en un signal audio numérique. Il est crucial de réaliser cette conversion afin de permettre aux systèmes informatiques de traiter la parole.

2. **Pré-traitement :** Après avoir capturé le signal audio, il est prétraité afin d'améliorer sa qualité. Cela implique de diminuer le bruit de fond et de normaliser le volume. Cette purification du signal permet de réduire les perturbations et d'améliorer la précision des prochaines étapes.

3. **Segmentation en Trames :** Par la suite, le signal audio est divisé en petites fractions temporelles appelées trames, qui varient habituellement de 10 à 25 millisecondes. Grâce à cette segmentation, il est possible de repérer les fluctuations rapides de la parole et de traiter de manière plus efficace le signal.

4. **Extraction des Caractéristiques :** On analyse chaque trame du signal vocal afin de déterminer les caractéristiques acoustiques essentielles. Il est fréquent d'utiliser les

coefficients cepstraux en fréquence Mel (MFCC) afin de représenter les informations essentielles du signal vocal. Ces caractéristiques jouent un rôle essentiel dans la distinction des différents sons de la parole.

5. **Modélisation Acoustique :** La prochaine étape est de représenter les sons de la parole en utilisant les caractéristiques extraites. Les systèmes statistiques tels que les modèles de Markov cachés (HMM) et les réseaux neuronaux profonds (DNN) sont employés afin d'associer les propriétés sonores aux éléments sonores (phonèmes). La modélisation permet de distinguer les divers sons générés par la parole humaine.

6. **Modélisation Linguistique :** En même temps que la modélisation acoustique, on procède à une modélisation linguistique afin de prédire les séquences de mots potentielles. On utilise des modèles tels que les modèles n-grammes et les réseaux neuronaux récurrents (RNN) afin de prendre en considération le contexte linguistique et d'améliorer la précision de la reconnaissance.

7. **Décodage :** Le processus de décodage consiste à déterminer la séquence de mots la plus probable en se basant sur les modèles acoustiques et linguistiques. On utilise fréquemment des algorithmes tels que la recherche de faisceau (beam search) pour accomplir cette tâche. Grâce à cette étape, les caractéristiques acoustiques sont transformées en texte.

8. **Post-traitement :** Finalement, le texte produit est soumis à un traitement ultérieur afin de rectifier les éventuelles erreurs et d'améliorer sa lisibilité. Cela englobe la rectification grammaticale et l'ajustement au contexte afin de garantir que le texte final soit conforme à la parole initiale et compréhensible.[61]

2.4.3 Modèles utilisés pour la reconnaissance vocale

- **Modèles acoustiques neuronaux :** Les réseaux de neurones, en particulier les réseaux récurrents (RNN) comme les LSTM et les CNN, sont largement utilisés pour modéliser la relation entre le signal audio et les représentations linguistiques comme les phonèmes.

- **Modèles de langage neuronaux :** Des modèles neuronaux comme les RNN ou les Transformers sont utilisés pour modéliser les probabilités des séquences de mots,

améliorant la précision de la reconnaissance.

- Encodeurs-décodeurs avec attention : Cette architecture encode le signal audio en une représentation, puis décode cette représentation en texte à l'aide d'un mécanisme d'attention.

- **Adaptation du locuteur :** Des techniques comme l'adaptation des vecteurs d'entrée ou l'adaptation des couches de sortie permettent d'adapter les modèles acoustiques et de langage à un locuteur spécifique.

- **Adaptation au bruit :** Des méthodes comme le masquage spectral ou les modèles multi-styles rendent les systèmes de reconnaissance plus robustes au bruit et aux conditions d'enregistrement variées.

- **Reconnaissance en ligne de bout en bout :** Des modèles comme les RNN-Transducer ou les Transformer-Transducer permettent une reconnaissance en continu sans nécessiter de segmentation explicite du signal audio.

- **Apprentissage par transfert :** Le transfert de connaissances à partir de grands modèles pré-entraînés sur de vastes données non annotées, comme wav2vec 2.0, améliore les performances avec peu de données annotées [62].

2.5 Les travaux connexes de la reconnaissance vocal

Dans cette section, nous passons en revue les travaux récents liés à la reconnaissance vocale. Ces travaux explorent différentes approches basées sur l'apprentissage profond pour relever les défis de cette tâche complexe

Référence	Année	Titre d'article	Accuracy	Notes (contributions) ou résumé	Limitations dans la reconnaissance de vocale
[63]	2020	Conformer : Convolution-augmented Transformer for Speech Recognition	Taux d'erreur de 5,6% sur Switchboard	-Combine transformers et convolutions pour modéliser dépendances globales et locales dans l'audio. -Propose une nouvelle architecture appelée Conformer qui combine les avantages des transformers et des convolutions -Les transformers capturent les dépendances globales tandis que les convolutions modélisent les motifs locaux dans le signal audio -Utilise une attention convolutive en plus de l'attention multi-tête classique des transformers.	Nécessite beaucoup de données et de puissance de calcul.
[22]	2020	wav2vec 2.0 : A Framework for Self-Supervised Learning of Speech Representations.	Réduction du WER de 9,3% à 5,8% sur l'ensemble de test de Librispeech	-Apprentissage auto-supervisé de représentations audio à partir de données non annotées, améliorant les performances en aval. -Cadre d'apprentissage auto-supervisé pour obtenir des représentations audio génériques à partir de données non annotées. -Pré-entraîne un transformers encodeur pour prédire les représentations d'autres parties masquées du signal audio. -Ces représentations peuvent être transférées pour des tâches en aval comme la reconnaissance vocale avec peu de données annotées -Approche efficace pour les langues peu dotées en données transcrites.	Limité par les données non annotées disponibles
[64]	2014	Deep Speech : Scaling up end-to-end speech recognition.	Taux d'erreur de mot (WER) de 16,0% sur le benchmark Switchboard	Cet article décrit Deep Speech, l'un des premiers systèmes de reconnaissance vocale de bout-en-bout entraînés de manière end-to-end avec des réseaux neuronaux profonds. Malgré des performances inférieures aux systèmes conventionnels à l'époque, il a ouvert la voie à de nouvelles approches neuronales.	Les modèles requièrent d'énormes quantités de données transcrites pour l'entraînement. Les performances se dégradent dans des environnements bruyants ou avec des accents non standards.

Tableu 2.2 : Les travaux connexes de la reconnaissance de vocale

2.6 l'Internet des objets (IoT)

2.6.1 Définition

L'Internet des objets (IoT) désigne le réseau étendu d'appareils et d'objets physiques connectés à Internet, capables de collecter et d'échanger des données. Ces objets, équipés de capteurs et de puces électroniques, peuvent détecter leur environnement, communiquer entre eux et avec des systèmes centraux via Internet pour transmettre leurs données. L'IoT connecte ainsi le monde physique au monde numérique, permettant une collecte massive de données et un contrôle à distance d'objets du quotidien comme des thermostats, des montres, des électroménagers ou encore des véhicules. L'objectif est d'améliorer l'efficacité opérationnelle, de créer de nouveaux services et de faciliter les interactions entre les objets et leur environnement.[65]

2.6.2 L'architecture IoT

L'architecture IoT comprend typiquement quatre composants principaux [66] :

La couche des objets/capteurs

Cette couche englobe tous les objets connectés et capteurs déployés, qui collectent les données de l'environnement physique (température, mouvement, humidité, etc.).

La couche réseau

C'est l'infrastructure réseau qui permet la transmission sécurisée des données des objets vers la plateforme cloud. Elle utilise différents protocoles et technologies comme WiFi, Bluetooth, réseaux cellulaires, etc.

La couche de services cloud

Il s'agit de la plateforme cloud qui reçoit, traite et stocke les données provenant des objets connectés. Elle intègre des bases de données, des outils d'analyse des données, des services d'intelligence artificielle, etc.

La couche application

C'est l'interface utilisateur qui présente les données et analyses de manière visuelle et interactive, via des applications mobiles, des tableaux de bord, etc. Elle permet aussi de contrôler et gérer les objets connectés. Cette architecture en couches interconnectées assure la collecte, la transmission, le stockage, le traitement et la visualisation des données de l'IoT, permettant ainsi de nombreuses applications.

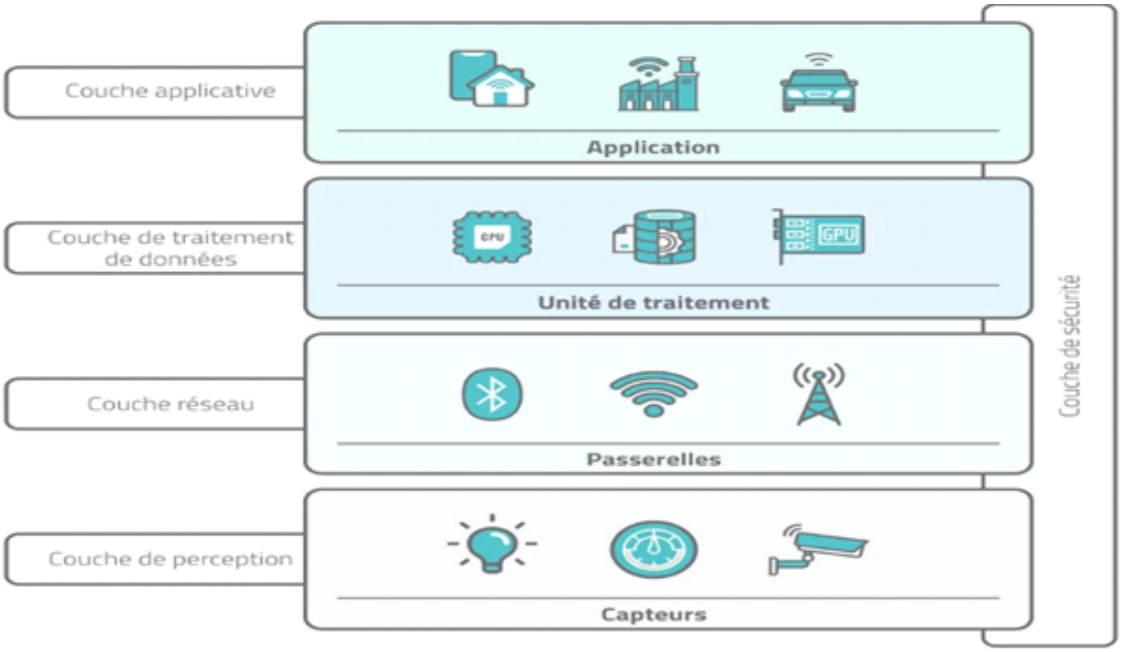


Figure 2.4 : IoT architecture [9]

2.6.3 Internet des objets pour les personnes aveugles

Les objets connectés de l'IoT combinés à l'intelligence artificielle ont le potentiel d'offrir de nouvelles solutions d'assistance aux personnes aveugles dans leur vie quotidienne. Quelques exemples [67] :

1. **Détection d'obstacles et navigation** Des capteurs IoT positionnés dans l'environnement peuvent détecter les obstacles et envoyer des alertes visuelles ou sonores à des lunettes connectées ou un smartphone pour guider la personne aveugle en toute sécurité.

2. **Reconnaissance d'objets** Des caméras connectées avec des algorithmes de vision

par ordinateur permettent d'identifier et de décrire vocalement les objets environnants pour l'utilisateur.

3. **Accès à l'information** Des étiquettes intelligentes RFID ou NFC sur les produits peuvent transmettre leurs informations (prix, ingrédients, mode d'emploi) à un smartphone compatible pour faciliter le shopping.

4. **Domotique** Un environnement domestique connecté avec des commandes vocales permet de contrôler facilement les appareils, la lumière, le chauffage, etc.

5. **Montre connectée** Une montre IoT vibrante couplée au GPS guide les déplacements et notifie des points d'intérêt à proximité. Bien que coûteuses, ces solutions IoT ont un fort potentiel pour améliorer considérablement l'autonomie et la qualité de vie des personnes aveugles.

2.7 IoT en tant que software

2.7.1 Raspbian

Raspbian est un système d'exploitation léger conçu pour les cartes Raspberry Pi, idéal pour les projets d'Internet des objets. Grâce à ses capacités de faible consommation d'énergie et sa compatibilité avec divers langages de programmation comme Python, Raspbian permet de transformer un simple Raspberry Pi en un puissant dispositif IoT à moindre coût. Cela en fait une plateforme privilégiée pour développer des applications IoT capables de collecter des données de capteurs, de contrôler des actionneurs et d'interagir avec le cloud. Raspbian offre ainsi un écosystème logiciel propice à l'expérimentation et à l'innovation dans le domaine de l'Internet des objets. [68]

Il existe différentes versions de Raspbian. Voici un aperçu des principales :

- **Raspbian "Wheezy" (2013)** - Basée sur Debian 7, première version officielle pour Raspberry Pi.

- **Raspbian "Jessie" (2015)** - Basée sur Debian 8, améliorations des performances et ajout de nouveaux paquets.

- **Raspbian "Stretch" (2017)** - Basée sur Debian 9, support des Raspberry Pi 3, noyau 4.9.

- **Raspbian "Buster" (2019)** - Basée sur Debian 10, noyau 4.19, nouveaux utilitaires de bureau.
- **Raspbian "Bullseye" (2021)** - Basée sur Debian 11, noyau 5.10, interface de bureau remaniée.
- **Raspbian "Bookworm" (prévu 2023)** - Basée sur la future version Debian 12.

À partir de 2022, le nom officiel est "Raspberry Pi OS" plutôt que Raspbian, bien que les versions récentes gardent une base Debian. Les principales différences entre versions sont les mises à jour du noyau Linux, des paquets logiciels et de l'environnement de bureau pour profiter des nouveaux Raspberry Pi [69].

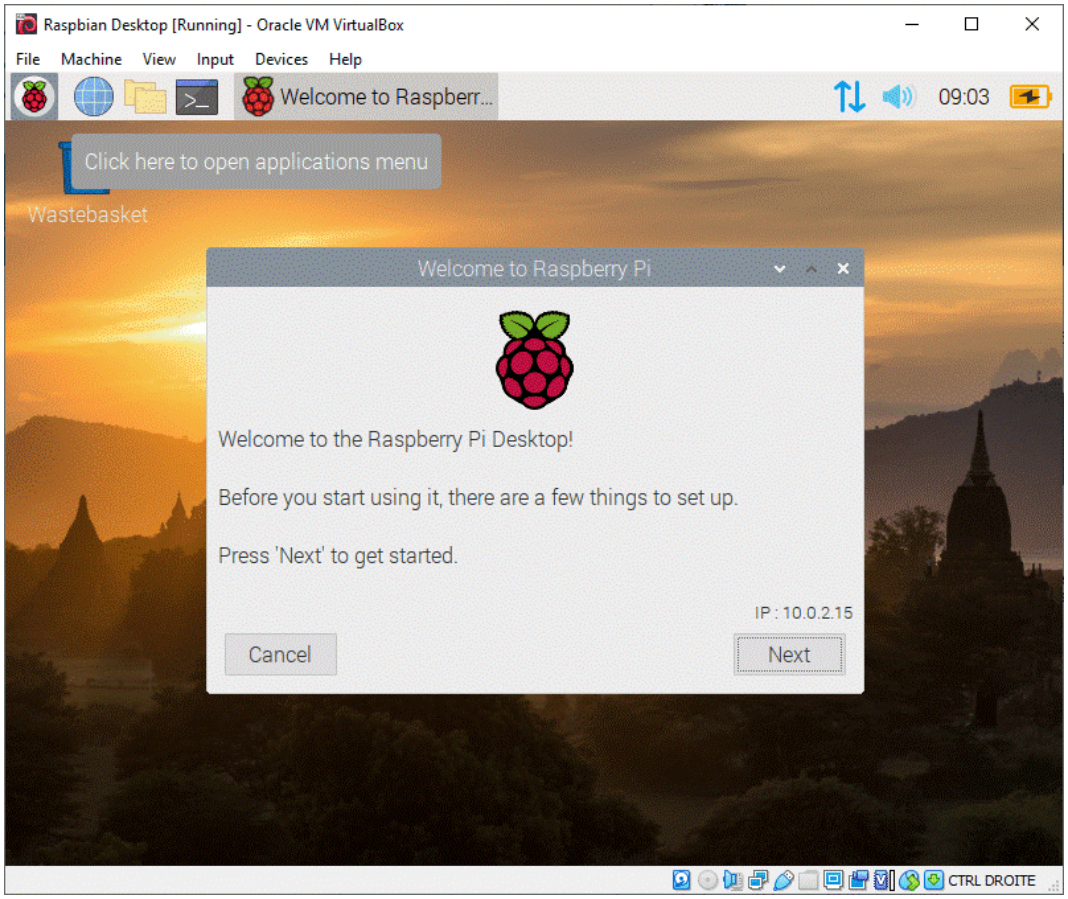


Figure 2.5 : Raspbian [10]

2.8 IoT en tant que hardware avec le Raspberry Pi :

Le Raspberry Pi est un petit ordinateur monocarte abordable et économe en énergie, idéal comme hardware pour les projets d'Internet des Objets (IoT). Avec ses interfaces

GPIO, son processeur ARM et sa connectivité réseau, le Raspberry Pi peut facilement être connecté à des capteurs et actionneurs pour créer toutes sortes de dispositifs IoT. Que ce soit pour surveiller des données environnementales, contrôler des systèmes domotiques ou collecter des données industrielles, le RPi offre une plateforme matérielle polyvalente et open source pour l'IOT. Couplé à un système d'exploitation léger comme Raspbian optimisé pour l'IoT, le Raspberry Pi se révèle un choix économique et performant pour prototyper et déployer des solutions innovantes dans l'Internet des Objets.[70]

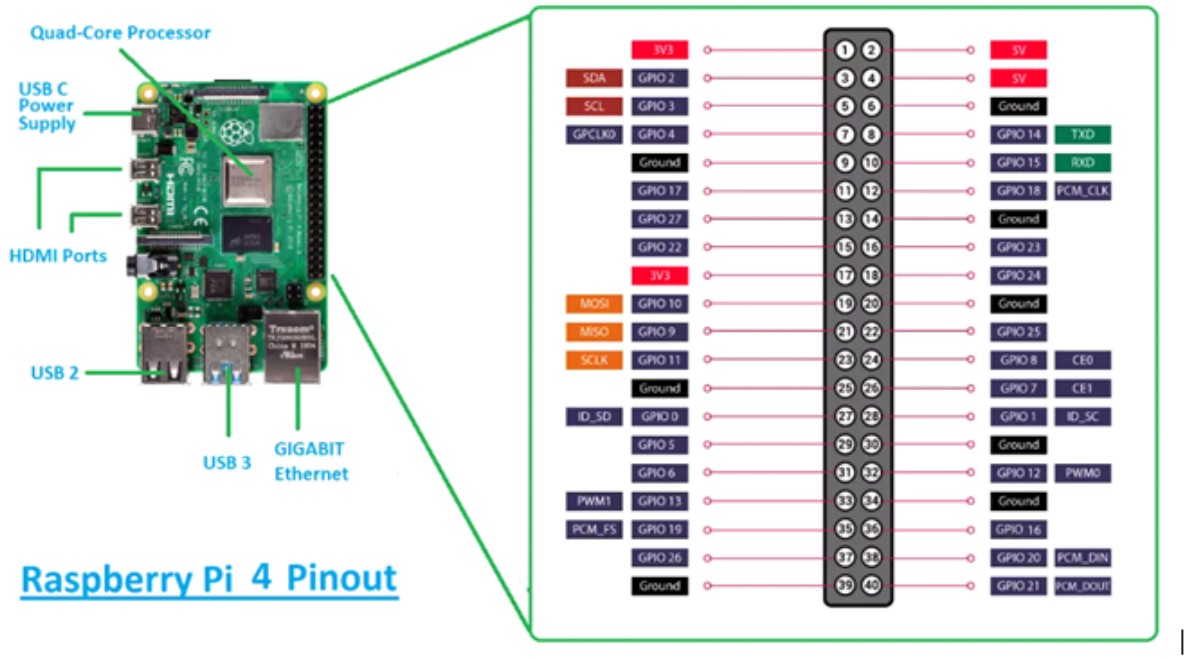


Figure 2.6 : Raspberry Pi 4 [11]

2.8.1 Présentation du Raspberry Pi :

Le Raspberry Pi est un nano-ordinateur monocarte à bas coût développé au Royaume-Uni par la Fondation Raspberry Pi. Malgré sa taille miniature, il s'agit d'un véritable ordinateur complet capable d'exécuter divers systèmes d'exploitation comme Raspberry Pi OS (anciennement Raspbian), une distribution Linux dérivée de Debian. Avec son processeur ARM, sa RAM, ses ports USB, Ethernet, HDMI et ses broches GPIO, le Raspberry Pi peut être utilisé pour de nombreuses applications : médiacenter, serveur web, station météo, émulateur de jeux rétro, etc. Mais il brille particulièrement dans le domaine de l'électronique et de la robotique grâce à ses capacités d'interfaçage avec des composants électroniques. Proposé à partir de 25$, abordable et peu énergivore, le

Raspberry Pi séduit un large public allant des makers aux développeurs en passant par les étudiants et les enseignants. Véritable outil éducatif, il permet d'initier facilement les débutants à la programmation et à l'électronique. Avec des millions d'unités vendues depuis 2012, le Raspberry Pi connaît un énorme succès mondial et conforte son statut de nano-ordinateur de référence pour les projets DIY et l'Internet des Objets. [71]

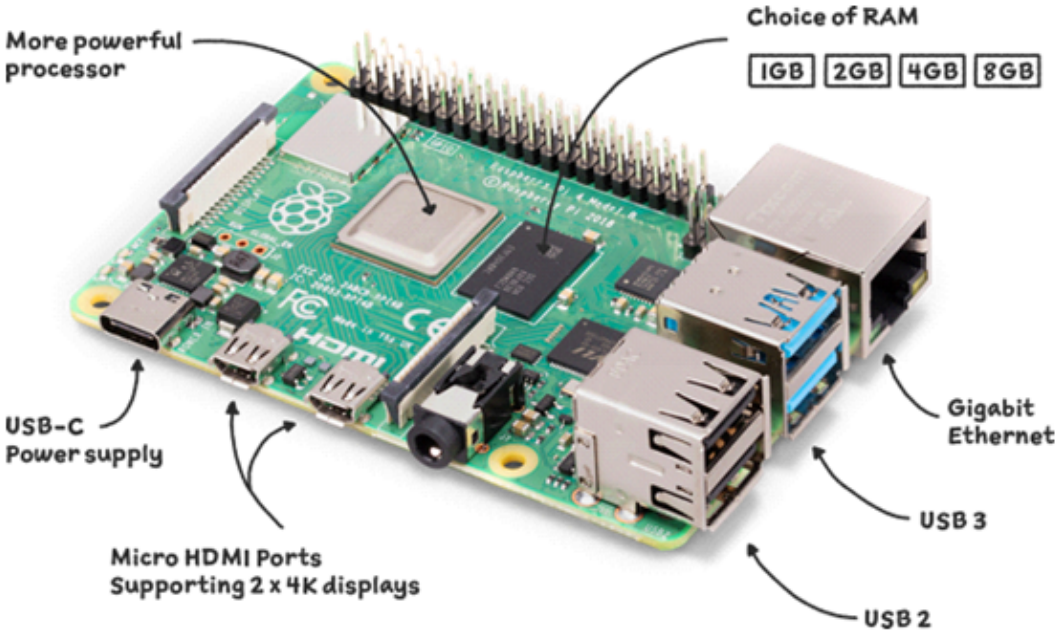


Figure 2.7 : Raspberry Pi [12]

2.8.2 Les types de Raspberry Pi :

Il y a plusieurs types de Raspberry Pi, chacun avec ses propres modèles principaux, Voici une brève explication de chaque type de Raspberry Pi :

Raspberry Pi 4 Modèle B :

Le plus puissant, avec jusqu'à 8Go de RAM et un processeur Quad-core 1,5GHz. Parfait pour des projets exigeants en ressources.[12]

Raspberry Pi 3 Modèle B+ :

Quad-core 1,4GHz, WiFi ac, Bluetooth 4.2. Une bonne mise à niveau du 3B. [13]

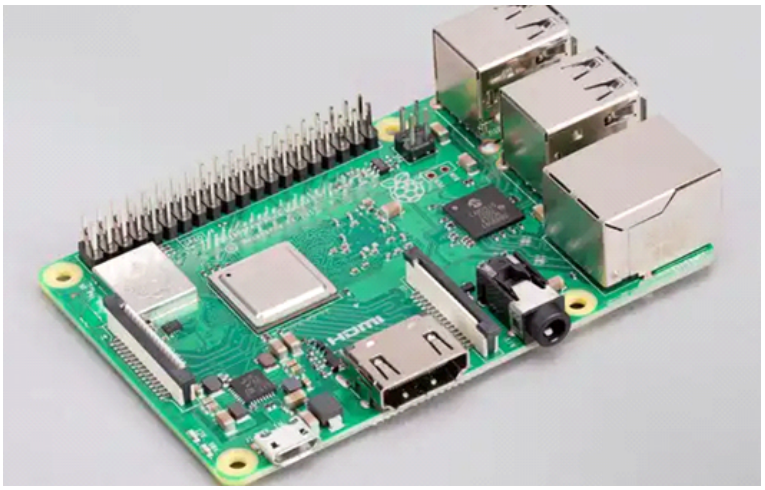


Figure 2.8 : Raspberry Pi 3 Modèle B+ [13]

Raspberry Pi 3 Modèle B :

Similaire au 3B+ mais WiFi n et Bluetooth 4.1 un peu moins rapides.[14]

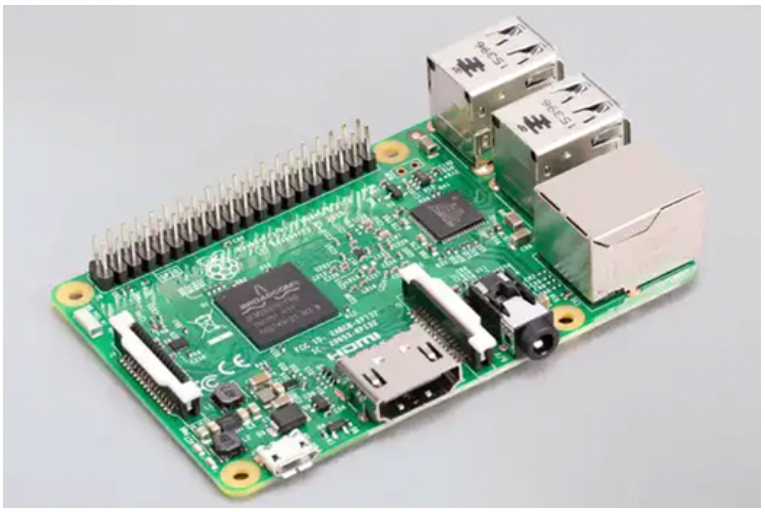


Figure 2.9 : Raspberry Pi 3 Modèle B [14]

Raspberry Pi 2 Modèle B :

Quad-core 900MHz, amélioration notable des performances par rapport aux premiers modèles.[72]

Raspberry Pi 1 Modèle B+ :

Processeur 700MHz, 512Mo de RAM. Une mise à jour incrémentielle du Raspberry Pi original.[15]

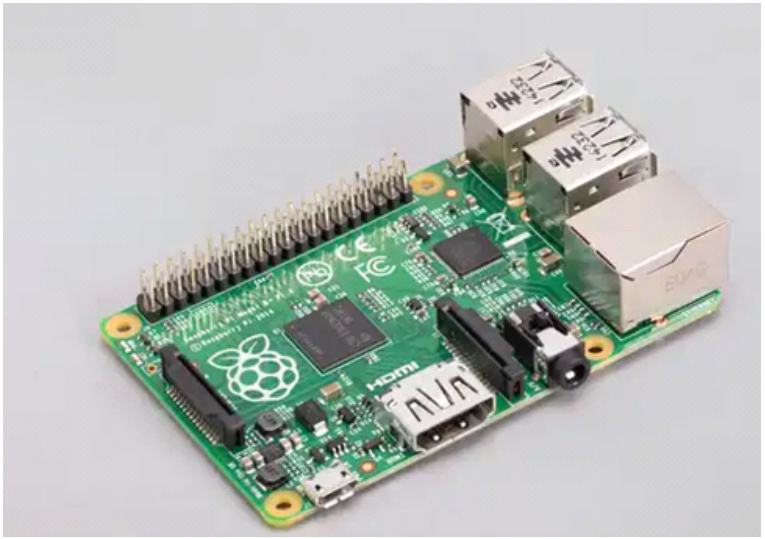


Figure 2.10 : Raspberry Pi 1 Modèle B+ [15]

Raspberry Pi Zero W :

Minuscule carte 1GHz avec WiFi et Bluetooth intégrés pour des projets compacts.[73]

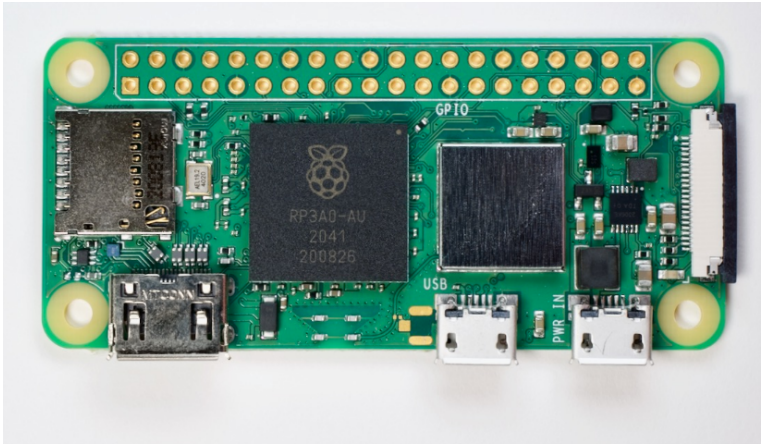


Figure 2.11 : Raspberry Pi Zero W [16]

Raspberry Pi Zero :

Encore plus petite que la Zero W, idéale pour l'embarqué très contraint en taille.[17]

Figure 2.12 : Raspberry Pi Zero [17]

Raspberry Pi 1 Model A+

Version compacte et très économe en énergie avec un processeur ARM11 700MHz et 256Mo de RAM. Parfait pour des projets légers et peu gourmands en ressources.[18]

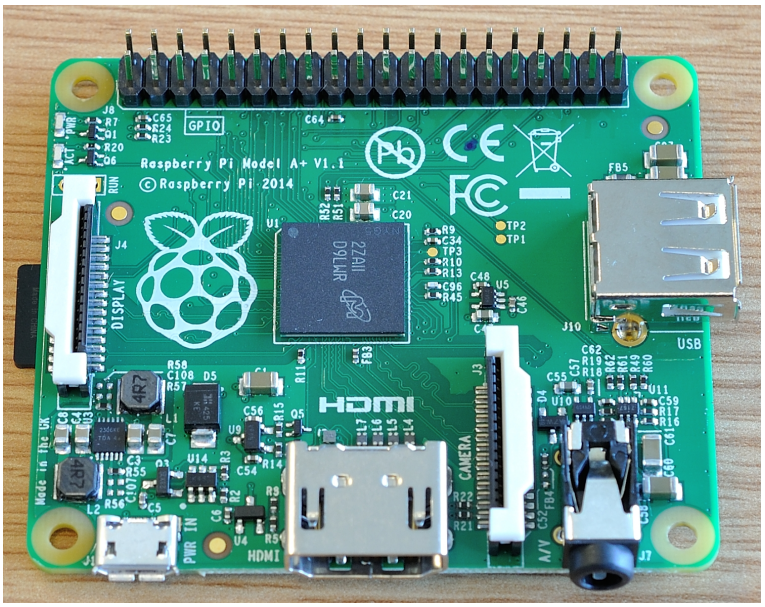


Figure 2.13 : Raspberry Pi 1 Model A+ [18]

Conclusion

En conclusion, ce chapitre a présenté une vue d'ensemble du traitement du langage naturel (NLP) et ses applications essentielles, notamment la reconnaissance de texte et

vocale. Nous avons exploré les travaux connexes dans ces domaines, soulignant les progrès réalisés et les défis à surmonter. De plus, nous avons examiné l'Internet des objets (IoT) et son potentiel pour améliorer la vie des personnes aveugles, en détaillant ses aspects logiciels et matériels. Les synergies entre NLP et IoT ouvrent la voie à des innovations qui promettent de rendre la technologie plus accessible et utile pour les personnes aveugles, transformant ainsi leur interaction avec le monde. Les développements continus dans ces domaines sont essentiels pour créer un environnement inclusif et connecté pour tous.

Chapitre 3

Conception et réalisation

Introduction

Dans ce chapitre, nous aborderons la conception globale d'un système destiné à faciliter l'accès des étudiants aveugles aux examens. Nous commencerons par une présentation de l'architecture générale du système. La partie matérielle se concentre sur les équipements nécessaires. Tandis que la partie logicielle inclut des technologies avancées de reconnaissance de texte et de synthèse vocale. Enfin, nous explorerons des méthodes de pointe basées sur le deep learning, telles que les modèles LSTM, WaveNet et Wav2Vec2, pour améliorer la précision et l'efficacité de la reconnaissance vocale et de la génération de la parole, garantissant ainsi une interaction fluide et intuitive pour l'utilisateur. Ce chapitre fournit une vue d'ensemble détaillée de chaque composant et de leur intégration, démontrant comment ces technologies se combinent pour créer un système performant pour les étudiants aveugles.

3.1 Conception

Le système que nous proposons est divisé en deux parties : la partie matérielle et la partie logicielle. L'objectif du projet est de créer un appareil intelligent que les étudiants aveugles peuvent utiliser. L'appareil se compose d'un Raspberry pi 4, d'une webcam, d'un casque microphone, les boutons poussoirs, et ainsi de suite. Des méthodes seront ensuite utilisées pour convertir les sujets d'examen pour les étudiants aveugles en audio audible, puis convertir les réponses de ces étudiants en texte écrit à l'aide de la technologie

automatique et de l'apprentissage profond. Pour cette raison, le système suggéré est créé comme indiqué en 3.1 :

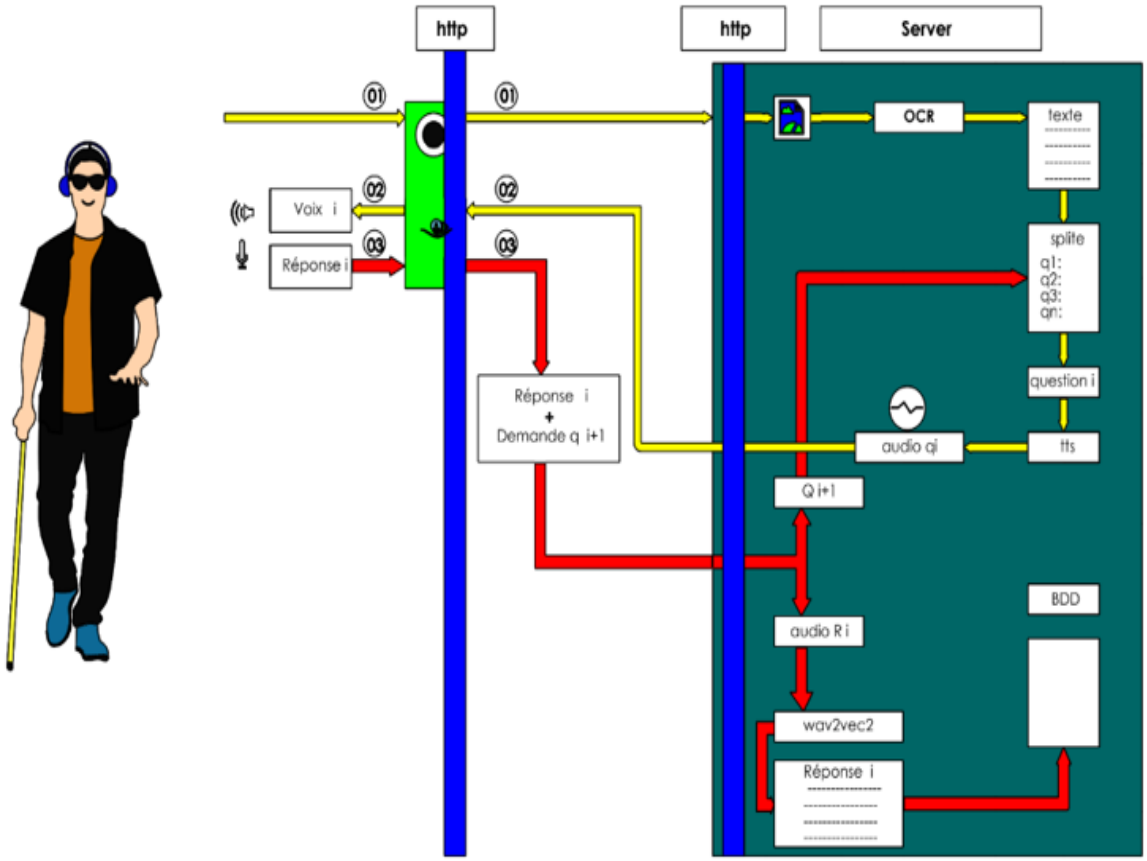


Figure 3.1 : Architecture générale

Architecture générale :

L'architecture présentée dans la figure 3.1 montre comment les étudiants aveugles peuvent utiliser un appareil appelé SDBS pour passer des examens sans l'aide directe d'un enseignant. L'étudiant prend une photo du sujet de l'examen avec l'appareil équipé d'une caméra. Cette photo est envoyée au serveur via HTTP où elle est traitée par un OCR (Reconnaissance Optique de Caractères) qui convertit l'image en texte. Ce texte est divisé en segments représentant les questions de l'examen, ensuite, chaque segment est converti en audio par un module TTS (Text-to-Speech). Les fichiers audio des questions sont renvoyés au SDBS, qui restitue les questions sous forme de voix pour que l'étudiant puisse les écouter. L'étudiant répond aux questions oralement, et l'appareil SDBS capte ces réponses vocales, qui sont ensuite envoyées au serveur. Le serveur convertit les réponses

audio en texte par l'utilisation des techniques de deep learning. Finalement, ces réponses sont stockées dans une base de données. Ce processus se répète pour chaque question, permettant ainsi à l'étudiant de passer l'examen de manière autonome, éliminant ainsi le besoin d'un enseignant qui lit les questions et écrit les réponses.

3.2 Partie matérielle

La partie matérielle doit être composée des pièces listées ci-dessous :

- Raspberry pi 4 Model B (4 GB RAM)
- Webcam qui permet de prendre une photo
- Boutons poussoirs
- Plaquette perforee double face permettant de fixer les boutons
- Carte mémoire
- Casque microphone
- Power bank pour l'alimentation
- Conception 3D pour réaliser la couverture
- Un couvercle pour protéger les éléments mentionnés précédemment

Nous avons sélectionné les composants suivants pour notre appareil :

- Raspberry Pi 4 comme micro-ordinateur (figure 3.2).

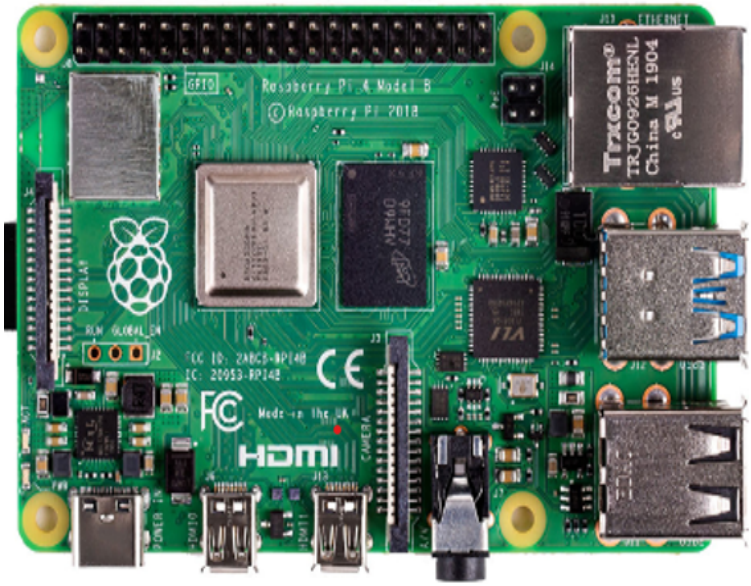


Figure 3.2 : Raspberry Pi 4 model b

- Webcam (figure 3.3) .

Figure 3.3 : webcam USB

- Casque microphone (figure 3.4).

Figure 3.4 : Casque microphone

- Plaquette perforee double face (figure 3.5).

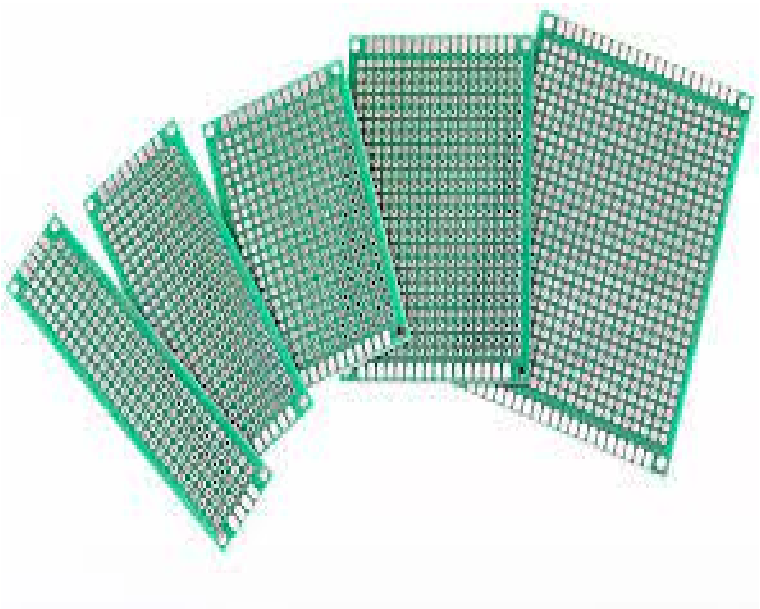


Figure 3.5 : Plaquette perforee double face

- Boutons poussoirs (figure 3.6).

Figure 3.6 : Boutons poussoirs

Architecture matérielle

La Figure 3.7 représente la conception du système matériel

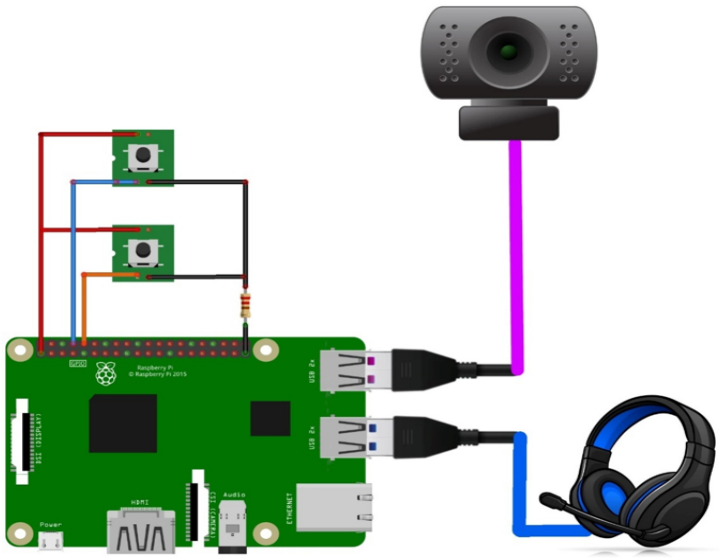


Figure 3.7 : Conception du système matériel

Le circuit de la figure 3.7 est représenté à l'aide du programme "Fritzing" [74]. Les pièces réelles sont maintenant connectées comme suit :

- La carte mémoire est branchée sur le Raspberry Pi4 .
- Le bouton responsable de l'image est connecté au pin GPIO 14 et au pin 3V3 du Raspberry pi 4.
- Le bouton responsable du son est connecté au pin GPIO 15 et au pin 3V3 du Raspberry pi 4.
- Webcam (USB) est connectée au port USB 2.0 du Raspberry pi 4.
- Casque microphone est connecté au port USB 2.0 du Raspberry pi 4.
- Power bank est connecté au USB-C power port 5V/3A du Raspberry pi 4.
- Les deux boutons sont connectés au GND du Raspberry pi 4.

3.3 Partie logicielle

Base de données

Un ensemble de données, ou dataset, est une collection structurée de données utilisée pour entraîner et tester les systèmes de reconnaissance automatique de la parole (ASR),Le Tableau 3.1 fournit une liste des ensembles de données vocales disponibles, ainsi que leurs caractéristiques principales telles que le temps total et les langues parlées [23] :

Dataset	Open-Source	Hours	Language
LibriSpeech	Yes	1000	English
HUB 5	No	2000	English
TIMIT	No	5.6	English
The CHiME-5	No	50.12	English
TED-LIUM	Yes	452	English
The Spoken Wikipedia	Yes	1005	Multilingual
Common Voice	Yes	1900	Multilingual
CSTR VCTK	Yes	09	English
AISHELL-1	Yes	107	Mandarin
Persian Consonant Vovel Combination (PCVC)	Yes	-	Persian
Arabic Speech Corpus	Yes	3.7	Arabic

Tableu 3.1 : Liste des ensembles de données vocales [23]

LibriSpeech

LibriSpeech est reconnu comme l'un des corpus de synthèse vocale les plus couramment utilisés disponibles en open source. Le corpus comprend un total de 1000 heures de livres audio échantillonnées à 16 kHz ainsi que leurs transcriptions correspondantes. Pour gérer le volume important de données collectées [23],L'ensemble de données LibriSpeech est structuré en différentes parties :

1. **Ensembles d'entraînement**
   train-clean-100 : 100 heures d'enregistrements audio propres.
   train-clean-360 : 360 heures d'enregistrements audio propres.
   train-other-500 : 500 heures d'enregistrements audio avec plus de bruit et des conditions variées.

2. **Ensembles de développement :**

dev-clean : Enregistrements propres pour le développement (validation), environ 5,4 heures.

dev-other : Enregistrements avec bruit pour le développement, environ 5,3 heures.

3. **Ensembles de test**

   test-clean : Enregistrements propres pour le test, environ 5,4 heures.

   test-other : Enregistrements avec bruit pour le test, environ 5,1 heures.

Nous utilisons LibriSpeech pour l'entraînement de nos modèles ASR en raison de son ensemble de données étendu et diversifié, pour évaluer leur efficacité dans des environnements contrôlés grâce à l'utilisation d'ensembles de données de développement et de test, et comme référence fondamentale dans nos recherches et nos avancées en matière de reconnaissance vocale. Nous avons téléchargé l'ensembles de données librispeech à partir de :`https://www.openslr.org/12`

3.3.1 Reconnaissance de texte à l'aide de pytesseract OCR

Tesseract :

Tesseract, un moteur de reconnaissance optique de caractères, a été créé par les ingénieurs de HP (Hewlett Packard) de 1984 à 1995, puis abandonné. Après une décennie de dormance, le code a été rendu public en 2005 sous la licence Apache, ce qui a conduit à sa renaissance sous la direction de Google [75].

Figure 3.8 : Moteur de reconnaissance optique de caractéres Tesseract [19]

Initialement limité aux caractères ASCII et s'adaptant parfaitement aux caractères UTF-8, les capacités actuelles du système incluent la reconnaissance de plus de 100 langues ainsi qu'un mécanisme d'apprentissage destiné à être amélioré. Ce mécanisme est conçu pour être compatible avec un large éventail de systèmes d'exploitation et est complété par une variété de wrappers qui facilitent son utilisation dans plusieurs langages de programmation. De plus, Tesseract intègre divers algorithmes de prétraitement tels que le redimensionnement, la binarisation et la réduction du bruit. De plus, le moteur est capable de fournir des résultats dans de nombreux formats [76].

Nous utilisons la reconnaissance optique de caractères (OCR) Tesseract en conjonction avec la bibliothèque Pytesseract dans le langage de programmation Python, car elle s'avère être une approche très efficace pour transcrire le texte contenu dans des images dans un format adapté à la manipulation numérique. Le processus de reconnaissance optique de caractères (OCR) est illustré visuellement sur la Figure 3.9.

Figure 3.9 : Le processus de reconnaissance optique de caractères (OCR).

La figure 3.9 présente le processus de conversion d'un document imprimé en texte numérique :

En premier lieu, on numérise le sujet imprimé en utilisant un appareil de numérisation, ce qui donne naissance à une image numérique du document. Par la suite, cette image est envoyée sur un serveur web. La prochaine étape consiste à utiliser la reconnaissance optique de caractères (OCR), une technologie qui examine l'image numérisée et extrait les caractères imprimés afin de les transformer en texte éditable.

Splite de texte :

Le processus de division du texte (splitting) intervient après l'extraction initiale du texte par l'OCR (Reconnaissance Optique de Caractères). Une fois que l'OCR a transformé l'image capturée en texte brut, ce texte est divisé en plusieurs segments individuels représentant les différentes questions de l'examen.

3.3.2 Google Text to Speech (gTTS)

Le TTS (Text-to-speech) consiste à convertir des mots en une forme audio. Le logiciel reçoit un texte de l'utilisateur et effectue une induction logique sur le texte. On passe

ce texte processé dans le prochain bloc où le processus de traitement numérique des signaux digitaux est effectué sur le texte processé. En utilisant de nombreux algorithmes et transformations, ce texte traité est finalement converti en format de parole. Toute cette procédure implique la synthèse du discours. Il existe plusieurs APIs disponibles pour convertir un texte en parole en Python. Une de ces APIs est la Google Text to Speech API, également appelée gTTS API. gTTS est un outil très simple à utiliser qui convertit le texte entré en audio, qui peut être sauvegardé en format mp3 [77].

Figure 3.10 : gtts (Google Text-to-Speech).

Dans notre projet nous utilisons la bibliothèque ‘ gtts‘ afin de convertir un fichier texte (.TXT) en fichier audio (.MP3). La figure 3.10 montre ce processus de conversion. Le début de ce processus consiste à préparer un fichier texte contenant le texte à transformer qui est la question a poser à l'étudiant. Par la suite, on utilise la bibliothèque gtts pour lire le contenu du fichier texte et produire un fichier audio au format MP3. Après la création de ce fichier, il est accessible sur n'importe quel lecteur audio compatible, ce qui permet d'accéder au texte sous forme audio.

3.3.3 Reconnaissance de voix en utilisant le deep learning

Grâce à sa capacité à détecter des patterns complexes, à gérer des données de grande taille, à effectuer des apprentissages complets, à gérer des dépendances à long terme, à être résistant aux données bruyantes et à être scalable, deep learning est souvent privilégié par rapport aux méthodes traditionnelles de machine learning pour la prévision. Les modèles de deep learning sont particulièrement adaptés à la prédiction car ils comprennent de nombreuses couches et des représentations hiérarchiques qui permettent à ces modèles

d'apprendre automatiquement des relations complexes dans les données. Ils peuvent traiter des données à grande échelle et séquentielles, extraire des caractéristiques pertinentes et même faire des prédictions efficaces en présence de bruit. Les modèles d'apprentissage profond peuvent bénéficier de la parallèle et de la taille efficace avec de grands datasets. Bien que le choix entre deep learning et machine learning traditionnel soit influencé par diverses variables, les avantages de deep learning en font un outil puissant pour prévoir des tâches.

1- Long short-term memory (LSTM)

LSTM (mémoire à court terme) est une variante d'un modèle RNN.Les données antérieures de séries chronologiques à long terme peuvent être rappelées par un modèle LSTM, qui a également un contrôle automatique pour maintenir les caractéristiques pertinentes ou éliminer les caractéristiques non pertinentes dans l'état de la cellule. Trois portes sont présentes dans un modèle LSTM pour contrôler les caractéristiques, à savoir la porte d'entrée, la porte d'oubli et la porte de sortie, comme illustré dans la figure 3.11. La porte d'entrée assure la circulation des nouvelles informations dans l'état de la cellule. Les informations antérieures sont supprimées par la porte d'oubli, peu importe l'état de la cellule. L'information extraite de l'état de la cellule est régulée par la porte de sortie, qui détermine ensuite l'état caché suivant. En utilisant ces portes, un modèle LSTM a la capacité d'enregistrer ou de supprimer automatiquement la mémoire stockée [78].

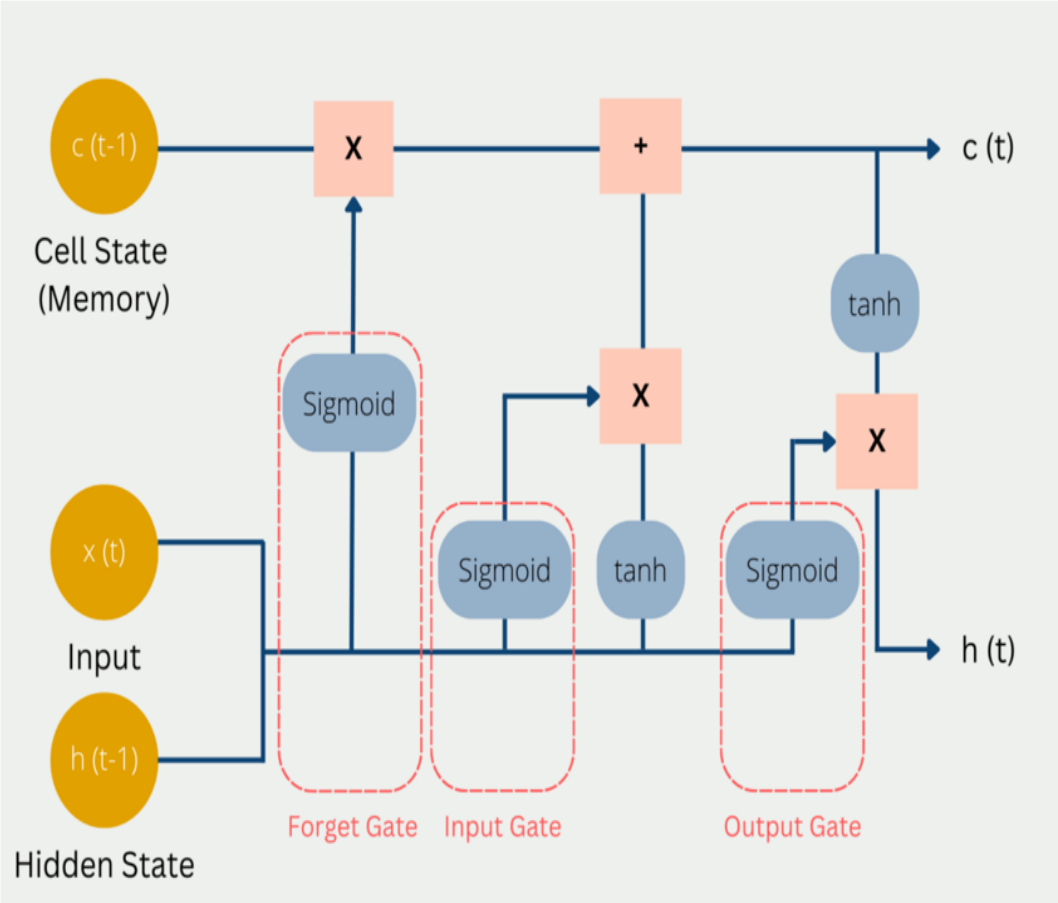


Figure 3.11 : Réseaux neurones LSTM [20].

Architecture de LSTM

- Dans notre architecture de reconnaissance vocale, nous utilisons plusieurs couches pour convertir les données audio en séquences de texte. La couche initiale, appelée InputLayer, reçoit des séquences de longueur maximale (100 pas de temps) avec 13 coefficients MFCC par pas. Ensuite, une couche Conv1D avec 32 filtres, une taille de noyau de 5 et une activation ReLU sont utilisées pour traiter cette entrée, ce qui permet d'extraire des caractéristiques locales pertinentes des séquences MFCC.

- Une fois la convolution terminée, on applique une couche de BatchNormalization afin de normaliser les activations et accélérer l'entraînement tout en améliorant la stabilité du réseau. Le modèle est ensuite composé de deux couches bidirectionnelles LSTM. La version de 128 unités de la première Bidirectional LSTM intègre un dropout de 0.25 afin de diminuer le risque de surapprentissage. La conception de cette couche vise à prendre en compte les liens temporels dans les deux sens (passé et

futur) des séquences d'entrée. La deuxième version du LSTM bidirectionnel dispose de 64 unités et d'un dropout de 0.25, ce qui permet une réduction de la complexité tout en maintenant la capture des relations temporelles en bidirection.

- Finalement, l'application d'un réseau dense avec une activation softmax à chaque étape de la séquence permet de prédire la répartition des probabilités pour chaque caractère à chaque étape temporelle. Le tokenizer détermine le nombre de neurones dans cette couche finale en fonction de la taille du vocabulaire. Cela donne au modèle la possibilité de produire des séquences de caractères qui correspondent aux transcriptions des données audio d'entrée.

- Il y a un total de 335,197 paramètres dans le modèle, dont 335,133 sont entraînables et 64 non-entraînables. Deux callbacks sont employés afin d'améliorer l'entraînement : EarlyStopping interrompt l'entraînement si la perte de validation ne s'améliore pas pendant 10 époques, et ModelCheckpoint sauvegarde le modèle le plus efficace en fonction de la perte de validation. Ce modèle a été développé afin de traiter des données audio en extraisant des caractéristiques MFCC, en les passant à travers des couches convolutives et récurrentes bidirectionnelles, et en générant des séquences de texte qui correspondent aux transcriptions des séquences audio.

Pourquoi cette architecture de ce modèle ?

Les données vocales présentent une nature séquentielle intrinsèque, avec des liens entre les différentes étapes temporelles. En utilisant une combinaison de couches Conv1D et LSTM, il est possible de capturer ces dépendances de manière efficace. Les LSTMs bi-directionnels permettent de saisir le contexte des états passés et futurs, ce qui permet d'avoir une compréhension plus approfondie de la séquence.

L'utilisation de la normalisation par couches et du Dropout permet de stabiliser le processus d'entraînement et de prévenir le sur apprentissage, ce qui rend le modèle plus solide et capable de s'appliquer à de nouvelles données.

Le mécanisme d'attention utilisé dans les modèles Transformer est extrêmement efficace pour détecter les dépendances à long terme, ce qui revêt une importance capitale pour des tâches telles que la reconnaissance vocale.

Conséquences concrètes :

Les caractéristiques MFCC sont couramment utilisées comme caractéristiques d'entrée dans le traitement de la parole. Ils captent les principales caractéristiques du signal vocal. Le modèle est complexe : La complexité et la performance sont équilibrées grâce à l'architecture, ce qui assure que le modèle puisse apprendre de manière efficace sans être trop complexe.

2- Wavenet

WaveNet a été initialement présenté par Google's DeepMind en 2016. Contrairement à RNN et à d'autres modèles d'attention, WaveNet a été conçu pour prendre en charge des ensembles de données séquentiels longs tels que les ondes sonores et les données de vibration. La base de WaveNet consiste à prendre la probabilité conditionnelle du signal audio brut à chaque point d'échantillon et à la combiner avec les points d'échantillon précédents. Donc, chaque point de données d'un échantillon séquentiel transmet des informations provenant des étapes précédentes de temps au sein de la séquence [79]. La figure 3.12 montre l'architecture de speech-to-text wavenet.

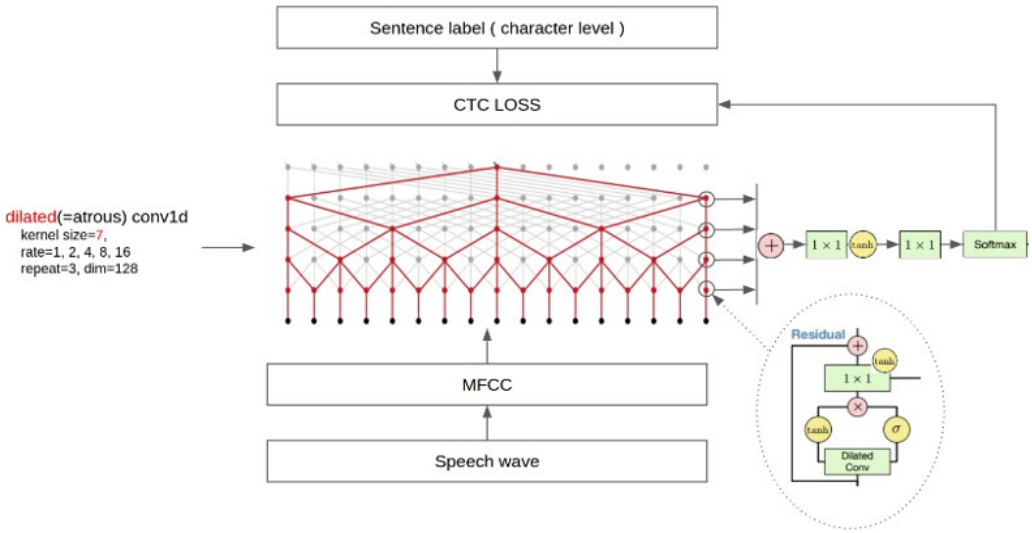


Figure 3.12 : Architecture de wavenet (speech-to-text)[21].

Architecture de wavenet

Le modèle wavenet que nous présentons est un réseau de neurones convolutif fonctionnel avancé, spécialement conçu pour gérer des séquences temporelles complexes comme des données audio. Elle débute avec une couche d'entrée (InputLayer) qui accepte des

séquences d'entrée de la forme (None, 100, 13), où Ńoneést la taille du lot, 100 est la longueur de la séquence temporelle et 13 est le nombre de caractéristiques par pas de temps.

Par la suite, on applique une série de couches Conv1D. L'utilisation de 64 filtres dans ces couches Conv1D permet d'extraire des caractéristiques locales des séquences d'entrée. Les couches de multiplication (Multiply) et d'addition (Add) sont suivies de chaque couche de convolution, ce qui permet de combiner les sorties des convolutions de manière non linéaire, ce qui accroît la capacité du modèle à apprendre des relations complexes au sein des données. Dans cette architecture, la structure des couches Conv1D, suivie de couches Multiply et Add, est répétée à plusieurs reprises (au moins 17 fois). Au fur et à mesure de chaque étape, cette répétition permet au modèle de saisir des caractéristiques de plus en plus complexes et de plus haut niveau. En utilisant ces différentes couches, le modèle peut extraire des caractéristiques subtiles et représenter des dépendances à long terme dans les séquences d'entrée, ce qui est essentiel pour des tâches comme la reconnaissance vocale ou la génération de texte à partir de la parole.

Ensuite, on applique une couche d'activation (Activation) afin d'introduire la non-linéarité, ce qui permet au modèle de modéliser des relations complexes dans les données. Les couches Conv1D les plus récentes sont équipées de filtres et de tailles de noyaux spécifiques afin de générer des sorties finales appropriées à la tâche. Par exemple, elles ont la capacité de convertir les données extraites en prédictions catégorielles ou en valeurs continues, en fonction des exigences de la tâche.

La complexité et la conception de cette architecture, qui comprend un total de 215,069 paramètres, tous entraînables, permettent de traiter de manière efficace des séquences temporelles complexes. L'utilisation stratégique des convolutions et des opérations non linéaires, ainsi que sa profondeur, le rendent particulièrement adapté à des applications comme la synthèse vocale, où il est crucial de capturer les nuances fines et les dépendances à long terme afin de produire des sorties de qualité supérieure.

3- Le modèle wav2vec 2.0

Facebook AI a développé Wav2Vec 2.0, un modèle de deep learning qui vise à l'apprentissage auto-supervisé des représentations de la parole à partir de l'audio brut. Il adopte une architecture de type Transformer afin de détecter les liens à long terme dans

les séquences audio et est pré-entraîné de manière automatique sur de grandes quantités de données audio non étiquetées, puis affiné sur des données étiquetées pour des tâches spécifiques telles que la reconnaissance automatique de la parole (ASR).la figure 3.13 illustre l'architecture de Wav2Vec 2.0 [22]

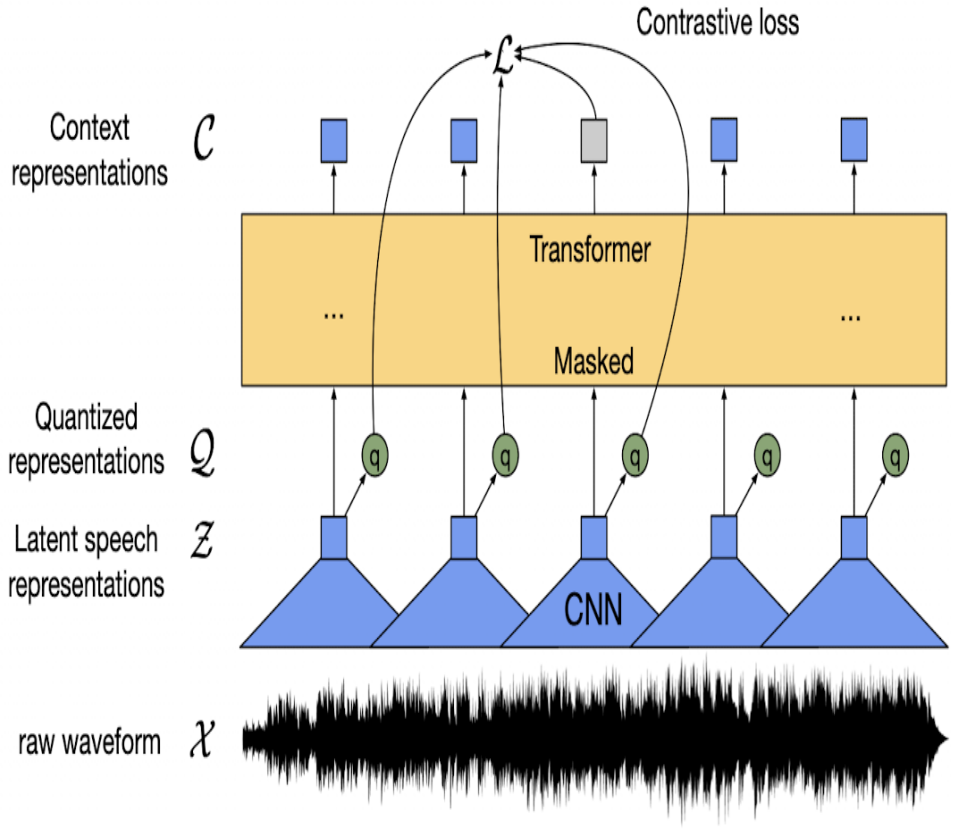


Figure 3.13 : Illustration de l'architecture de Wav2Vec 2.0[22].

Architecture de Wav2Vec 2.0

L'architecture de Wav2Vec 2.0 est constituée de divers éléments essentiels :

- **Extracteur de caractéristiques :** Les couches convolutives 1D de l'extracteur de caractéristiques initiales permettent de convertir les signaux audio bruts en représentations de caractéristiques plus compactes. Grâce à cette méthode de convolution, il est possible de diminuer la taille et de saisir les caractéristiques locales du signal audio.

- **Codage à l'aide de Transformer :**Plusieurs blocs de Transformateurs sont ensuite utilisés pour passer les caractéristiques extraites. Chaque composante des Transfor-

mateurs est équipée de mécanismes d'attention multi-tête et de couches de référence. Les Transformateurs ont pour mission de détecter les liens à long terme et les interactions contextuelles dans les séquences audio.

- **Préformation supervisée par soi-même :** Au cours de la pré-formation, le modèle est formé à anticiper des segments cachés de l'audio en se basant sur les contextes entourant. Le modèle est contraint par cette tâche contrastive d'acquérir des représentations solides et informatives des données audio non étiquetées.

- **Affinage supervisé :** Suite à la formation préliminaire, le modèle est ajusté en se basant sur des données étiquetées. Au-dessus des représentations apprises par les Transformateurs, on ajoute une couche de sortie aléatoirement initialisée afin de prédire des caractères ou des phonèmes [22].

3.3.4 Stockage des Réponses

Lorsque l'étudiant répond aux questions posées par le système, ses réponses doivent être stockées de manière fiable dans la base de données SQLite préalablement créée. Une fois que l'étudiant enregistre sa réponse audio, cette réponse est capturée et convertie en texte à l'aide du modèle de reconnaissance vocale Wav2Vec2. Le texte transcrit est ensuite préparé pour l'insertion dans la base de données. À chaque réponse, le système génère une requête SQL qui insère les informations pertinentes, telles que l'identifiant du sujet, l'identifiant de l'appareil, l'identifiant de la question, et la réponse textuelle dans la table dédiée de la base de données SQLite. Cette table a été conçue pour stocker chaque réponse de manière structurée, garantissant ainsi une organisation et un accès faciles aux données pour une évaluation ultérieure. Ce processus automatisé permet de centraliser toutes les réponses des étudiants de manière sécurisée et ordonnée.

3.4 Conception 3D

Le modèle 3D de l'appareil que nous avons créé, est représenté dans les figures ! 3.14 , 3.15 et 3.16 :

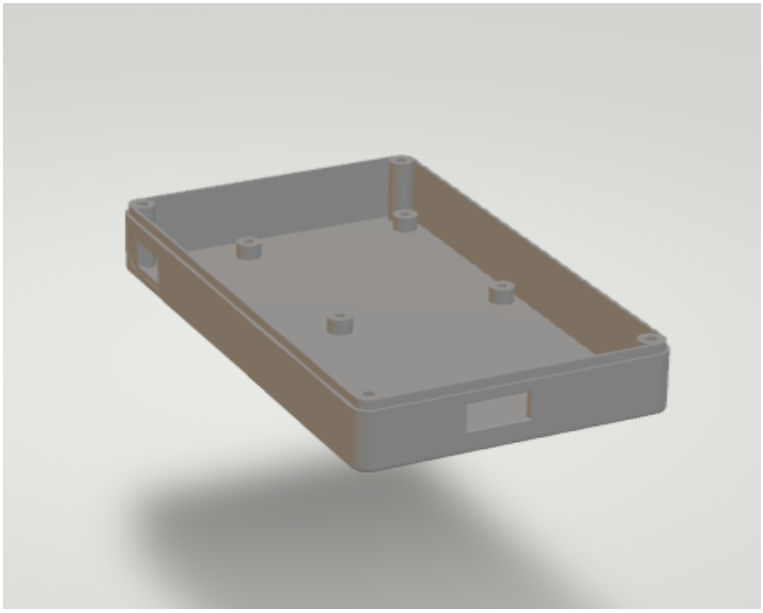


Figure 3.14 : Conception 3D.

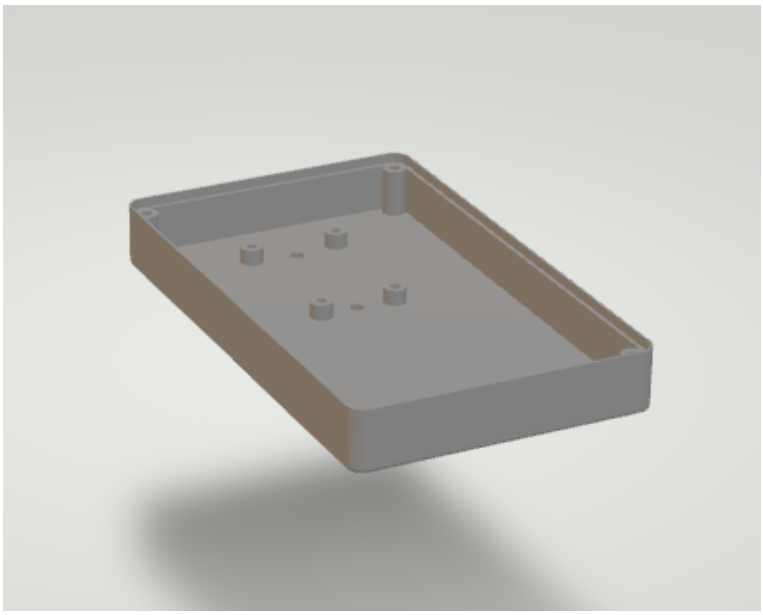


Figure 3.15 : Conception 3D (couvercle).

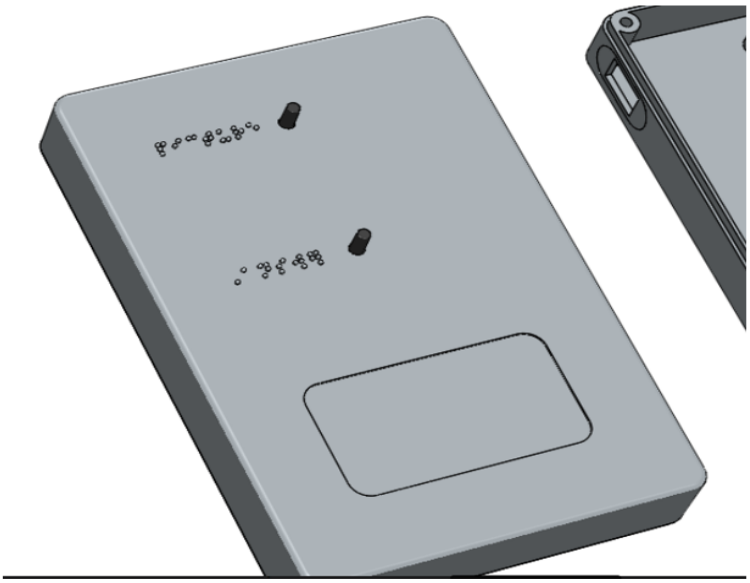


Figure 3.16 : Conception 3D (braille).

Conclusion

Dans ce chapitre, nous avons examiné en détail la création d'un système visant à rendre les examens plus accessibles aux étudiants aveugles. En examinant les éléments matériels, nous avons souligné l'importance capitale de la caméra dans la capture des sujets d'étude. En ce qui concerne le logiciel, on a développé une intégration approfondie de technologies telles que Pytesseract OCR pour la reconnaissance de texte et Google Text to Speech (gTTS) pour la conversion en audio. Des méthodes avancées basées sur le deep learning, comme LSTM, WaveNet et Wav2Vec2, ont également été étudiées afin d'améliorer la précision de la reconnaissance vocale et la génération de la parole. Tous ces éléments illustrent comment des technologies de pointe peuvent être intégrées de manière cohérente afin de concevoir un système efficace, répondant aux besoins particuliers des étudiants aveugles et proposant une solution novatrice pour faciliter l'accès aux examens.

Chapitre 4

implémentation et résultats obtenus

Introduction

Dans ce chapitre, nous allons explorer le développement et l'implémentation de notre système de reconnaissance vocale et de traitement OCR, en couvrant à la fois les aspects matériels et logiciels. Nous débuterons par une présentation des langages et outils de développement utilisés, qui constituent la base de notre projet. Ensuite, nous décrirons la réalisation matérielle, y compris les composants matériels essentiels tels que le Raspberry Pi, la webcam, et le casque microphone. Nous passerons ensuite à la réalisation logicielle, expliquant les étapes de configuration matérielle et l'implémentation des différents modèles de reconnaissance vocale, y compris LSTM, Wavenet, et Wav2Vec 2.0. Nous inclurons également les résultats des évaluations de ces modèles. Nous terminerons par une comparaison des performances des modèles et une explication détaillée du fonctionnement du serveur qui orchestre ces processus.

4.1 Langage et outils de developpement

Dans cette section, nous couvrirons les outils que nous avons utilisés dans notre système :

1. Python : est un langage de programmation orienté objet, à haut niveau, interprété et open-source, doté de semantics dynamiques et fournissant de nombreuses librairies de support. On utilise Python pour l'apprentissage automatique, l'analyse de données et même le design.[80]

2. Anaconda : Anaconda est une distribution scientifique de Python. Permet d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. Offert par anaconda Enterprise (gratuit), utilisé Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique.[81]

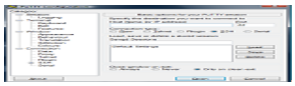
3. SQLite : est une bibliothèque en cours qui implémente un moteur de base de données SQL transactionnel autonome, sans serveur et sans configuration. Le code de SQLite est dans le domaine public et peut donc être utilisé gratuitement à des fins commerciales ou privées. SQLite est la base de données la plus largement déployée au monde avec plus d'applications que nous ne pouvons en compter, y compris plusieurs projets de grande envergure.[82]

4. BASH : (Bourne-Again SHell) est un interpréteur de commandes souvent utilisé sur les systèmes d'exploitation Linux et Mac OS. C'est une interface sous forme de ligne de commande qui permet aux utilisateurs d'exécuter des commandes système, des scripts et des programmes. [83]

5. logigramme : est un outil d'analyse qui permet de représenter de façon ordonnée et séquentielle l'ensemble des tâches ou évènements mis en oeuvre pour réaliser une activité donnée. Il est constitué d'un ensemble de symboles relié par des flèches.[84]

6. Blender : est un logiciel libre et open-source qui permet l'animation, la modélisation 3D et la création d'images 3D. Il comprend des fonctionnalités avancées telles que la sculpture 3D, le dépliage UV, le texturage, le rigging, l'armature, l'animation et le rendu. Blender est responsable de l'édition non linéaire, des compositions, de la création de matériaux, des applications interactives 3D et des simulations corporelles. Il fonctionne sur diverses plateformes et est basé sur Python. [85]

7. SOLIDWORKS : MBD 2021 étend la capacité d'organisation des données de produits et de fabrication (PMI) en vous offrant la possibilité d'ajouter des symboles de référence partielle à votre schéma de cotation DimXpert et en incluant des tables de zones de pliage dans les PDF 3D.[86]



8. PuTTY : est un client SSH et telnet, créé initialement par Simon Tatham pour la plateforme Windows. PuTTY est un logiciel libre accessible en source code, créé et soutenu par un groupe de volontaires.[87]

9. Fritzing : est un projet de révolution informatique open-source qui rend l'électronique accessible comme un moyen de création pour tout le monde. Nous proposons un logiciel, une plateforme de partage et des services dans le cadre de Processing et Arduino, favorisant ainsi un environnement créatif qui permet aux utilisateurs de documenter leurs prototypes, de les partager avec d'autres, de former des cours d'électronique en classe et de concevoir et fabriquer des PCBs de qualité.[88]

4.2 Réalisation matérielle

Les composants réels sont généralement connectés en fonction des connexions spécifiées dans le logiciel une fois que nous avons modélisé notre circuit dans Fritzing. Les schémas de circuits et les configurations de circuits imprimés sont très appréciés grâce à Fritzing, qui permet de visualiser les pièces et les connexions du circuit. Selon notre configuration spécifique, des fils, des câbles de démarrage , En suivant les connexions indiquées dans le schéma de circuit du logiciel Fritzing, il est possible de connecter soigneusement les broches ou bornes appropriées des composants afin de créer les connexions électriques nécessaires.

Le résultat physique de l'opération de connexion informatique précédemment mentionnée est illustré graphiquement dans la Figure "connexion". Il fournit une vision claire et approfondie du circuit achevé en mettant en évidence les véritables composants et leurs connexions. L'image offre un guide visuel qui permet une compréhension plus approfondie de la configuration physique et du disposition des équipements associés. On peut observer la position précise des pièces et leurs connexions correspondantes en observant la Figure (4.1)

Figure 4.1 : connexion du matériel.

4.3 Réalisation logiciel

4.3.1 Configuration Matérielle

Utilisant PuTTY, un logiciel de terminal, pour se connecter à un Raspberry Pi 4 via SSH, la configuration initiale implique plusieurs étapes clés pour préparer le Raspberry Pi à exécuter des scripts spécifiques :

1. **Connexion SSH :** Établir une connexion sécurisée avec le Raspberry Pi en utilisant PuTTY pour entrer des commandes directement sur le système d'exploitation Linux du Pi.

2. **Installation de Dépendances :** Installation des packages nécessaires comme gpio pour le contrôle des pins GPIO, nécessaires pour les interactions matérielles avec les boutons.

3. **Scripts Bash :** Création et modification de fichiers bash (”checkbtn.bash”, ”script.bash”, ”respond.bash”) pour définir des comportements spécifiques tels que la capture d’image, la réponse aux appuis sur les boutons, et la gestion des entrées audio.

Fonctionnement des Scripts

- **”checkbtn.bash”** : Ce script surveille les états des boutons connectés aux GPIOs. Il gère deux boutons : un pour prendre une photo et démarrer la séquence de questions, et l’autre pour enregistrer une réponse et préparer la question suivante(4.2)

```
if  [[ $(raspi-gpio get 14 | tail -c 23 | head -c 1) == 1 ]] ; then
if [[ $apressed == 0 ]] ; then
echo photo appuye
ffplay -autoexit beep.wav
./script.bash &
let apressed=1
fi
else
let apressed=0
fi

if  [[ $(raspi-gpio get 15 | tail -c 23 | head -c 1) == 1 ]] ; then
if [[ $bpressed == 0 ]]  then
echo reponse appuye
if [[ $isrecording == 0 ]] then
ffplay -autoexit -hide_banner -loglevel error answernow.wav &
arecord -f S16_LE -r 44100 --device="hw:4,0" -c 1 test.wav &
let isrecording=1
```

Figure 4.2 : checkbtn.bash

- **”script.bash” :** Utilisé pour prendre une photo via une webcam connectée et la transférer au serveur pour traitement OCR via HTTP .(Figure 4.3)

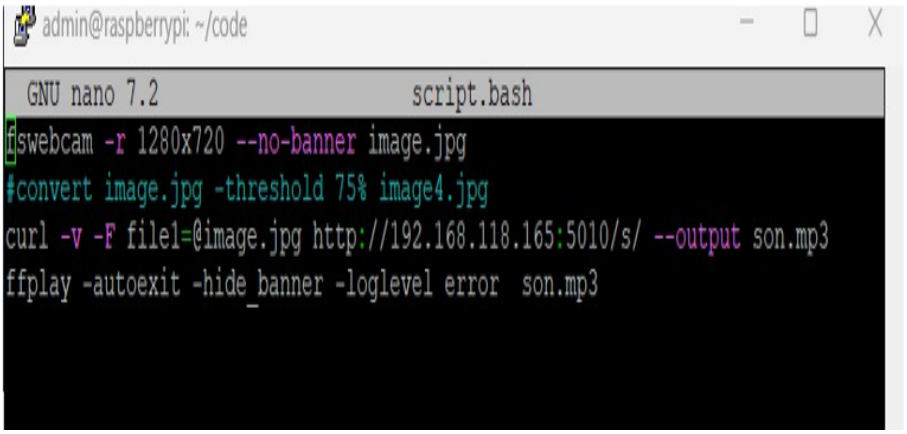


Figure 4.3 : script.bash

- **"respond.bash"** : Gère l'envoi des réponses audio, capturées via le microphone, au serveur pour conversion en texte.(4.4)

Figure 4.4 : respond.bash

4.3.2 Prétraitement de données

Le processus de prétraitement des données vise à produire un fichier manifeste JSON contenant des renseignements sur les fichiers audio et leurs transcriptions liées dans le jeu de données LibriSpeech. Pour débuter, il est nécessaire d'importer les modules 'os' et 'json', qui sont indispensables pour naviguer dans le système de fichiers et manipuler les données JSON, respectivement.

La fonction initiale, définie "load_transcriptions", utilise un fichier texte pour charger les transcriptions. Elle examine le fichier de transcription indiqué et examine chaque ligne, séparant ainsi l'identifiant du fichier audio de sa transcription. Le dictionnaire contient les identifiants des fichiers audio et leurs transcriptions, qui sont ensuite retournés. Cela facilite la mise en relation des fichiers audio avec leurs transcriptions correspondantes de manière facile. (Figure 4.5)

```python
def load_transcriptions(transcription_path):
    """ Load transcription from a file. """
    transcriptions = {}
    with open(transcription_path, 'r') as file:
        for line in file:
            parts = line.strip().split(maxsplit=1)
            file_id = parts[0]
            transcription = parts[1] if len(parts) > 1 else ''
            transcriptions[file_id] = transcription
    return transcriptions
```

Figure 4.5 : La fonction load_transcriptions

La fonction "generate_manifest_entries" effectue une recherche des fichiers audio et de leurs transcriptions dans la base de données en utilisant "os.walk". Ensuite, il découvre des fichiers de transcription et d'audio, charge les transcriptions pour chacun d'eux, extrait les

noms des fichiers audio, puis cherche les transcriptions dans le dictionnaire pour chaque fichier trouvé, ce qui donne lieu à une entrée dans le dictionnaire. (Figure 4.6)

```
def generate_manifest_entries(data_directory):
    """ Generator function to create manifest entries for audio files and their transcriptions. """
    for subdir, dirs, files in os.walk(data_directory):
        transcription_file = next((f for f in files if f.endswith('.trans.txt')), None)
```

Figure 4.6 : La fonction generate_manifest_entries

À partir des entrées générées, la fonction "create_manifest" génère un fichier manifeste JSON. Le fichier de sortie est ouvert en mode écriture et une expression génératrice est utilisée pour générer les entrées JSON pour chaque fichier audio et sa transcription. Par la suite, ces informations sont intégrées dans le fichier sous forme d'un tableau JSON, chaque entrée étant enregistrée sur une nouvelle ligne afin de faciliter la lecture et le traitement ultérieurs.(Figure 4.7)

```
def create_manifest(data_directory, output_file):
    """ Create a JSON manifest file from the directory of audio files and their transcriptions. """
    with open(output_file, 'w') as outfile:
        json_entries = (json.dumps(entry) for entry in generate_manifest_entries(data_directory))
```

Figure 4.7 : La fonction create_manifest

Enfin, le script principal établit le chemin de sortie du jeu de données et le nom du fichier manifeste, puis utilise ces paramètres pour appeler la fonction "create_manifest". Ce processus produit le fichier manifeste JSON qui renferme les chemins des fichiers audio ainsi que leurs transcriptions, prêt à être exploité dans des applications de traitement de la parole.

4.3.3 Implémentation des modèles

nous avons implémenté les modèles en utilisant anaconda `https://www.anaconda.com/download`

1- Modèle de reconnaissance vocale (LSTM)

Nous avons commencé par importer les bibliothèques requis, comme montré dans la figure 4.8.

```
import json
import librosa
import numpy as np
import logging
import math
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras import layers, models
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
```

Figure 4.8 : les bibliothèques nécessaires pour le modèle LSTM

json : Pour charger et analyser des fichiers JSON.

librosa : Pour le traitement audio et l'extraction de caractéristiques.

numpy : Pour les opérations numériques.

logging : Pour consigner les informations et les erreurs.

math : Pour les opérations mathématiques.

tensorflow.keras : Pour construire et entraîner le modèle de réseau neuronal.

Chargement du "manifest"

```
def load_manifest(filepath):
    try:
        with open(filepath, 'r') as f:
            data = json.load(f)
        return data
    except FileNotFoundError:
        logging.error(f"Manifest file not found: {filepath}")
        return []
    except json.JSONDecodeError:
        logging.error(f"Error decoding JSON from {filepath}")
        return []
```

Figure 4.9 : la fonction load manifest

L'objectif de cette fonction est de charger le fichier manifeste qui contient les chemins d'accès aux fichiers audio et à leurs transcriptions.

Extraction des caractéristiques

```python
def extract_features(data, max_length):
    X, y_raw = [], []
    for item in data:
        try:
            file_path = item['audio_filepath']
            transcription = item['transcription']
            audio, sr = librosa.load(file_path, sr=None)
            mfcc = librosa.feature.mfcc(y=audio, sr=sr, n_mfcc=13)
            padded_mfcc = pad_sequences([mfcc.T], maxlen=max_length, padding='post')[0]
            X.append(padded_mfcc)
            y_raw.append(transcription)
        except Exception as e:
            logging.error(f"Failed processing {file_path}: {str(e)}")
    return np.array(X), y_raw
```

Figure 4.10 : la fonction feature_extraction

Son objectif est d'extraire les caractéristiques MFCC des fichiers audio et les compléter à une longueur fixe.

Prétraitement 'batch'

```python
def preprocess_batch(X, y_raw, max_length):
    padded_mfcc = pad_sequences(X, maxlen=max_length, padding='post', dtype='float32')
    sequences = tokenizer.texts_to_sequences(y_raw)
    padded_sequences = pad_sequences(sequences, maxlen=max_length, padding='post')
    return np.array(padded_mfcc), padded_sequences
```

Figure 4.11 : la fonction preprocess_batch

L'objectif de cette fonction est de compléter les séquences MFCC et les transcriptions à une longueur fixe.

Construction et entraînement du modèle'

```python
# Build and train model
model = build_model(max_length, vocab_size)
train_gen = data_generator(filepath, batch_size=batch_size, max_length=max_length)
steps_per_epoch = math.ceil(len(all_data) / batch_size)

# Callbacks for model saving and early stopping
callbacks = [
    EarlyStopping(monitor='val_loss', patience=10, verbose=1),
    ModelCheckpoint('best_model.keras', monitor='val_loss', save_best_only=True, verbose=1)
]

# Assuming a validation generator is also defined similarly
model.fit(train_gen, epochs=100, steps_per_epoch=steps_per_epoch, callbacks=callbacks)
```

Figure 4.12 : Construction et entraînement du modèle

Elle a comme Objectif la Construction du modèle et la configuration du processus d'entraînement.

EarlyStopping Callback : Arrête l'entraînement si la perte de validation ne s'améliore pas après un certain nombre d'époques.

ModelCheckpoint Callback : Sauvegarde le modèle avec la meilleure perte de validation.

Évaluation des Résultats

L'évaluation des performances du modèle LSTM pour la reconnaissance vocale peut être résumée à partir des résultats d'entraînement. Le modèle est conçu avec plusieurs couches, dont des convolutions 1D, des normalisations par batch, des LSTM bidirectionnels, et une couche TimeDistributed pour traiter les séquences de caractéristiques MFCC.

Résultats de l'entraînement

L'entraînement du modèle a duré 100 époques, avec des lots de 32 tailles différentes. Les outils de rappel 'EarlyStopping' et 'ModelCheckpoint' ont été employés afin de mettre fin à l'entraînement si la perte de validation ne s'améliore pas et de sauvegarder le modèle le plus efficace basé sur la perte de validation. Toutefois, il semble que la validation de perte n'était pas accessible, ce qui a entravé l'utilisation adéquate de ces callbacks.

Les performances du modèle pendant l'entraînement sont les suivantes (Tableau 4.1)

Époque	**Précision**	**Perte**
1	0.2160	2.7598
2	0.2160	2.7598
3	0.2247	2.7044
4	0.2277	2.6851
...	...	...
100	0.3175	2.7362

Tableu 4.1 : Les performances du modèle LSTM.

Le tableau (4.1) montre que la précision stagne autour de 0.3 et la perte fluctue légèrement mais ne diminue pas de manière significative au cours des 100 époques. Cela

suggère que le modèle pourrait bénéficier de réglages supplémentaires, tels que l'ajustement des hyperparamètres, la modification de l'architecture du modèle, ou une meilleure gestion des données de validation.

En résumé, bien que le modèle ait été correctement construit et entraîné, ses performances indiquent qu'il y a des opportunités pour des améliorations afin d'atteindre une précision et une perte plus optimales pour la tâche de reconnaissance vocale.

2- Modèle de reconnaissance vocale (wavenet)

Nous avons commencé par importer les bibliothèques requis, comme montré à la Figure (4.13)

```
import json
import librosa
import numpy as np
import logging
import math
import matplotlib.pyplot as plt
import seaborn as sns
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras import layers, Model, Input
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, History, Callback
import noisereduce as nr
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score
```

Figure 4.13 : les bibliothèques nécessaires pour le modèle Wavenet.

- 'json' : Pour charger et analyser des fichiers JSON.

- 'librosa' : Pour le traitement audio et l'extraction de caractéristiques.

- 'numpy' : Pour les opérations numériques.

- 'logging' : Pour consigner les informations et les erreurs. - 'math' : Pour les opérations mathématiques.

- 'matplotlib.pyplot' : Pour tracer des graphiques et des visualisations.

- 'seaborn' : Pour des visualisations de données avancées.

- 'tensorflow.keras' : Pour construire et entraîner le modèle de réseau neuronal.

- 'noisereduce' : Pour réduire le bruit dans les données audio.

- 'sklearn' : Pour l'évaluation des modèles et les métriques.

Normalisation de l'audio

```python
def normalize_audio(audio):
    return (audio - np.mean(audio)) / np.std(audio)
```

Figure 4.14 : la fonction normalize_audio.

L'objectif de cette fonction : normaliser le signal audio pour avoir une moyenne nulle et une variance unitaire.

Extraction des caractéristiques

```python
def extract_features(data, max_length):
    X, y_raw = [], []
    for item in data:
        try:
            file_path = item['audio_filepath']
            transcription = item['transcription']
            audio, sr = librosa.load(file_path, sr=None)
```

Figure 4.15 : la fonction feature_extraction.

Objectif : Extraire les caractéristiques MFCC des fichiers audio, appliquer une réduction de bruit, normaliser l'audio et compléter les caractéristiques MFCC à une longueur fixe.

Construction du modèle WaveNet

```python
def build_wavenet_model(input_shape, num_blocks=3, num_residual_layers=3, num_filters=64):
    inputs = Input(shape=input_shape)
    skip_connections = []
    x = inputs
    for i in range(num_residual_layers):
        dilation_rate = 2 ** i
        for _ in range(num_blocks):
            x, skip = residual_block(x, num_filters, dilation_rate)
            skip_connections.append(skip)
    x = layers.Add()(skip_connections)
    x = layers.Activation('relu')(x)
    x = layers.Conv1D(128, kernel_size=1, activation='relu')(x)
    outputs = layers.Conv1D(vocab_size, kernel_size=1, activation='softmax')(x)
    model = Model(inputs, outputs)
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    model.summary()
    return model
```

Figure 4.16 : Construction du modèle WaveNet..

L'objectif de cette fonction :construire un modèle WaveNet pour la reconnaissance vocale en utilisant des blocs résiduels, des connexions de saut et des couches de convolution 1D.

Objectif de chaque couche dans le modèle WaveNet

1. **Input Layer :** Définit la forme des données d'entrée
2. **Residual Blocks :** Capturent les dépendances temporelles à plusieurs échelles en utilisant des convolutions dilatées.
3. **Skip Connections :** Combinent les caractéristiques de tous les blocs résiduels, préservant l'information et améliorant le flux de gradients.
4. **Activation Layer :** Introduit la non-linéarité aux caractéristiques combinées.
5. **1D Convolutional Layer :** Traite davantage les caractéristiques combinées.
6. **Output Layer :** Produit les prédictions finales en utilisant une fonction d'activation softmax.

Évaluation des Résultats

Le modèle WaveNet, conçu pour la reconnaissance vocale, utilise des blocs résiduels, des connexions de saut et des couches de convolution 1D pour capturer les dépendances à long terme dans les données séquentielles. Les résultats d'entraînement montrent les performances du modèle en termes de précision, de rappel, de score F1 et de perte de validation sur 10 époques.

Résultats de l'Entraînement

Les performances du modèle pendant l'entraînement sont résumées dans le tableau suivant (tableau 4.2)

Époque	**Précision**	**Perte de validation**
1	0.1768	2.8778
2	0.1776	2.8767
3	0.1776	2.8761
4	0.1776	2.8753
5	0.1776	2.8749
6	0.1777	2.8749
7	0.1777	2.8749
8	0.1776	2.8745
9	0.1777	2.8745
10	0.1777	2.8739

Tableu 4.2 : Les performances du modèle Wavenet.

Selon le tableau (4.2), on observe une stabilité dans la précision du modèle tout au long des époques, avec une légère amélioration de la perte de validation. Cela laisse entendre que le modèle est stable mais pourrait être amélioré pour améliorer la précision.L'absence de disponibilité a entravé l'utilisation adéquate de ces callbacks.

Le graphique de la perte de validation (Figure 4.17)montre une diminution constante au fil des époques d'entraînement.

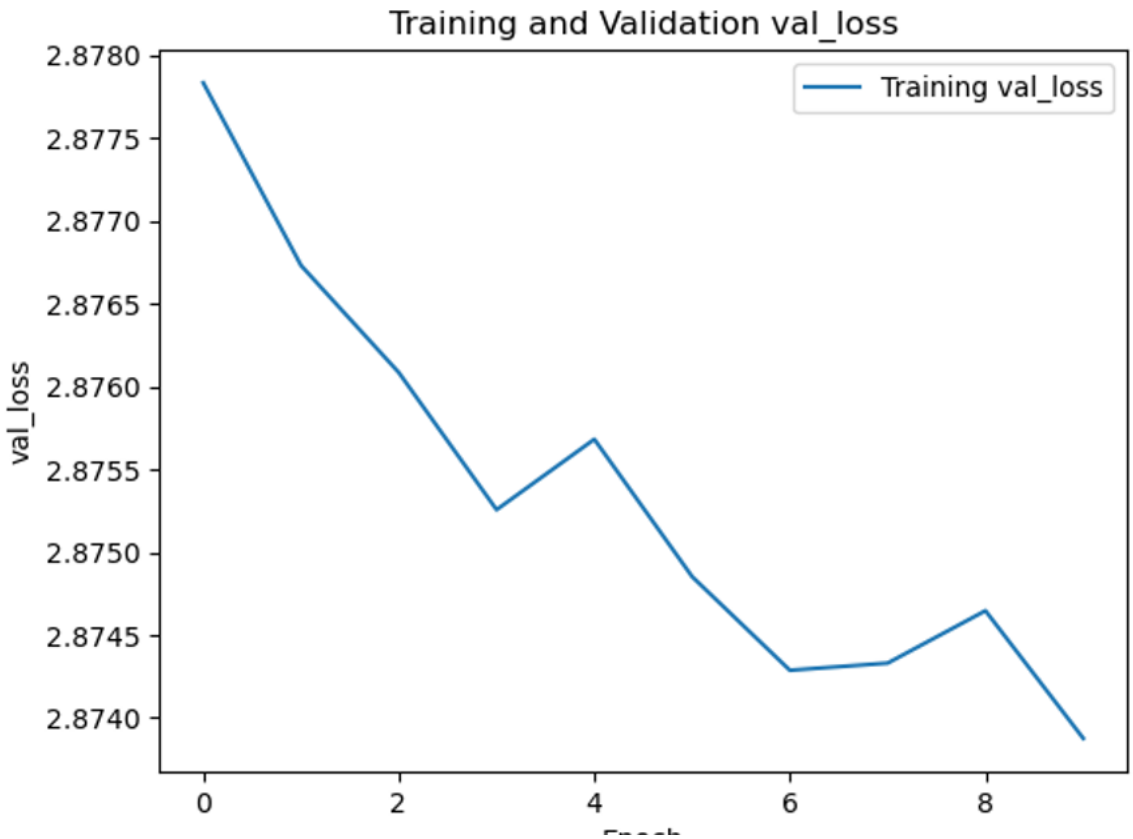


Figure 4.17 : perte de validation.

La perte de validation diminue régulièrement, indiquant que le modèle s'améliore et généralise mieux sur les données de validation. Cependant, la diminution est relativement faible, suggérant que des ajustements supplémentaires, tels que l'optimisation des hyperparamètres ou des techniques de régularisation supplémentaires, pourraient être nécessaires pour des gains de performance plus significatifs.

Le graphique de la précision(figure 4.18) montre une légère amélioration de la précision au fil des époques.

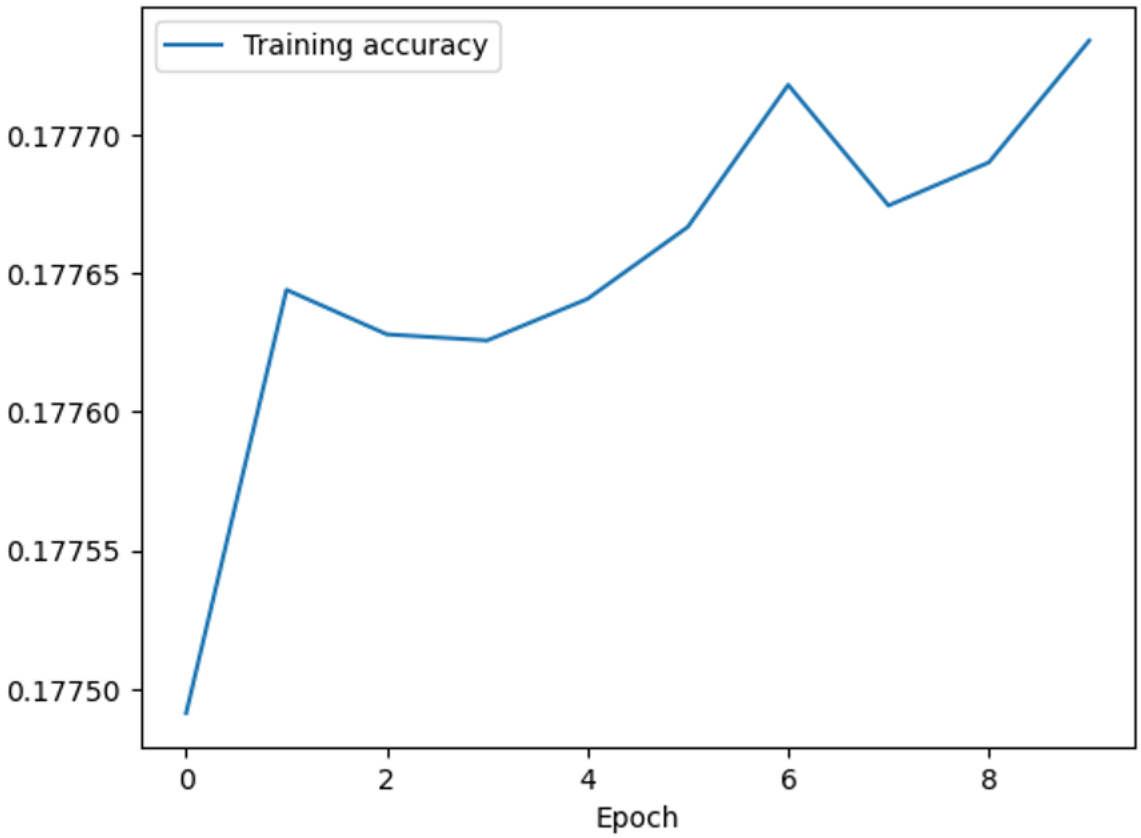


Figure 4.18 : la précision

La précision affiche une tendance à la hausse, bien que modeste. Cela montre que le modèle améliore lentement sa capacité de prédiction correcte. Une amélioration constante mais faible peut indiquer que le modèle pourrait bénéficier de modifications de son architecture ou d'une augmentation de la taille des données d'entraînement pour améliorer davantage sa performance.

En résumé, bien que le modèle WaveNet montre des performances stables, il existe des opportunités pour des ajustements supplémentaires pour améliorer sa précision et réduire davantage la perte de validation. L'architecture utilisée, avec des convolutions dilatées et des connexions résiduelles, est efficace pour capturer les dépendances à long terme dans les données séquentielles, ce qui est crucial pour la reconnaissance vocale.

3- Le modèle Wav2Vec 2.0

Wav2Vec 2.0 est un cadre d'apprentissage auto-supervisé pour les représentations de la parole. [22]

- **Jeux de Données :**
  - **Données non étiquetées :** Librispeech (960 heures) et LibriVox (53 200 heures).
  - **Données étiquetées :** Subsets de Librispeech étiquetées (960 heures), subsets de Libri-light (10h, 1h, 10min), et TIMIT pour la reconnaissance de phonèmes.
- **Pré-entraînement :**
  - **Encodage des Caractéristiques :** Sept blocs de convolutions temporelles avec 512 canaux chacun.
  - **Masquage :** 49% des étapes temporelles sont masquées avec une longueur de span moyenne de 299 ms.
  - **Modèles :** Deux configurations (Base et Large) avec des différences dans la configuration du Transformer.
- **Affinage :**
  - **Librispeech/Libri-light :** Affinage avec une couche de sortie aléatoirement initialisée pour prédire des caractères ou des phonèmes
  - **Protocole d'évaluation :** Évaluation sur des ensembles de développement et de test standard de Librispeech[22]

Résultats

1. **Précision :**
   - **Modèle Base :** Comprend 12 blocs de Transformer avec une dimension de modèle de 768, et 8 têtes d'attention. L'entraînement est effectué sur 64 GPU V100 pendant 1,6 jour.
   - **Modèle Large :** Comprend 24 blocs de Transformer avec une dimension de modèle de 1024 et 16 têtes d'attention. L'entraînement est effectué sur 128 GPU V100 pendant 2,3 jours pour Librispeech et 5,2 jours pour LibriVox.
2. **Taux d'erreur de mot (WER) :**

- **Librispeech :** Le WER sur l'ensemble de test « clean » et « other » montre une réduction significative par rapport aux méthodes semi-supervisées traditionnelles.
- **Libri-light :** WER réduit sur des sous-ensembles de données limitées, démontrant l'efficacité de Wav2Vec 2.0 dans des scénarios de faible ressource. .[22]

Analyse

- **Apprentissage Auto-supervisé :** L'utilisation de grandes quantités de données non étiquetées pour pré-entraîner le modèle permet de capturer des représentations riches et généralisables de la parole.
- **Affinage Efficace :** L'affinage avec des quantités limitées de données étiquetées montre que Wav2Vec 2.0 peut atteindre des performances de pointe même avec des données d'entraînement limitées.
- **Architecture :** Les améliorations de l'architecture du Transformer et des techniques de masquage contribuent à la robustesse et à la performance du modèle.[22]

Wav2Vec 2.0 montre une avancée significative dans l'apprentissage des représentations de la parole avec un apprentissage auto-supervisé, offrant des performances supérieures dans des tâches de reconnaissance vocale avec des ressources limitées. Cette approche réduit le besoin en données étiquetées tout en améliorant la précision des modèles de reconnaissance vocale.[22]

4.3.4 Tableau de Comparaison

Le tableau 4.3 montre une comparaison entre : LSTM, WaveNet, et Wav2Vec 2.0

Critère	**LSTM**	**WaveNet**	**Wav2Vec 2.0**[22]
Architecture	Réseau de mémoire à long terme (LSTM)	Convolutions dilatées et résiduelles	Transformer avec masquage et quantification des représentations latentes
Entrées	Séquences de caractéristiques (MFCC)	Séquences de caractéristiques (MFCC)	Signaux audio bruts
Précision	Environ 0.22	Environ 0.18	WER réduit significativement sur Librispeech et Libri-ligh
Perte de Validation	Environ 2.74	Environ 2.87	Réduction notable de la perte de validation
Complexité	Élevée (GPU)	Élevée (GPU)	Très élevée (GPU)
Nombre d'époques	100	10	Variable (souvent 100 à 400)
Temps d'entraînement	11 jours	5 jours	Variable (souvent plusieurs jours sur GPU)

Tableu 4.3 : comparaison entre : LSTM, WaveNet, et Wav2Vec 2.0.

- **LSTM :** Utilisé pour capturer les dépendances temporelles avec une complexité élevée nécessitant des ressources GPU substantielles. Précision d'environ 0.22 avec une perte de validation autour de 2.74. Entraîné pendant 100 époques en 11 jours.
- **WaveNet :** Capte efficacement les dépendances à long terme grâce aux convolutions dilatées et résiduelles. Précision d'environ 0.18 avec une perte de validation autour de 2.87. Exécuté sur GPU et entraîné pendant 10 époques en 5 jours.
- **Wav2Vec 2.0 :** Utilise un pré-entraînement auto-supervisé pour apprendre des représentations riches. Affine ensuite ces représentations avec des données étiquetées, ce qui réduit significativement le WER. Nécessite des ressources substantielles pour l'entraînement sur GPU. Le temps d'entraînement et le nombre d'époques peuvent varier mais sont souvent de plusieurs jours.

4.3.5 Fonctionnement du Serveur

Le serveur que nous avons configuré utilise principalement le micro-framework Flask pour gérer les interactions web et l'automatisation de certains processus audio et de reconnaissance d'image/texte. Voici un aperçu détaillé de son fonctionnement, basé sur les extraits de code fournis :

Configuration Initiale

1. **Initialisation de l'application Flask :** Nous initialisons une application Flask, qui est un cadre léger pour développer des applications web en "Python.app = Flask(__name__)" sert à créer une instance de l'application

2. **Configuration du Dossier de Téléchargement :** Nous définissons un dossier où les fichiers téléchargés seront stockés avec "app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER".

Routes

1. **Route /s/ :** Cette route gère le téléchargement de fichiers. Si un fichier est posté, il est enregistré dans le dossier spécifié, puis traité par Tesseract OCR pour extraire du texte à partir d'une image. Le texte est ensuite vérifié pour les questions, qui, si trouvées, sont converties en parole (TTS) et renvoyées à l'utilisateur.

2. **Routes /q/ et /song/ :** Ces routes servent à renvoyer des fichiers médias statiques aux utilisateurs, probablement pour des tests ou des fonctionnalités spécifiques non décrites explicitement dans les extraits.

3. **Route /stt/ (Speech to Text) :** Cette route prend un fichier audio téléchargé, utilise les modèles Wav2Vec2 pour la transcription automatique du discours en texte. Cela pourrait être utilisé pour transcrire des réponses audio en texte.

4. **Route /r/ :** Similaire à /s/, mais semble être configurée pour traiter les réponses en utilisant des identifiants de session ou de question spécifiques pour suivre les interactions.

Base de Données

1. **SQLite3 :** Nous utilisons SQLite pour enregistrer les réponses dans une base de données locale. Nous créons une table si elle n'existe pas déjà et y insérons les réponses des utilisateurs après leur transcription.

Conclusion

Ce chapitre nous a permis de couvrir de manière exhaustive les différentes étapes et composantes nécessaires à la mise en place de notre système de reconnaissance vocale et de traitement OCR. Nous avons détaillé les langages et outils de développement utilisés, et comment ils s'intègrent dans la réalisation matérielle et logicielle du projet. La configuration matérielle a été soigneusement décrite pour garantir une compréhension claire des besoins du système. Nous avons exploré et évalué les performances de plusieurs modèles de reconnaissance vocale, y compris LSTM, Wavenet, et Wav2Vec 2.0, et comparé leurs résultats pour déterminer le plus efficace pour notre application.Enfin, le fonctionnement du serveur, qui constitue le noyau de notre système, a été expliqué en détail, mettant en lumière la manière dont les différentes composantes interagissent pour fournir une solution cohérente et efficace.

Conclusion

Ce projet a examiné les nombreux obstacles rencontrés par les étudiants aveugles, particulièrement lors des examens, en soulignant les insuffisances des méthodes traditionnelles basées sur des assistants humains. Ces méthodes ont souvent montré leurs limites en termes d'accessibilité et d'équité. Pour surmonter ces défis, un appareil intelligent combinant des technologies de reconnaissance optique de caractères (OCR) et de synthèse vocale (gTTS), ainsi que des modèles d'apprentissage profond, a été conçu. Cet appareil permet aux étudiants aveugles de passer leurs examens de façon autonome, améliorant ainsi leur expérience académique sans nécessiter d'assistance humaine pour la lecture des questions et l'écriture des réponses. Les exigences de recherche comprenaient l'utilisation de l'Internet des objets (IoT) pour une communication efficace et en temps réel, ainsi que l'intégration de techniques avancées d'intelligence artificielle (IA) pour garantir une reconnaissance vocale et textuelle de haute précision. L'ergonomie et l'accessibilité ont également été des priorités dans la conception de l'appareil, assurant une utilisation facile pour les étudiants aveugles tout en protégeant la sécurité et la confidentialité des données. La structure de ce mémoire a permis de présenter de manière systématique les différentes étapes du projet, de la conception à l'évaluation des résultats. Le Chapitre 1 a examiné les défis quotidiens des personnes aveugles et les solutions offertes par l'IA. Le Chapitre 2 s'est penché sur les techniques de traitement du langage naturel et les technologies de reconnaissance. Le Chapitre 3 a décrit en détail les contributions matérielles et logicielles du projet. Enfin, le Chapitre 4 a présenté l'implémentation pratique, les résultats obtenus, et un tableau comparatif des différentes approches.

Afin de rendre notre appareil intelligent destiné aux étudiants aveugles plus utile, nous prévoyons plusieurs améliorations majeures. Tout d'abord, avant le début de l'examen,

l'appareil collectera des informations personnelles de l'étudiant pour une personnalisation accrue. Après la prise d'une photo de l'examen et avant d'utiliser l'OCR, nous souhaitons amender la qualité d'image obtenue (a cause des problèmes rencontrés avec OCR). Ensuite, nous ajouterons deux autres boutons permettant de naviguer entre les questions et de revenir en arrière pour modifier les réponses, offrant ainsi une plus grande flexibilité et autonomie pendant les examens. Une fonctionnalité de répétition des questions sera également intégrée, améliorant la compréhension et réduisant les erreurs. Nous prévoyons également de prendre en charge plusieurs langues, y compris des expressions mathématiques et des descriptions d'images présentes dans les sujets d'examen. Cette fonctionnalité visera à répondre aux besoins divers des utilisateurs et à offrir une assistance complète dans différents contextes académiques. De plus, nous créerons une plateforme destinée aux enseignants, simplifiant la gestion des examens, la communication et le suivi des performances. Ce portail centralisé deviendra un point de convergence pour toutes les interactions académiques entre enseignants, optimisant ainsi l'organisation et la coopération au sein des institutions éducatives. Enfin, nous nous engageons à améliorer continuellement la précision de nos modèles d'apprentissage profond. En affinant les algorithmes et en enrichissant les jeux de données, nous visons à offrir une reconnaissance vocale et de texte encore plus fiable et précise. Ces perspectives futures visent à transformer notre appareil intelligent en une solution complète et adaptable, capable de répondre aux besoins évolutifs des étudiants aveugles et d'assurer une expérience d'examen équitable et efficace.

Bibliographie

[1] "Tout comprendre : Ia, machine learning, deep learning." https://www.sales-hacking.com/post/intelligence-artificielle-vs-machine-learning-vs-deep-learning. 2024.Consulté le 10 mai 2024.

[2] Linedata, "Schema-wide-fr [web image]." https://fr.linedata.com/sites/default/files/styles/max_1300x1300/public/inline-images/Schema-wide-fr.webp?itok=xI5LzheR. n.d. Consulté le 21 mai 2024.

[3] Groupe de Travail "Usage des Données en Vie Réelle dans la Prise de Décision", "L'apprentissage non-supervisé." https:gt2.ariis.fr/les-algorithmes-dexploitation/lapprentissage-non-supervise. n.d.Consulté le 14 mai 2024.

[4] DataScientest, "Plan de travail 2 [image]." https://datascientest.com/wp-content/uploads/2020/07/Plan-de-travail-2.png. 2020 ,Consulté le 21 mai 2024.

[5] H. Maâmatou, *Apprentissage semi-supervisé pour la détection multi-objets dans des séquences vidéos : Application à l'analyse de flux urbains.* PhD thesis, Université Clermont Auvergne [2017-2020] ; Université de Sfax (Tunisie), 2017.

[6] "Machine learning vs deep learning." https://culturesciencesphysique.ens-lyon.fr/images/articles/ia-rousseau/fig-machine-vs-deep.png, n.d. Consulté le 14 mai 2024.

[7] "Introduction au nlp (natural language processing)." https://datascientest.com/introduction-au-nlp-natural-language-processing. n.d.Consulté le 2 mai 2024.

[8] "Traitement langage naturel." https://ledigitalizeur.fr/ia/traitement-langage-naturel. n.d.Consulté le 11 mai 2024.

[9] "architecture-iot.webp." https://iotindustriel.com/wp-content/uploads/2022/01/architecture-iot.webp. 2022.Consulté le 21 mai 2024.

[10] Robotics Backend, "Install raspbian desktop on a virtual machine (virtualbox)." https://roboticsbackend.com/install-raspbian-desktop-on-a-virtual-machine-virtualbox/. n.d.Consulté le 30 mai 2024.

[11] David watson, "What is raspberry pi 4 ? pinout, specs, projects and datasheet." https://images.theengineeringprojects.com/image/webp/2021/03/raspberry-pi-4.png.webp?ssl=1, March 2021. Consulté le 22 mai 2024.

[12] "Raspberry pi 4 model b." https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. n.d.Consulté le 23 mai 2024 sur.

[13] "Raspberry pi 3 model b+." https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/. n.d .Consulté le 23 mai 2024.

[14] "Raspberry pi 3 model b." https://www.raspberrypi.com/products/raspberry-pi-3-model-b/. n.d.Consulté le 23 mai 2024.

[15] "Raspberry pi 1 model b+." https://www.raspberrypi.com/products/raspberry-pi-1-model-b-plus/. n.d.Consulté le 23 mai 2024 sur.

[16] J. Geerling, "Look inside the raspberry pi zero 2 w and the rp3a0-au." `https://www.jeffgeerling.com/blog/2021/look-inside-raspberry-pi-zero-2-w-and-rp3a0-au`. 2021.Consulté le 23 mai 2024.

[17] "Raspberry pi hardware." `https://www.raspberrypi.com/documentation/computers/raspberry-pi.html`. n.d.Consulté le 12 mai 2024.

[18] "Raspberry pi model a+ launched today." `https://raspi.tv/2014/raspberry-pi-model-a-launched-today`. n.d.Consulté le 2 mai 2024.

[19] S. Algun, "Review for tesseract and kraken ocr for text recognition." `https://medium.datadriveninvestor.com/review-for-tesseract-and-kraken-ocr-for-text-recognition-2e63c2adedd0`. n.d.Consulté le 12 mai 2024.

[20] "Long short-term memory networks (lstm)- simply explained !." `https://databasecamp.de/en/ml/lstms`. n.d. Consulté le 21 mai 2024.

[21] M. Vafaie, "Dialect normalisation with deep learning-based automatic speech recognition," *Saarbrücken : University of Saarland MSc thesis*, 2017.

[22] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0 : A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[23] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition : a survey," *Multimedia Tools and Applications*, vol. 80, pp. 9411–9457, 2021.

[24] A. E. Patla, "Understanding the roles of vision in the control of human locomotion," *Gait & posture*, vol. 5, no. 1, pp. 54–69, 1997.

[25] G. Montagné, *L'inclusion des personnes aveugles et malvoyantes dans le monde d'aujourd'hui*. Ministère du travail, des relations sociales et de la solidarité, 2007.

[26] J. Heyraud, "La déficience visuelle, une atteinte complexe qui touche tous les âges," in *L'accompagnement au quotidien des personnes déficientes visuelles*, pp. 27–45, Érès, 2013.

[27] S. S. Administration, "Disability evaluation under social security." `https://www.ssa.gov/disability/professionals/bluebook/2.00-SpecialSensesandSpeech-Adult.htm`. 2018.Consulté le 5 mai 2024.

[28] I. Audo, J. A. Sahel, S. Bhattacharya, and C. Zeitz, "Trpm1, un nouveau gène impliqué dans la cécité nocturne congénitale stationnaire," *médecine/sciences*, vol. 26, no. 3, pp. 241–244, 2010.

[29] Organisation mondiale de la Santé, "Cécité et déficience visuelle." `https://www.who.int/fr/news-room/fact-sheets/detail/blindness-and-visual-impairment`. 2019.Consulté le 5 mai 2024.

[30] S. Chokron, "La cécité corticale : sémiologie, étiologie et perspectives de prise en charge neuropsychologique," *Revue de neuropsychologie*, no. 1, pp. 38–44, 2013.

[31] A. R. Galiano, S. Portalier, N. Baltenneck, M. Griot, and M. Poussin, "Étude pragmatique des compétences référentielles des personnes aveugles," *Bulletin de psychologie*, vol. 518, no. 2, pp. 129–139, 2012.

[32] N. Baltenneck, S. Portalier, P.-M. Chapon, and F. Osiurak, "Parcourir la ville sans voir : effet de l'environnement urbain sur la perception et le ressenti des personnes aveugles lors d'un déplacement in situ," *L'année psychologique*, no. 3, pp. 403–433, 2012.

[33] W. Ertel, *Introduction to artificial intelligence*. Springer, 2018.

[34] S. Russell and P. Norvig, *Artificial Intelligence : A Modern Approach*. Hoboken, NJ, USA : Pearson, 4th ed., 2021.

[35] R. Demichelis, "Comment l'intelligence artificielle va révolutionner la vie des malvoyants." `https://www.lesechos.fr/tech-medias/intelligence-artificielle`. 2019. Consulté le 21 mai 2024 sur.

[36] A. F. for the Blind, "Technologies d'assistance pour les déficients visuels." `https://www.afb.org/blindness-and-low-vision/using-technology`. 2023. Consulté le 21 mai 2024 sur.

[37] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[39] P. E. Ludwig, V. Reddy, and M. Varacallo, "Neuroanatomy, neurons," in *StatPearls [Internet]*, StatPearls Publishing, 2023.

[40] S.-H. Han, K. W. Kim, S. Kim, and Y. C. Youn, "Artificial neural network : understanding the basic concepts without mathematics," *Dementia and neurocognitive disorders*, vol. 17, no. 3, p. 83, 2018.

[41] M. Nohair, A. St-Hilaire, and T. B. Ouarda, "Utilisation des réseaux de neurones et de la régularisation bayésienne en modélisation de la température de l'eau en rivière," *Revue des sciences de l'eau*, vol. 21, no. 3, pp. 373–382, 2008.

[42] S. Haykin, *Neural networks and learning machines, 3/E*. Pearson Education India, 2009.

[43] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[45] O. Chapelle, B. Schölkopf, and A. Zien, eds., *Semi-Supervised Learning*. MIT Press, 2006.

[46] X. Zhu, "title=Deep learning, author=Goodfellow, Ian and Bengio, Yoshua and Courville, Aaron, year=2016, publisher=MIT press," Tech. Rep. 3, Computer Science, University of Wisconsin-Madison, 2005.

[47] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[48] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[49] C. Staff, "Deep lear,vs machine learning : A beginners guide." `https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide`. 2024.Consulté le 21 mai 2024 sur.

[50] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.

[51] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[52] "Natural language processing." `https://www.ibm.com/fr-fr/topics/natural-language-processing`. n.d.Consulté le 21 mai 2024.

[53] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert : Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv :1810.04805*, 2018.

[54] "Qu'est-ce que la reconnaissance optique de caractères (ocr) ?." `https://aws.amazon.com/fr/what-is/ocr/`. n.d. Consulté le 21 mai 2024.

[55] J. Poignant, "Détection et reconnaissance de texte dans les documents vidéos - et leurs apports à la reconnaissance de personnes," in *CORIA 2011 - COnférence en Recherche d'Information et Applications - 6e Rencontres Jeunes Chercheurs en Recherche d'Information (RJCRI)*, (Avignon, France), pp. 409–414, 2011.

[56] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, and H. Lee, "What is wrong with scene text recognition model comparisons ? dataset and model analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4715–4723, 2020.

[57] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, and C. Xiao, "Scene text recognition from two-dimensional perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8714–8721, 2019.

[58] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans : Autonomous, bidirectional and iterative language modeling for scene text recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7098–7107, 2021.

[59] S. Walter, S. Scherer, M. Schels, M. Glodek, D. Hrabal, M. Schmidt, R. Böck, K. Limbrecht, H. C. Traue, and F. Schwenker, "Multimodal emotion classification in naturalistic user behavior," in *Human-Computer Interaction. Towards Mobile and Intelligent Interaction Environments : 14th International Conference, HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part III 14*, pp. 603–611, Springer, 2011.

[60] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, Ieee, 2013.

[61] M. Anusuya and S. K. Katti, "Speech recognition by machine, a review," *arXiv preprint arXiv :1001.2267*, 2010.

[62] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014.

[63] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, *et al.*, "Conformer : Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv :2005.08100*, 2020.

[64] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, *et al.*, "Deep speech : Scaling up end-to-end speech recognition," *arXiv preprint arXiv :1412.5567*, 2014.

[65] A. Whitmore, A. Agarwal, and L. Da Xu, "The internet of things—a survey of topics and trends," *Information systems frontiers*, vol. 17, pp. 261–274, 2015.

[66] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[67] A. Kamilaris and A. Pitsillides, "Mobile phone computing and the internet of things : A survey," *IEEE Internet of things Journal*, vol. 3, no. 6, pp. 885–898, 2016.

[68] J. Saadi, L. El Echi, W. Boulila, and K. Ghedira, "Towards a multi agent based iot middleware for smart environment monitoring," *Applied Sciences*, vol. 10, no. 13, p. 4541, 2020.

[69] S. Eckert, *Raspberry Pi : Exploiting the Compute Module 4*. Elektor International Media BV, 2021.

[70] G. Saaters, *Raspberry Pi IoT Projects : Prototyping Experiments for Makers*. Apress, 2021.

[71] "What is a raspberry pi ?." `https://www.raspberrypi.com/products/`.

[72] "Raspberry pi 2 model b." `https://www.raspberrypi.com/products/raspberry-pi-2-model-b/`, 2015. n.d.Consulté le 23 mai 2024.

[73] "Raspberry pi zero w." `https://www.raspberrypi.com/products/raspberry-pi-zero-w/`. n.d.Consulté le 12 mai 2024.

[74] "fritzing." `https://fritzing.org/`. n.d.Consulté le 12 mai 2024.

[75] "Tesseract ocr." `https://doc.ubuntu-fr.org/tesseract-ocr`. n.d.Consulté le 12 mai 2024.

[76] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2, pp. 629–633, IEEE, 2007.

[77] A. J. Sarmah, K. Bhagawati, K. Duwarah, S. D. Purkayastha, A. Boro, and D. Muchahary, "Object detection and conversion of text to speech for visually impaired," *ADBU Journal of Engineering Technology*, vol. 12, no. 2, 2023.

[78] C.-L. Hung, "Deep learning in biomedical informatics," in *Intelligent Nanotechnology*, pp. 307–329, Elsevier, 2023.

[79] F. Yen and Z. Katrib, "Wavenet based autoencoder model : Vibration analysis on centrifugal pump for degradation estimation.," in *Annual Conference of the PHM Society*, vol. 12, pp. 6–6, 2020.

[80] "python." https://www.python.org/. Consulté le 1 mai 2024.

[81] "anaconda." https://www.anaconda.com/about-us. Consulté le 1 mai 2024.

[82] "sqlite." https://sqlite.org/about.html. Consulté le 1 mai 2024.

[83] "bash." https://www.data-bird.co/blog/bash-script#:~:text=C'est%20quoi%20le%20Bash,des%20scripts%20et%20des%20programmes. Consulté le 1 mai 2024.

[84] "logigramme." https://www.xl-consultants.com/ressources/glossaire/logigramme-de-processus#:~:text=Le%20logigramme%20est%20un%20outil,symboles%20reli%C3%A9%20par%20des%20fl%C3%A8ches*. Consulté le 7 mai 2024.

[85] "Blender." https://www.techno-science.net/glossaire-definition/Blender.html. Consulté le 10 mai 2024.

[86] "solidworks." https://www.solidworks.com/fr/media/solidworks-2021-model-based-definition. Consulté le 10 mai 2024.

[87] "putty." https://www.putty.org/. Consulté le 10 mai 2024.

[88] "frizting." https://fritzing.org/projects/raspberry-pi-3. Consulté le 1 juine 2024.