الجمهورية الجزائرية الديمقراطيةالشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Filière Électronique
Spécialité Instrumentation

Présenté par :

BARA Sabrine

&

MORSLI Mohamed Ridha

Reconnaissance automatique du locuteur à l'aide des réseaux de neurones convolutifs

Proposé par : YKHLEF Farid

Année Universitaire 2022-2023

Remerciements

Nous tenons à exprimer nos sincères remerciements au Dieu tout-puissant, qui nous a accordé la volonté, la santé et le courage nécessaires pour mener à bien ce travail.

Nous tenons également à exprimer notre gratitude envers notre encadrant ***Mr. F. YKHLEF*** *pour sa confiance en notre capacité à mener ce projet à terme, à ses conseils éclairés, son expertise et son soutien constant ont été essentiels pour nous guider dans la bonne direction.*

Nous tenons également à exprimer notre gratitude envers les membres du jury, ***Mr. KHORISSI*** *&* ***Mme. BOUGHRIRA*** *qui ont accepté de nous honorer en acceptant d'examiner, de juger et d'évaluer notre mémoire fin d'étude.*

À toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce travail, nous tenons à exprimer notre profonde gratitude et à dire, MERCI.

Dédicaces

Je dédie ce travail

Aux deux personnes les plus nobles, précieux et les plus chères au Monde Ma mère, Mon père que DIEU les gardes. A mon cher père qui n'a jamais cessé de m'encourager et de me Donner les conseils fructueux, qui a fait de son mieux pour assurer La continuité de mes études.

A ma très chère mère, Mère exemplaire pour mes frères et pour Moi-même, tu as su donner l'éducation qu'il nous faut pour Affronter les épreuves de la vie.

Tu nous as comblés de ton amour maternel et tu répondais Présente à chacune de nos sollicitations. Puisse le Tout Puissant T'accorder longue vie afin de profiter des fruits de ce labeur.

A mes secoures et mes frères, pour leurs aides, disponibilités et précieux conseils, Que la vie vous apporte toute la joie et le bonheur.

A mes meilleurs amis

Yasmine, Taoues, Chiraz, Ouissel & Zineb

A mon binôme,

Je tiens à exprimer ma profonde gratitude. Notre collaboration, notre complémentarité et notre travail d'équipe ont été la clé de notre réussite.

Sabrine

Dédicaces

Je dédie ce modeste travail à :

Ma mère et Mon père :

Qui ont œuvré à ma réussite, de par leur amour, leur soutien, tous les sacrifices consentis et leurs précieux conseils, pour toute leur assistance et leur présence dans ma vie.

A mes frères et mes sœurs : ***Sarah, Bilel, Houssem-El-Din et Hadil :***

Vos encouragements et votre présence partout m'ont été d'une grande aide durant mes études. Je tiens à vous exprimer ma gratitude et à vous exprimer mes sentiments sincères et ma gratitude éternelle. Dieu vous protège tous.

A la famille : ***MORSLI*** et ***BOURAHLA.***

A mon binôme.

A mes enseignants qui m'ont formé tout au long de mon cursus scolaire et universitaire.

A toute personne ayant contribué de prêt ou de loin à l'élaboration de ce travail.

A tous ceux qui me sont chers.

Mohamed-Ridha

ملخص :يركز مشروع التخرج الخاص بنا على استخدام الشبكات العصبية التلافيفية) CNN) لتحسين دقة التعرف التلقائي على السماعات. قمنا بتدريب نموذج CNN على مجموعة بيانات صوتية وحصلنا على نتائج مهمة من حيث دقة تحديد المتحدث. يفتح هذا النهج آفاقًا جديدة لتطبيقات الأمان القائمة على الصوت ويوفر فرصًا بحثية لزيادة تحسين أداء أنظمة التعرف على المتحدثين من خلال استخدام التعلم العميق وشبكات CNN .

كلمات المفاتيح: التعرف على السماعات، الشبكات العصبية البناء CNN، تصنيف السماعات، الخصائص الصوتية.

Résumé : Notre projet de fin d'étude se concentre sur l'utilisation des réseaux de neurones convolutifs (CNN) pour améliorer la précision de la reconnaissance automatique du locuteur. Nous avons entraîné un modèle CNN sur un ensemble de données vocales et obtenu des résultats significatifs en termes de précision d'identification des locuteurs. Cette approche ouvre de nouvelles perspectives pour des applications de sécurité basées sur la voix et offre des opportunités de recherche pour améliorer encore les performances des systèmes de reconnaissance du locuteur grâce à l'utilisation du deep Learning et des CNN.

Mots Clés : Reconnaissance du locuteur, Réseaux de neurones constructifs (CNN), Classification des locuteurs, Caractéristiques audio.

Abstract: Our final year project focuses on the use of Convolutional Neural Networks (CNN) to enhance the accuracy of automatic speaker recognition. We trained a CNN model on a dataset of speech samples and achieved significant results in terms of speaker identification accuracy. This approach opens up new possibilities for voice-based security applications and provides research opportunities to further improve speaker recognition systems performance through the utilization of deep learning and CNN.

Keywords: Speaker recognition, Convolutional neural networks (CNN), Speaker classification, Audio features.

Listes des abréviations

RAP : Reconnaissance Automatique De la Parole

RAL : Reconaissance Automatique de Locuteur

TD : Text Dependent

TI : Texte Independent

ID : Identification

VAL : Vérification Automatique du Locuteur

IA : Intelligence Artificielle

RNA : Réseau Neurone Artificiel

CNN : Convolutional Neural Network

RNN : Recurrent Neural Network

GMM : Gaussian Mixture Model

RELU : Rectified Linear Unit

GPU : Graphical Processing Unit

CPU : Central Processing Unit

FFT : Fast Fourier Transform

Table des matières

Liste des Figures
Chapitre1 :

Chapitre 2 :

Liste des Tableaux

Chapitre 2 :

Introduction générale

La reconnaissance du locuteur est une discipline de la reconnaissance de forme qui vise à identifier de manière automatique et précise l'identité d'une personne à partir de sa voix. Elle trouve des applications dans de nombreux domaines tels que la sécurité, la biométrie, la communication vocale et l'assistance virtuelle. Au fil des années, de nombreuses approches ont été développées pour résoudre le problème de la reconnaissance du locuteur. Cependant, l'émergence du deep learning a révolutionné ce domaine en permettant des avancées significatives dans la précision et la performance des systèmes de reconnaissance du locuteur.

Le deep learning, ou apprentissage profond, est une branche de l'intelligence artificielle qui repose sur des réseaux de neurones artificiels profonds. Ces réseaux sont capables d'apprendre de manière autonome à partir de grandes quantités de données et de découvrir des représentations hiérarchiques complexes. Dans le cas de la reconnaissance du locuteur, le deep learning permet d'extraire des caractéristiques discriminantes à partir des signaux vocaux et de construire des modèles capables d'identifier de manière fiable l'identité d'un locuteur.

L'objectif de ce mémoire est d'explorer les possibilités offertes par le deep Learning pour la reconnaissance du locuteur. Nous chercherons à développer un système de reconnaissance du locuteur basé sur des modèles de deep learning, en utilisant des techniques avancées telles que les réseaux de neurones convolutionnels (CNN). Nous évaluerons également les performances de notre système en termes de précision, de robustesse et d'évolutivité.

La présente étude se concentrera sur l'analyse des données vocales, le prétraitement des signaux et l'apprentissage automatique supervisé pour la classification des locuteurs. Nous utiliserons un ensemble de données de référence comprenant des enregistrements vocaux provenant de locuteurs différents, et nous explorerons différentes architectures de réseaux neuronaux pour extraire les caractéristiques pertinentes et effectuer la classification.

La structure de ce mémoire est la suivante : Le premier chapitre représente une vue générale sur la reconnaissance du locuteur, commençons par aborder la production de la parole afin de comprendre comment un individu peut être reconnu par sa voix les différentes taches de la reconnaissance du locuteur et aussi comment fonctionne le système de reconnaissance de locuteur.

Le deuxième chapitre aborde les deux principaux domaines de l'intelligence artificielle qui sont le machine learning et le deep learning, ceci en mettant l'accent sur les réseaux de neurones

artificiels et plus précisément sur les réseaux de neurones convolutives, dans ce chapitre, nous avons examiné leur architecture, leurs paramètres, ainsi que leurs avantages dans le domaine de la reconnaissance de locuteur.

Le troisième chapitre est abordé pour l'implémentation et l'évaluation de notre système de reconnaissance du locuteur basé sur les CNN.

1.1 Introduction

Reconnaitre une personne par sa voix est de plus en plus un enjeu fort en matière d'authentification des personnes à des fins de vérification et de sécurité. Dans ce chapitre, nous allons présenter une vue générale sur la reconnaissance du locuteur. Commençons par aborder la production de la parole afin de comprendre comment un individu peut être reconnu par sa voix. Ensuite, nous passerons à l'état de l'art de la reconnaissance du locuteur, en examinant son architecture, ses différentes applications et ses principales branches, aussi la reconnaissance de la parole et la reconnaissance du langage.

1.2 La parole

La parole est le mode de communication le plus naturel dans toute société humaine. Elle est définie comme un signal réel, continu, d'énergie finie et non stationnaire, généré par l'appareil vocal humain. La parole offre aux êtres humains un moyen facile d'établir une communication claire et compréhensible [1].

Dans le domaine du traitement du signal, la parole occupe une place prépondérante en raison de ses caractéristiques acoustiques distinctives, qui trouvent leur origine dans les mécanismes de production spécifiques.

1.2.1 Production de la parole

Les différences de voix entre les individus sont dues à la structure de leurs organes articulatoires, tels que la longueur du tractus vocal, les caractéristiques des cordes vocales et les variations dans leurs habitudes de parole.

Chez un adulte, le tractus vocal mesure généralement environ 17 cm et fait partie des organes impliqués dans la production de la parole, situés au-dessus des plis vocaux (Anciennement appelés cordes vocales). Comme illustré dans la Figure 1.1, ces organes comprennent le pharynx laryngé (Situé sous l'épiglotte), le pharynx oral (derrière la langue, entre l'épiglotte et le voile du palais), la cavité buccale (en avant du voile du palais et délimitée par les lèvres, la langue et le palais), le pharynx nasal (Au-dessus du voile du palais, à l'extrémité arrière des cavités nasales) et la cavité nasale (Au-dessus du palais, s'étendant du pharynx aux narines).

Le larynx est composé des plis vocaux, de la partie supérieure du cartilage cricoïde, des cartilages aryténoïdes et du cartilage thyroïde. La région située entre les plis vocaux est appelée la glotte [2].

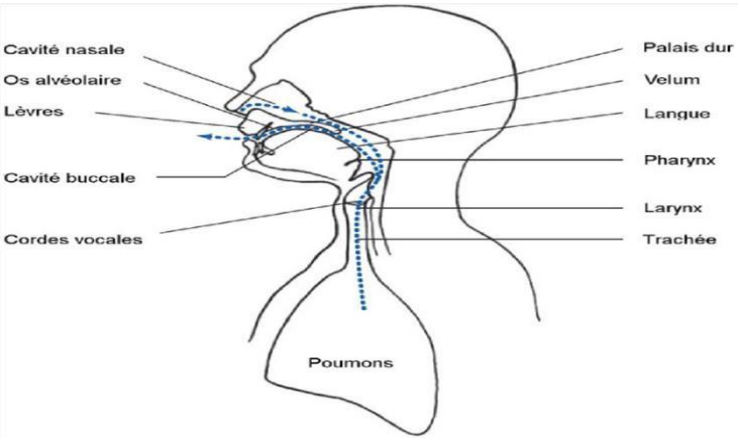

Figure 1.1. Organes de production de la parole humaine [3].

Le résonateur du tractus vocal modifie le spectre acoustique à mesure qu'il traverse le tractus vocal. Les résonances du tractus vocal sont appelées formants. Par conséquent, la forme du tractus vocal peut être estimée à partir de la forme spectrale (Par exemple, l'emplacement des formants et l'inclinaison spectrale) du signal vocal. Les systèmes de reconnaissance des locuteurs utilisent généralement des caractéristiques dérivées uniquement du tractus vocal.

La source d'excitation de la voix humaine contient également des informations spécifiques à chaque locuteur. L'excitation est générée par le flux d'air provenant des poumons, qui passe ensuite par la trachée puis par les plis vocaux. L'excitation est classée en phonation, chuchotement, friction, compression, vibration ou une combinaison de ces éléments. L'excitation de la phonation se produit lorsque le flux d'air est modulé par les plis vocaux. Lorsque les plis vocaux se ferment, la pression s'accumule en dessous jusqu'à ce qu'ils se séparent. Les plis sont ensuite ramenés ensemble par leur tension, leur élasticité et l'effet Bernoulli. L'oscillation des plis vocaux provoque une excitation pulsée du tractus vocal. La fréquence d'oscillation est appelée fréquence fondamentale et elle dépend de la longueur, de la masse et de la tension des plis vocaux. La fréquence fondamentale est donc une autre caractéristique distinctive pour un locuteur donné [2]

1.2.2 Mécanisme de production de la parole

La production de la parole est un processus qui débute par la formulation d'un message linguistique et se transforme en une série d'actions motrices impliquant diverses parties du corps humain. Ce processus aboutit à la création d'un signal vocal intelligible. On peut le diviser en trois étapes distinctes selon les travaux de Brown et Haooort (2000) ainsi que Blank et al. (2002) [4].

a) La conceptualisation (Ou préparation conceptuelle)

Durant cette étape, l'intention de communiquer se traduit par la génération des concepts et idées correspondant au message à transmettre.

b) La formulation

A ce stade, la forme linguistique nécessaire pour exprimer le message désiré est élaborée. Cela implique le choix des mots appropriés, la construction grammaticale adéquate, ainsi que le découpage des mots en syllabes et leur encodage phonétique.

c) L'articulation et l'exécution motrice de la parole

Cette étape consiste en l'exécution de la séquence d'actions articulatoires qui donne vie au message. Le locuteur produit une série de signaux neuromusculaires qui permettent de contrôler les cordes vocales, les lèvres, la mâchoire, la langue et le voile du palais, contribuant ainsi à la production des sons et à la création de la séquence sonore souhaitée.

La production de la parole est un processus complexe et coordonné qui implique la coordination précise de différentes parties du corps et de mécanismes de contrôle pour produire les sons et les mots nécessaires à la communication humaine.

1.2.3 Paramètres du signal de parole

Le signal de parole est un signal continu, non stationnaire et d'énergie finie. Il présente une grande variabilité de sons en fonction du locuteur et des conditions environnementales. Sa structure est complexe et variable dans le temps. Il peut être représenté directement sous forme de signal analogique.

L'analyse d'un tel signal est une tâche difficile en raison du grand nombre de paramètres associés. Néanmoins, trois paramètres principaux se dégagent : la fréquence fondamentale, le spectre fréquentiel et l'énergie. Ces paramètres sont appelés caractéristiques acoustiques et sont énumérés ci-dessous [3].

a) Fréquence fondamentale

Cela correspond à la fréquence à laquelle les cordes vocales s'ouvrent et se ferment, on l'appelle parfois la hauteur de la voix. Elle varie en fonction de la taille du larynx. Un enfant a un larynx plus petit qu'une femme, qui a elle-même un larynx plus petit qu'un homme. C'est pourquoi un enfant a une voix plus aiguë. Cette fréquence caractérise uniquement les sons sonores et peut varier [5] :

• de 80 Hz à 200 Hz pour une voix d'homme ;

• de 150 Hz à 450 Hz pour une voix de femme ;

• de 200 Hz à 600 Hz pour une voix d'enfant.

b) Spectre

Le spectre fréquentiel est une représentation d'un signal qui indique la répartition des fréquences présentes dans ce signal. Il est constitué d'un ensemble de fréquences disposées de manière arithmétique. Une caractéristique essentielle du spectre fréquentiel est le timbre, qui permet d'identifier chaque locuteur par sa voix [3].

c) L'énergie

L'énergie sonore correspond à l'intensité du signal. Elle est généralement plus élevée pour les segments de parole voisés que pour les segments non voisés. La Figure 2.5 présente un exemple concret du signal de parole pour le mot "Tashghil", qui signifie "Allumer" [3].

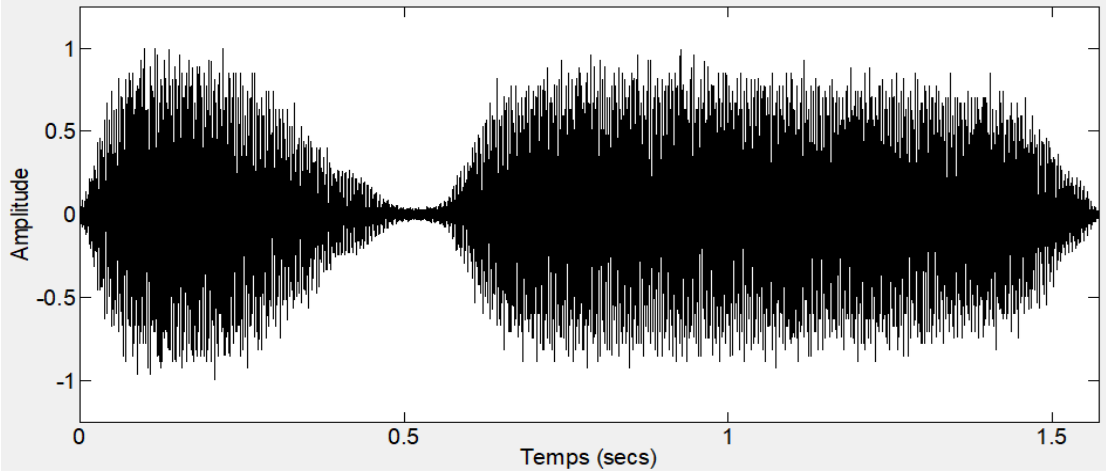

Figure 1.2. Signal de parole.

1.3 La reconnaissance

En termes plus généraux, la reconnaissance fait référence à la capacité de reconnaître, de comprendre ou de distinguer quelque chose. Cependant, la reconnaissance peut être utilisée dans différents domaines et a des significations spécifiques.

1.3.1 La reconnaissance de parole

La reconnaissance automatique de la parole est une technologie informatique qui permet à un logiciel d'interpréter la parole humaine naturelle. Cela Permet à la machine d'extraire le message verbal contenu dans le signal vocal et de l'analyser ce signal en une chaîne de mots ou phonèmes représentant ce que la personne a prononcé. Cette technologie utilise des méthodes informatiques dans le domaine du traitement du signal et de l'intelligence artificielle [6].

Les systèmes de reconnaissance de parole se sont considérablement améliorés ces dernières années grâce aux progrès de l'apprentissage automatique, en particulier du deep learning, qui permettent une meilleure compréhension et interprétation de la parole humaine, même dans des environnements bruyants ou avec des accents différents. Cependant, ils ne sont pas encore parfaitement précis et peuvent avoir des difficultés avec certains mots ou accents particuliers, ce qui nécessite encore une relecture ou une correction manuelle dans certains cas**.**

Le schéma suivant résume les étapes de base d'un système RAP :

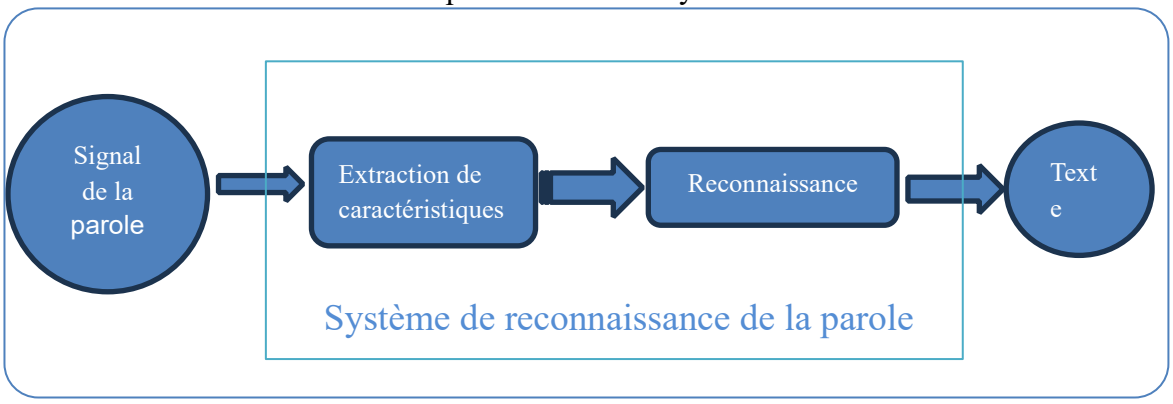

Figure 1.3. Structure de base d'un système RAP.

Les applications de la RAP sont nombreuses et peuvent varier selon leur type, les évolutions des techniques de RAL ont permis aux systèmes d'évoluer et d'être de plus en plus efficaces. De plus, la majorité des systèmes RAL sont des systèmes dépendants ou indépendants de texte. Les systèmes dépendant de texte nécessitent une phase d'apprentissage où de nombreuses heures de parole sont généralement indispensables [8].

Cette partie décrit brièvement les quatre principaux types de systèmes de reconnaissance vocale.

a) Commandes vocales

Les systèmes de reconnaissance de parole sont utilisés dans les assistants vocaux, tels que Siri, Alexa et Google Assistant, pour permettre aux utilisateurs de contrôler des appareils électroniques et de rechercher des informations à l'aide de commandes vocales.

b) Applications de sécurité

La reconnaissance de parole peut être utilisée dans les systèmes de sécurité pour identifier les voix autorisées et permettre l'accès à des zones sécurisées

c) Systèmes de dictée automatique

La dictée de texte est l'une des applications les plus courantes de la reconnaissance de parole. Elle permet aux utilisateurs de dicter du texte et de le transcrire automatiquement en texte écrit.

d) Systèmes de compréhension

Essentiellement, ils permettent de dialoguer avec une machine. Par conséquent, l'utilisateur prononce une série de mots-clés que le système est peu de reconnaître. Contrairement aux systèmes à commande vocale, ce genre de système utilise aussi un dispositif de compréhension de mots pour interpréter et réagir en conséquence [9].

1.3.2 La reconnaissance automatique de locuteur

La reconnaissance automatique du locuteur est interprétée comme une tâche particulière de reconnaissance de formes. Elle consiste à identifier la personne qui parle en se basant sur sa voix. La variabilité de la parole entre locuteurs est essentielle pour la RAL, car cela permet de distinguer une voix parmi plusieurs. Contrairement à la reconnaissance automatique de la parole, la RAL s'intéresse aux informations non linguistiques contenues dans le signal vocal. Cependant, la RAL bénéficie souvent des avancées de la reconnaissance automatique de la parole, avec de nombreuses techniques appliquées en RAP avant d'être adaptées à la RAL.

Elle peut être réalisée à l'aide de différents algorithmes, tels que les réseaux de neurones, les machines à vecteurs de support et les modèles de Markov cachés. Les performances de ces algorithmes dépendent de nombreux facteurs, tels que la qualité de l'enregistrement audio, le bruit de fond et la variabilité interpersonnelle. Malgré ces défis, la reconnaissance automatique du locuteur continue de progresser et de trouver de nouvelles applications dans le monde de la technologie.

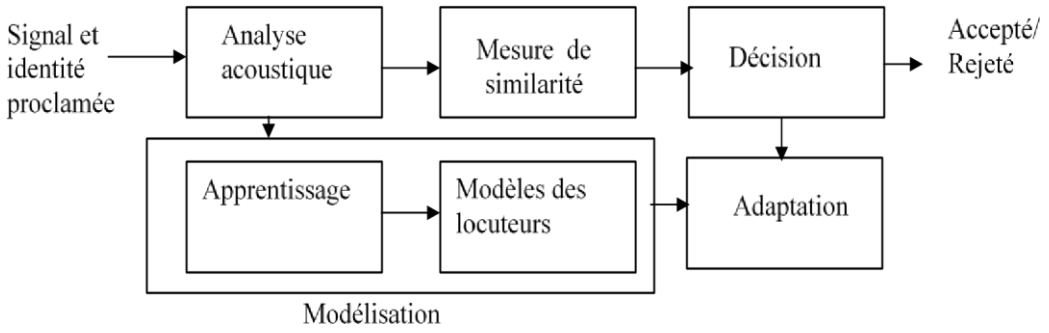

Figure 1.4. Schéma d'un système RAL [11].

Le système de reconnaissance de locuteur peut fonctionner soit en mode dépendant du texte (TD) soit en mode indépendant du texte (TI). En mode TD, l'utilisateur parle une transcription de texte prédéfinie ou sollicitée. En mode TI, l'utilisateur est autorisé à parler librement. Étant donné que le mode TD fournit des informations supplémentaires au système de reconnaissance de locuteur, il fonctionne généralement mieux que le mode TI. Différentes études ont été menées afin de réduire l'écart de performance entre les deux modes de fonctionnement [12].

a) Dépendante du texte

En mode TD, un système de reconnaissance de locuteur peut être trompé par l'enregistrement et la lecture de la voix prédéfinie d'un locuteur inscrit. Pour défendre un système de reconnaissance de locuteur contre de telles attaques malveillantes, le système peut demander à l'utilisateur de prononcer un texte aléatoire. Dans la plupart des cas, un système de reconnaissance de locuteur en mode TD fonctionne mieux qu'un système en mode TI car des informations supplémentaires (transcription de texte) sont fournies. Cependant, un système de reconnaissance de locuteur en mode TD ne peut pas être utilisé lorsque la transcription sousjacente réelle n'est pas fournie, comme dans la situation où quelqu'un parle librement au téléphone [12].

b) Indépendante du texte

Un système de reconnaissance de locuteur en mode TI ne nécessite pas de transcription sousjacente réelle d'une parole en entrée. Cela peut être utile pour un système de reconnaissance de locuteur en forensique, tel que l'identification d'un locuteur dans une conversation interceptée par fil ou dans une interface homme-robot [12].

1.3.3 Domaine d'application de la RAL

La reconnaissance automatique du locuteur est utilisée dans de nombreuses applications, Voici certains domaines d'application de cette technologie [11] :

a) Applications sur site

La personne doit faire l'objet d'une présentation physique à un endroit précis.

• Verrouillages vocaux (pour locaux, compte informatique, etc.)

• Interactivité matérielle (retrait d'argent à un guichet automatique, …)

b) Applications liées aux télécommunications

La vérification s'opère à distance :

• Accès à des services pour des abonnés, ou des données confidentielles.

• Transaction à distance.

c) Applications commerciales

• Associer un même mot de passe pour une petite population de locuteur (membre d'une famille, d'une société).

• Protection de matériel contre le vol.

d) Applications judiciaires

• Recherche de suspects et de preuves.

• Les juges, avocats, enquêteurs de police ou de gendarmerie souhaitent utiliser des procédés de reconnaissance vocale pour enquêter ou confirmer le coupable ou l'innocent.

1.3.4 Evaluation d'un système de RAL

La reconnaissance automatique de la parole est un domaine en constante évolution, avec des chercheurs se concentrant sur les limites pratiques de la reconnaissance automatique du locuteur ainsi que sur la manière de présenter, diffuser et interpréter les résultats des systèmes. Pour assurer des performances acceptables dans les applications de reconnaissance du locuteur, plusieurs caractéristiques sont généralement nécessaires [11] :

-Un environnement calme et l'utilisation d'un microphone de bonne qualité ;

-La connaissance et le contrôle des conditions d'enregistrement et de traitement du signal audio ;

-La disponibilité de données de parole enregistrées dans les mêmes conditions que le signal de test, permettant de référencer les locuteurs dans le système ;

-L'interdiction pour les locuteurs d'utiliser des techniques sophistiquées pour modifier ou déguiser leur voix ;

-L'étalonnage de la mesure de ressemblance en effectuant des expériences dans des conditions contrôlées, telles que mentionnées précédemment ;

-La coopération des locuteurs qui souhaitent être acceptés par le système et qui collaborent avec celui-ci ;

-L'interdiction d'utiliser un système de synthèse de la parole.

1.3.5 Structure de base de RAL

Les systèmes de reconnaissance des locuteurs se composent généralement de trois unités principales comme le montre la figure1.4, L'entrée du premier étage ou du système de traitement frontal est le signal de parole. Ici, le discours est numérisé et ensuite l'extraction des caractéristiques a lieu. Il n'y a pas des caractéristiques exclusives qui transmettent l'identité des locuteurs dans le signal vocal, mais il est connu de la théorie du filtre source de la production de la parole que la forme du spectre de la parole code les informations sur la forme du conduit vocal des locuteurs via des formants et la source glottale via des harmoniques de hauteur [2].

Par conséquent, une forme ou l'autre des caractéristiques spectrales est utilisée dans la plupart des systèmes de reconnaissance des locuteurs. Le processus final de l'étape de traitement frontal est une forme de compensation de canal. Différents dispositifs d'entrée (par exemple différents combinés téléphoniques) imposent différentes caractéristiques spectrales au signal de parole, telles que la limitation de bande et la mise en forme. Par conséquent, une compensation de canal est effectuée pour éliminer ces effets indésirables. Le plus souvent, une certaine forme de compensation de canal linéaire, telle que la soustraction moyenne cepstrale à long et à court terme, est appliquée aux entités.

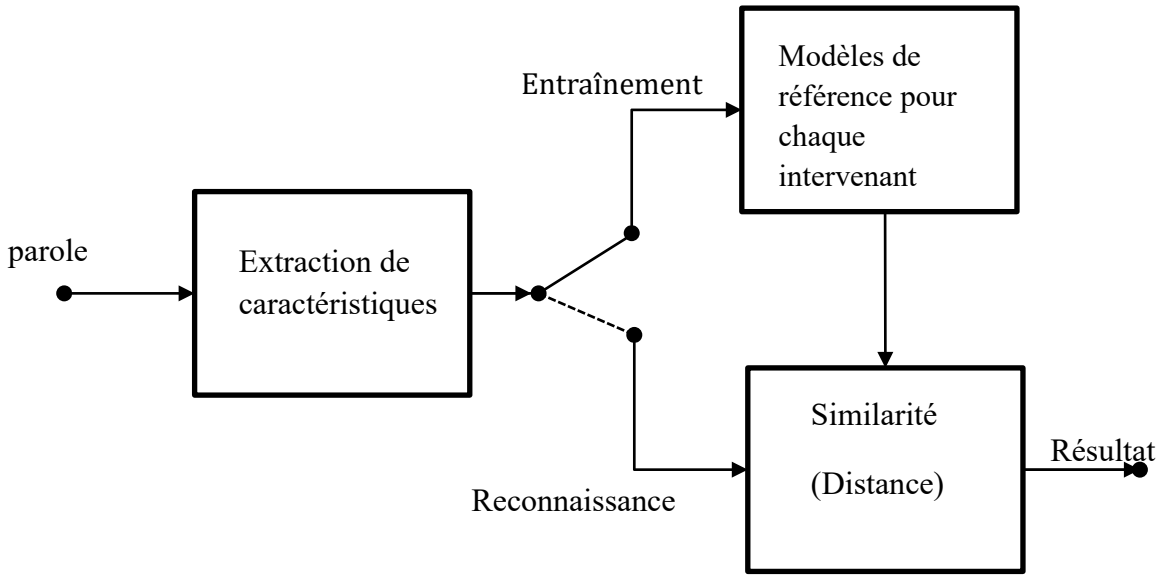

Figure 1.5. Schéma fonctionnel d'un système de RAL [2].

Le processus de reconnaissance du locuteur comprend la phase de formation et la phase de reconnaissance. Dans la phase d'apprentissage, les caractéristiques d'un signal de parole de haut-parleur sont stockées en tant que caractéristiques de référence. Les vecteurs caractéristiques de la parole sont utilisés pour créer un modèle de locuteurs. Le nombre de modèles de référence requis pour une reconnaissance efficace du locuteur dépend du type de fonctionnalités ou de techniques utilisées par le système pour reconnaître le locuteur. Dans la phase de reconnaissance, des caractéristiques similaires à celles utilisées dans le modèle de référence sont extraites d'un énoncé d'entrée du locuteur dont l'identité doit être déterminée. La décision de reconnaissance dépend de la distance calculée entre le modèle de référence et le modèle conçu à partir de l'énoncé d'entrée. Dans l'identification du locuteur, la distance entre un énoncé d'entrée et tous les modèles de référence disponibles est calculée. Le modèle de l'utilisateur enregistré, dont la distance avec le modèle d'énoncé d'entrée est la plus petite, est finalement sélectionné comme locuteur de l'énoncé d'entrée.

En cas de vérification du locuteur, la distance est calculée uniquement entre l'énoncé d'entrée et le modèle de référence du locuteur revendiqué. Si la distance est inférieure au seuil prédéterminé, le locuteur est accepté autre le locuteur est rejeté en tant qu'imposteur [2].

1.3.6 Les différentes branches de la RAL

L'identification automatique du locuteur et la vérification automatique du locuteur sont les deux tâches les plus répandues dans le domaine de RAL [13].

a) Identification du locuteur

L'identification du locuteur sert à trouver l'identité d'un locuteur spécifique à l'aide de sa voix parmi un ensemble limité de locuteurs possibles. Cette tâche peut également être de deux types in-set et off-set. Dans le cas de in-set, le système automatisé suppose que le locuteur test doit être parmi les locuteurs connus, tandis que dans le cas off-set, le système peut déterminer que le locuteur test ne fait pas partie des locuteurs identifiés [5].

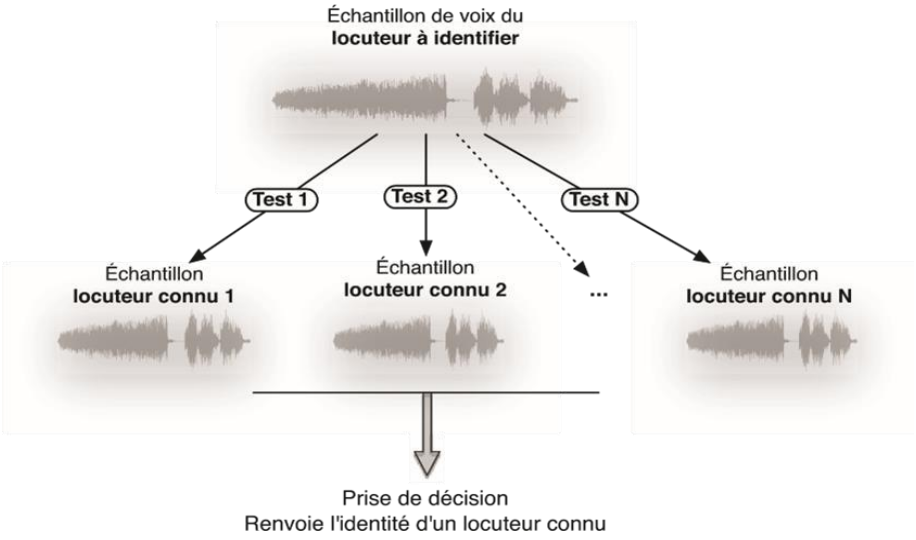

Figure1.6. Principe de l'identification automatique du locuteur.

• Le fonctionnement de système d'identification du locuteur :

L'identification automatique du locuteur est divisée en deux étapes : une étape d'apprentissage et une étape de test. A partir d'un ensemble d'enregistrements pour chaque locuteur, le système apprend un modèle pour chaque locuteur dans une étape d'apprentissage. Différentes techniques d'apprentissage automatique, telles que les réseaux de neurones peuvent être utilisées pour construire le modèle de locuteur. Une fois qu'un modèle de locuteur est développé, il peut être utilisé pour prédire l'identité de locuteurs inconnus. Pendant la phase de test, le système compare les échantillons de parole reçus avec différents modèles qu'il a appris pour déterminer si l'identité du locuteur est connue [5].

• Les applications de l'identification du locuteur :

Il existe plusieurs applications de l'identification de locuteur, notamment la sécurité, les services bancaires, les centres d'appels, l'assistance vocale et la surveillance. Par exemple, les applications de sécurité utilisent souvent l'identification de locuteur pour authentifier les

utilisateurs et permettre l'accès à des zones sécurisées, tandis que les banques peuvent l'utiliser pour vérifier l'identité des clients lorsqu'ils appellent le service client. Les centres d'appels peuvent également utiliser l'identification de locuteur pour acheminer les appels vers le bon agent en fonction de l'identité du client. Enfin, les applications de surveillance peuvent utiliser l'identification de locuteur pour identifier les personnes qui parlent dans une pièce ou une conversation téléphonique [4].

b) Vérification de locuteur

La vérification du locuteur est une tâche de la reconnaissance du locuteur qui vise à déterminer si une personne est bien celle qu'elle prétend être en fonction de son échantillon vocal. Contrairement à l'identification du locuteur. La vérification du locuteur se concentre sur la comparaison entre l'échantillon vocal de la personne et les caractéristiques vocales préalablement enregistrées pour cette personne dans une base de données.\

Le système de vérification du locuteur enregistre un échantillon vocal de la personne, extrait les caractéristiques vocales pertinentes, crée un modèle vocal spécifique à cette personne, puis compare les caractéristiques vocales d'un nouvel échantillon vocal pour prendre une décision de vérification.

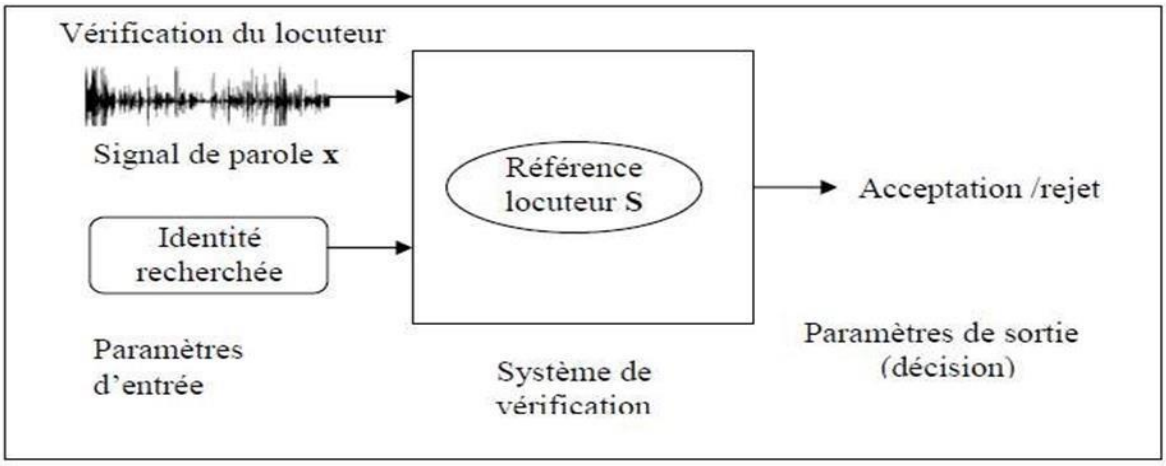

Figure 1.7. Principe de base de la vérification du locuteur.

- **Le fonctionnement de système de vérification du locuteur :**

Le système, comme le montre la (Figure 1.11), comprend principalement deux parties, à savoir la partie formation et la partie test du locuteur. Chaque partie consiste en un prétraitement numérique de la parole, une extraction de caractéristiques et une réduction de caractéristiques (Facultatif, dépend de la performance).

Dans la partie d'entraînement, nous partons du module d'inscription des utilisateurs. Lorsqu'un nouvel utilisateur est ajouté au système, le module est utilisé pour "apprendre" au système la voix du nouvel utilisateur. L'entrée de ce module est constituée des mots d'identification que le nouvel utilisateur prononce dans un microphone. Ces mots prononcés sont utilisés pour les « données d'entraînement ». Après avoir échantillonné le signal vocal analogique et l'avoir converti en signal numérique, nous effectuons le prétraitement numérique de la parole, l'extraction et la réduction des caractéristiques. Les caractéristiques ainsi obtenues sont ensuite compressées par un bloc de compression de caractéristiques (regroupement) pour former un livre de codes. Le nouveau livre de codes est stocké dans la base de données pour une utilisation future. Le nouveau livre de codes se voit attribuer un index pour indiquer le nouvel utilisateur avec les données de formation et le nouveau livre de codes, nous pouvons évaluer les performances du système dans le module de génération de seuil. Ce module permet de définir un niveau de sensibilité du système vis-à-vis de chaque utilisateur. Cette valeur de sensibilité est appelée seuil et doit être générée chaque fois qu'un nouvel utilisateur est inscrit.

La valeur de seuil peut être réinitialisée, par exemple, lorsqu'un utilisateur a reçu de nombreux faux rejets et doit ajuster le niveau de sensibilité. Dans la partie test du locuteur, le module de vérification du locuteur est utilisé pour identifier un utilisateur. Tout d'abord, un utilisateur informe le système qu'il ou elle est un utilisateur. Le système indiquera alors à l'utilisateur de prononcer les mots de vérification. Cette prononciation des mots est appelée le discours de test. Le module procédera aux mêmes prétraitements numériques de la parole, extraction et réduction de caractéristiques (facultatifs, utilisés si la partie apprentissage l'utilise) que ceux utilisés dans la partie apprentissage. Les caractéristiques extraites de la parole de test sont ensuite comparées aux livres de codes de la base de données. Sur la base de certaines métriques de similarité, le système décidera si l'utilisateur a réussi ou échoué le test de vérification vocale [14].

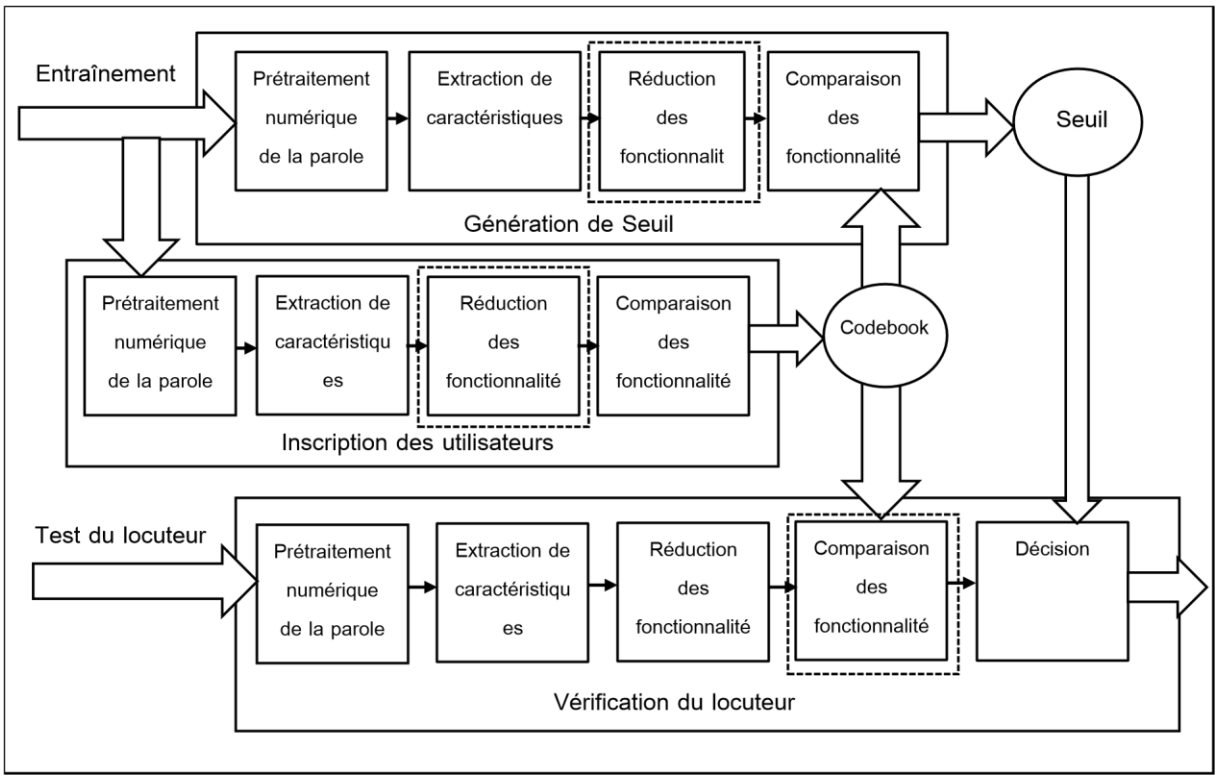

Figure 1.8. Configuration de système de vérification des locuteurs [14**].**

- **Les applications de la vérification du locuteur**

Les applications du VAL sont diverses et principalement commerciales :

- Verrouillage vocal pour le contrôle d'accès aux installations.
- Authentification pour l'accès à distance.
- Protection contre le vol d'appareils (Téléphone portable, voiture etc…).

1.3.7 L'évolution de la reconnaissance du locuteur

La reconnaissance du locuteur a considérablement évolué au fil du temps en raison des progrès de la technologie et des progrès de la recherche. Initialement, ces systèmes étaient principalement basés sur les propriétés acoustiques, mais ils étaient sensibles aux changements des conditions environnementales et de la parole. Par la suite, l'intégration des caractéristiques linguistiques permet de mieux distinguer les locuteurs.

L'avènement de l'apprentissage automatique a révolutionné le domaine en utilisant des méthodes telles que les modèles de mélange gaussien (GMM), les réseaux de neurones et l'apprentissage en profondeur. Cela rend la reconnaissance du locuteur plus précise. Des caractéristiques comportementales telles que le débit de parole sont également intégrées pour

améliorer l'identification du locuteur. L'adoption de l'apprentissage par transfert surmonte les limites associées au manque de données sur la formation. Enfin, les développements récents ont vu l'intégration des caractéristiques visuelles dans la reconnaissance du locuteur grâce à l'analyse des mouvements des lèvres et des traits du visage. Le développement continu de la reconnaissance du locuteur promet d'améliorer les performances et de s'adapter à divers scénarios d'utilisation [15].

1.4 Conclusion

La technologie de reconnaissance automatique du locuteur (RAL) a progressé de manière significative, atteignant un niveau de maturité qui lui permet d'être considérée comme une solution fiable pour des applications réelles. L'authentification basée sur la voix est de plus en plus répandue dans les systèmes de sécurité modernes.

Dans ce chapitre, nous avons examiné dans la première partie la production de la parole et ses paramètres, puis nous avons passé à étudier l’état de l’art de système de reconnaissance automatique du locuteur, présentons son structure de base ainsi que ses deux tâches principales, avec ses applications, ses fonctionnements et enfin l’évolution de la RAL.

2.1 Introduction

La croissance du Deep Learning améliore notre capacité à comprendre et à analyser des données complexes, ouvrant ainsi de nouvelles perspectives dans divers domaines tels que la médecine, la fabrication, le commerce, le marketing et d'autres domaines. Avec sa capacité à extraire des connaissances et à atteindre une grande précision dans les tâches de calcul, le Deep Learning est devenu un outil essentiel pour améliorer les systèmes intelligents et stimuler le progrès technologique. Au cours de ce chapitre, nous allons examiner plusieurs aspects clés liés à l'utilisation des réseaux de neurones convolutifs (CNN) dans le domaine de la reconnaissance du locuteur.

2.2 l'intelligence artificielle

L'intelligence artificielle (I.A) est un domaine scientifique qui se concentre sur la création des programmes qualifiés d'intelligents. Par programmes intelligents, on entend généralement des programmes capables de résoudre des problèmes qui ont traditionnellement été considérés comme relevant des capacités humaines [16].

Il convient de mentionner que l'intelligence artificielle englobe le domaine de l'apprentissage automatique, également connu sous le nom de "machine learning" qui à son tour, inclut l'apprentissage profond, également appelé "deep learning". Ces trois concepts sont intimement liés et interdépendants [17].

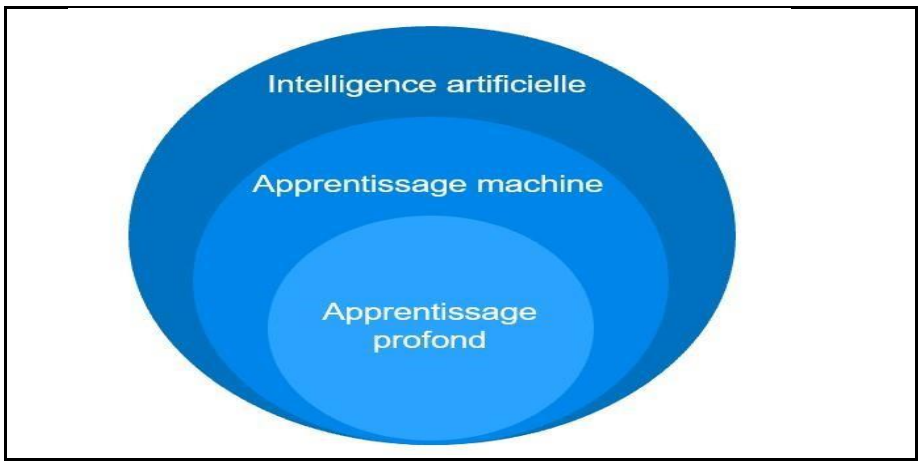

Figure 2.1. Les différents domaines de l'intelligence artificielle [17]

2.3 Apprentissage automatique

L'apprentissage automatique, également connu sous le nom de machine learning, est un domaine de l'intelligence artificielle qui cherche à doter les machines de la capacité d'apprendre à partir de données en utilisant des modèles mathématiques. En substance, il s'agit du procédé par lequel des informations significatives sont extraites à partir d'un ensemble de données d'entraînement.

L'objectif de cette phase est d'obtenir les paramètres d'un modèle qui atteindront les meilleures performances, notamment lors de l'exécution de la tâche assignée au modèle. Une fois l'apprentissage réalisé, le modèle peut ensuite être déployé en production.

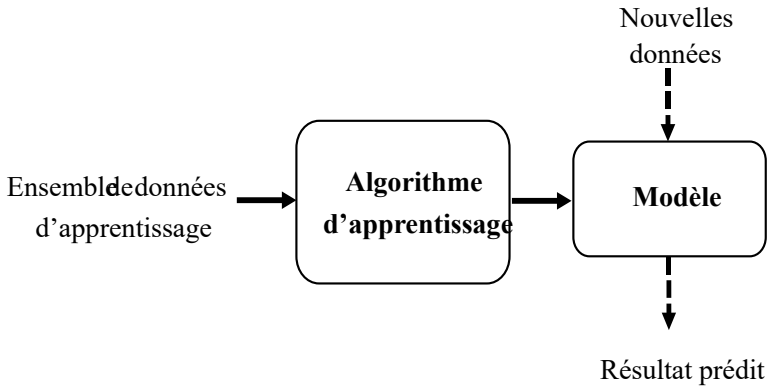

Figure 2.2. Processus de l'apprentissage machine [3].

L'apprentissage automatique se subdivise en différents types, chacun étant défini par la nature des tâches à accomplir. Dans les prochaines sections, nous examinerons les principaux types d'apprentissage. [16].

2.4 Apprentissage profond

L'apprentissage profond est une branche de l'apprentissage automatique utilisée pour entraîner des systèmes informatiques appelés réseaux neuronaux artificiels (RNA). Ces techniques reposent sur des algorithmes capables de reproduire les comportements du cerveau humain. Les RNA peuvent résoudre des problèmes complexes tels que la reconnaissance d'images, la reconnaissance vocale et le traitement du langage naturel. Les algorithmes d'apprentissage profond mettent en œuvre plusieurs couches de neurones ou de nœuds artificiels

avec diverses connexions entre eux. Les couches de nœuds sont liées par différents types de connexions.

Ces connexions sont entrainées à reconnaître et comprendre les caractéristiques d'un ensemble de données donné. Cette structure permet aux algorithmes d'apprendre de leurs expériences et d'améliorer leur manière d'accomplir leurs tâches [3].

2.5 Types d'apprentissage

2.5.1 Apprentissage supervisé

L'apprentissage supervisé est un processus d'apprentissage à partir d'un ensemble d'exemples d'apprentissage étiquetés fournis par un superviseur externe compétent. Cela signifie qu'une expertise humaine est nécessaire pour étiqueter les données. Chaque exemple consiste en une description d'une situation accompagnée d'une étiquette (une classe pouvant être représentée par des valeurs numériques ou nominales) indiquant l'action correcte que le système doit prendre dans cette situation. L'objectif de ce type d'apprentissage est que le système puisse généraliser ses réponses et agir correctement dans des situations non présentes dans l'ensemble d'apprentissage. Ainsi, l'utilisateur fournit à l'algorithme des paires d'entrées/sorties souhaitées (X, y), comme illustré dans la Figure 2.3, et l'algorithme trouve un moyen de produire la sortie souhaitée à partir des entrées. Plus précisément, l'algorithme est capable de générer une sortie pour une entrée qu'il n'a jamais rencontrée auparavant [3].

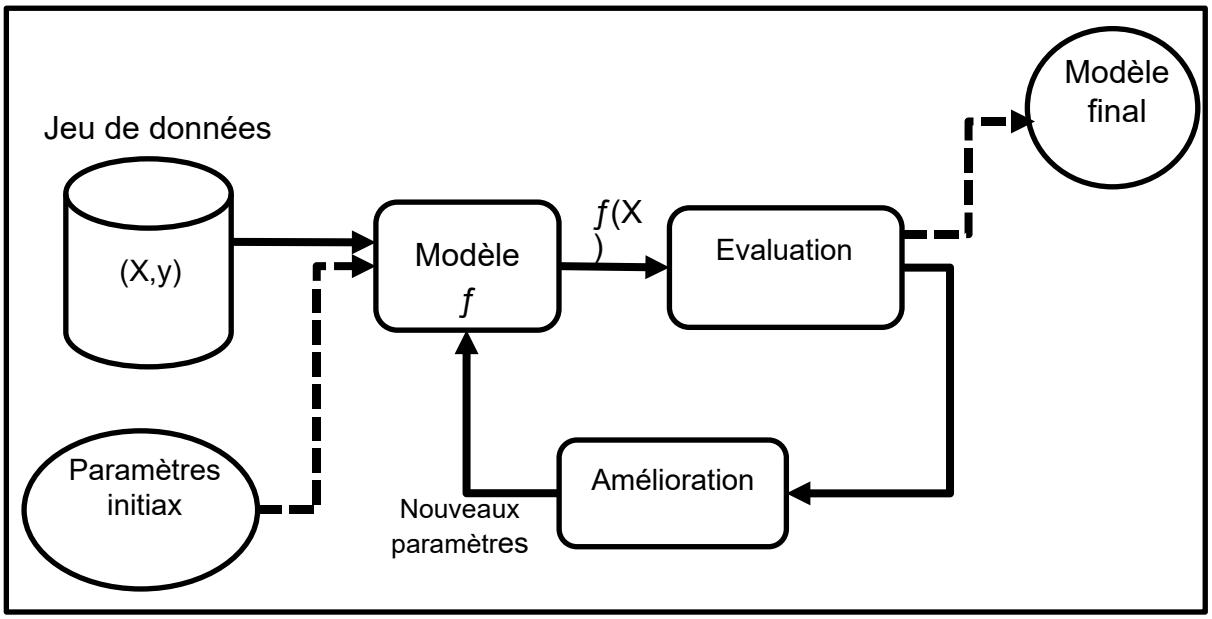

Figure 2.3. Processus de l'apprentissage supervisé [3].

On distingue deux grands types de problèmes d'apprentissage supervisé : la régression et la classification.

a) La classification

Généralement, pour les problèmes de classification l'ensemble de données utilisé avec un nombre fini de classes, chaque exemple est associé à l'une d'entre elles. La cible pour chaque exemple a une valeur discrète représentant une classe particulière. Avec ces données, le modèle d'apprentissage automatique apprendra à attribuer des catégories aux entrées. Par exemple, dans le contexte de la classification de documents, imaginez un ensemble de données composé de trois catégories. Chaque classe correspond à une matière :"Economie", "Politique" et "Autre". Le modèle doit apprendre à déterminer si un document concerne l'économie, la politique ou d'autres sujets, puis doit associer le document à une valeur qui représente la catégorie correcte [16].

b) Régression

Dans le contexte d'un problème de régression, la variable cible est constituée d'un ou de plusieurs éléments ayant des valeurs continues. Un modèle d'apprentissage automatique est entraîné à prédire une ou plusieurs valeurs réelles. La météorologie offre un bon exemple de problème de régression, par exemple la prédiction de la température. En effet, la valeur à prédire dans ce cas est une quantité continue. On peut également inclure d'autres éléments dans la variable cible tels que la pression atmosphérique et le taux d'humidité, ce qui crée un vecteur de valeurs continues [16].

2.5.2 Apprentissage non-supervisé

Lorsque le système ou l'opérateur dispose uniquement d'exemples sans étiquettes, et que le nombre et la nature des classes n'ont pas été prédéterminés, on parle d'apprentissage non supervisé (ou regroupement). Aucune expertise n'est disponible ni requise. L'algorithme doit découvrir par lui-même la structure, plus ou moins cachée des données. Dans cet apprentissage, le système doit cibler les données dans l'espace de description en fonction de leurs attributs disponibles, afin de les regrouper en ensembles homogènes d'exemples. La similarité est généralement calculée à l'aide d'une fonction de distance entre les paires d'exemples. Ensuite, il revient à l'opérateur d'associer ou de déduire une signification pour chaque groupe [18].

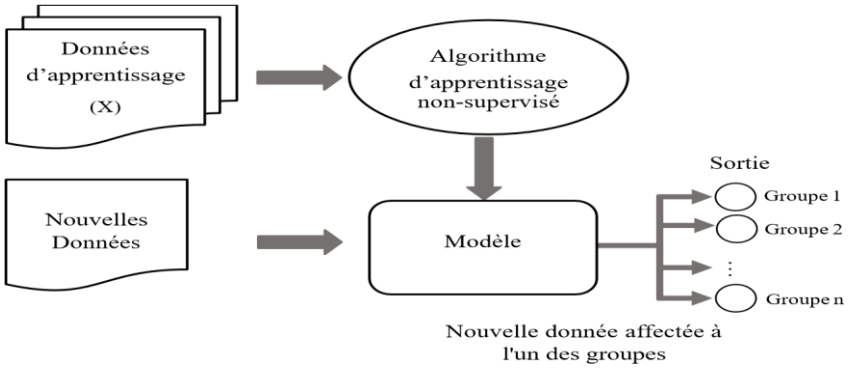

Figure2.4. Processus de l'apprentissage non-supervisé [3].

Il distingue plusieurs problèmes parmi ceux-ci : le regroupement et la réduction de dimensionnalité.

a) Regroupement

La mise en grappes est l'une des techniques algorithmiques les plus importantes et les plus populaires pour l'apprentissage non supervisé. Cet algorithme trouve le modèle et catégorise la collecte des données. Dans cette méthode, nous pouvons traiter les données et identifier les groupes à partir de ces données. Dans ce type d'apprentissage non supervisé, nous pouvons également définir le nombre de groupes que nous souhaitons trouver. Le regroupement se divise ensuite en différents groupes : Exclusif, Agglomérat, Chevauchement et Probabiliste.

b) Réduction de dimensionnalité

Ces méthodes traitent des problèmes de classification et d'apprentissage automatique en fonction de nombreux facteurs. Ces facteurs, appelés caractéristiques, sont des variables des données. Plus vous alimentez l'algorithme de fonctionnalités, plus il est difficile de comprendre l'ensemble d'apprentissage. Ces fonctionnalités sont parfois redondantes et liées. À ce stade, l'aide de l'algorithme de réduction de la dimensionnalité est nécessaire. Cet algorithme non supervisé réduira les variables aléatoires et obtiendra une justification de ces variables.
L'algorithme le divise en différentes extractions de caractéristiques et de sélections.

En conclusion pour les tâches d'apprentissage supervisé et non supervisé, il est nécessaire d'avoir une représentation des données d'entrée compréhensible par un ordinateur. Ces données sont souvent considérées comme un tableau, où chaque ligne représente un exemple de données (instance) et chaque colonne représente une propriété ou une caractéristique qui décrit cet exemple de données.

Il est important de distinguer les sorties discrètes des sorties continues dans les algorithmes d'apprentissage machine. Les sorties discrètes ont tendance à provenir d'un ensemble distinct et fini de valeurs, tandis que les sorties continues sont généralement des valeurs appartenant à un ensemble continu, avec un nombre potentiellement infini de valeurs [3].

La Table suivante résume les cas d'utilisation de ce type de données dans les contextes d'apprentissage supervisé et non supervisé.

Tableau 2.1. Type de données vs type d'apprentissage [3].

Type de donnée	Apprentissage supervisé	Apprentissage non-supervisé
Discrète	Classification	Regroupement
Continue	Régression	Réduction de dimensionnalité

2.5.3 Apprentissage par renforcement

Le domaine de l'apprentissage par renforcement vise à enseigner à un agent comment se comporter de manière appropriée dans un environnement spécifique, c'est-à-dire atteindre un objectif préalablement choisi par l'utilisateur. Le problème à résoudre est divisé en une séquence d'étapes. À chaque étape, l'agent doit choisir parmi un ensemble d'actions, ce qui lui donne la possibilité d'interagir avec son environnement. Contrairement à l'apprentissage supervisé, il n'y a pas de cible permettant d'apprendre un comportement. À la place, l'agent reçoit un signal (déterminé par l'utilisateur) qui lui indique s'il a agi correctement. À chaque étape de la séquence, l'agent reçoit des informations sur son environnement qui l'aideront à choisir l'action appropriée. Pendant l'apprentissage, l'agent cherchera à maximiser le nombre de signaux positifs afin d'améliorer son comportement [16].

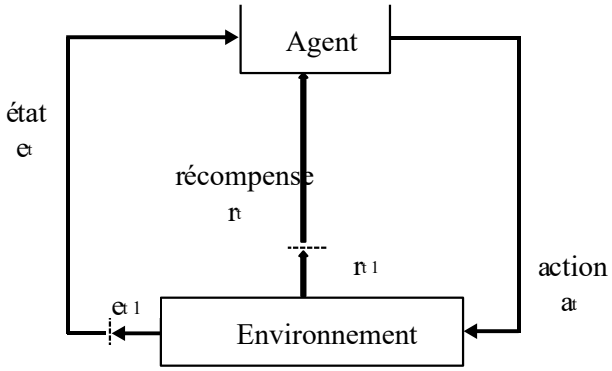

Figure 2.5. Interaction agent-environnement [3].

Ce type d'apprentissage est particulièrement adapté à de nombreuses applications robotiques. Il se distingue de l'apprentissage supervisé et non-supervisé par l'utilisation d'un signal de récompense qui indique simplement si l'action prise par l'agent est bonne ou mauvaise, sans fournir de détails sur la meilleure action à prendre. De plus, il n'utilise ni les données d'apprentissage ni les étiquettes [3].

2.6 Neurone biologique

Le cerveau humain est composé d'environ 1011 neurones, soit mille milliards, avec un nombre de connexions (synapses) allant de 1000 à 10000 par neurone. Le neurone est une cellule qui possède un corps cellulaire, qui agit comme un centre de contrôle et effectue la sommation des informations qui lui parviennent (voir Figure 2.6). Les dendrites, qui se ramifient à partir du corps cellulaire, permettent le transport des informations de l'extérieur vers le corps du neurone. Le neurone traite ensuite ces informations et les transmet le long de l'axone à d'autres neurones. La connexion entre deux neurones est appelée synapse [3].

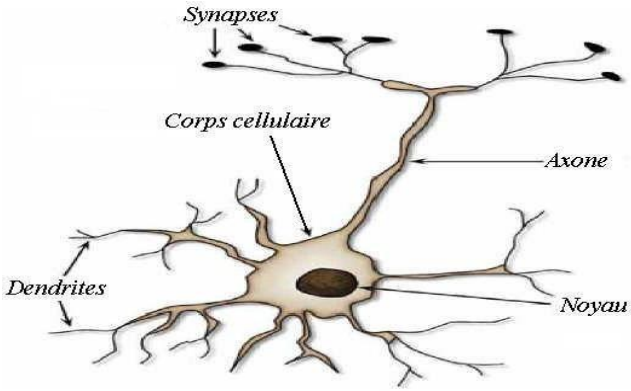

Figure 2.6. Le neurone biologique.

Les réseaux de neurones biologiques sont capables d'accomplir facilement certaines fonctions telles que la mémorisation, l'apprentissage par l'exemple, la généralisation, la reconnaissance des formes et le traitement du signal. À partir du principe que le comportement intelligent provient de la structure et du fonctionnement des neurones biologiques, des recherches ont conduit au développement des neurones formels, également appelés neurones artificiels [3].

2.7 Le perceptron

Le perceptron est le tout premier réseau de neurones artificiels évolutif, capable d'apprentissage. Son objectif initial était de reconnaître des lettres de l'alphabet à l'aide de cellules photoélectriques en tant que capteurs.

En réalité, le perceptron est une fonction mathématique. Les données d'entrée (x) sont multipliées par des coefficients de poids (w), et le produit obtenu est une valeur numérique. Cette valeur peut être positive ou négative. Le neurone artificiel s'active si la valeur est positive, c'est-à-dire que le poids calculé des données d'entrée dépasse un certain seuil [18].

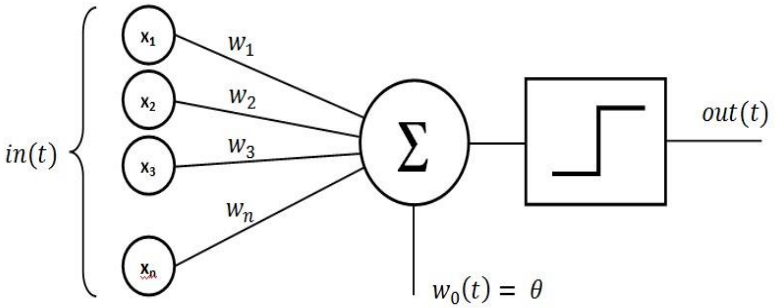

Figure2.7. Réseau monocouche.

2.8 Perceptron multicouche

Le perceptron multicouche ou encore multilayers perceptron en anglais est le premier réseau de neurones à avoir trouvé de nombreuses applications pratiques telles que la reconnaissance de fleurs, la détection de fraudes, etc. Il est composé de plusieurs couches de neurones, y compris une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque neurone

dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante. Chaque connexion entre les neurones est associée à un poids qui détermine l'influence de l'activation d'un neurone sur l'activation des neurones de la couche suivante [19]

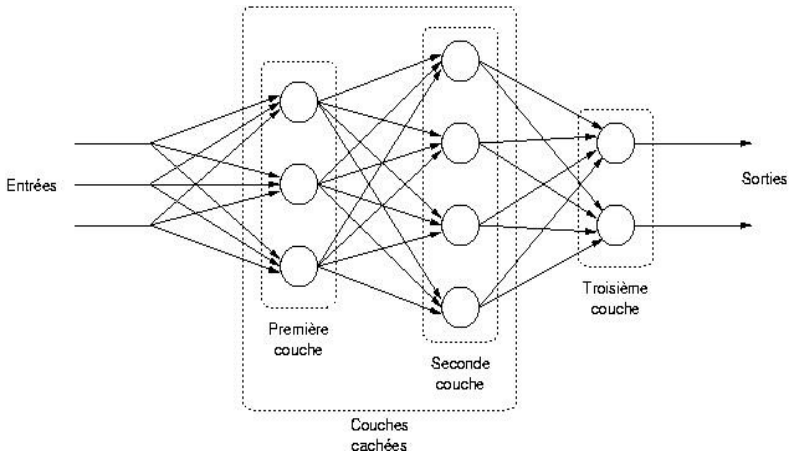

Figure 2.8. Perceptron multicouche.

2.9 Réseaux de neurones artificiels (RNA)

Un réseau de neurones artificiels est un système informatique inspiré du fonctionnement du cerveau humain, utilisé dans les ordinateurs dotés de capacités d'intelligence artificielle. Les réseaux de neurones artificiels sont conçus en se basant sur la structure des neurones biologiques du cerveau humain. Ils sont composés d'au moins deux couches de neurones - une couche d'entrée et une couche de sortie - et comprennent généralement des couches intermédiaires appelées "couches cachées ou hidden layer". La complexité du problème à résoudre détermine le nombre de couches nécessaires dans le réseau de neurones artificiels.

Chaque couche est composée d'un grand nombre de neurones artificiels spécialisés.

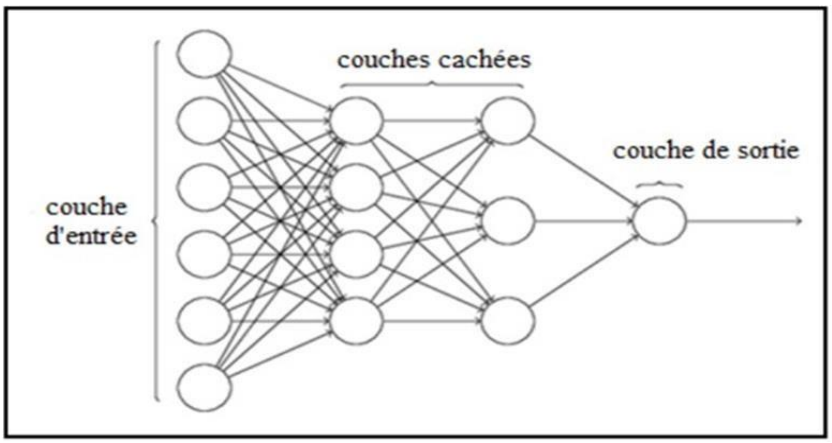

Figure 2.9. Réseau de neurone artificiel.

Les réseaux de neurones trouvent des applications dans divers domaines. Ils sont utilisés pour résoudre des problèmes de classification, de régression et même pour estimer la densité de probabilité. Ils sont employés à la fois dans l'apprentissage supervisé et non supervisé, ainsi que dans les modèles discriminatifs et génératifs. En résumé, ils constituent une famille de modèles extrêmement flexibles et puissants qui méritent d'être explorés davantage [20].

2.9.1 Le fonctionnement des réseaux de neurones artificiels

Le réseau de neurones artificiels repose sur l'utilisation de plusieurs processeurs qui fonctionnent en parallèle. Ces processeurs sont organisés en couches. La première couche est responsable de la réception des entrées de données brutes. Chaque couche subséquente reçoit ensuite les sorties d'informations provenant de la couche précédente. La dernière couche génère les résultats du système. Pour traiter des problèmes plus complexes, il est souvent nécessaire d'avoir plusieurs couches. Chaque neurone possède une valeur spécifique qui détermine quelle information peut être transmise dans le système. La fonction d'activation est utilisée pour calculer la valeur de sortie de chaque neurone. Ce calcul détermine combien de neurones doivent être activés pour résoudre le problème. Un algorithme est ensuite créé associant un résultat à chaque entrée. L'algorithme permet à l'ordinateur d'apprendre à partir de nouvelles informations qu'il reçoit.

Le réseau de neurones permet à l'ordinateur d'analyser des exemples et d'acquérir des capacités pour effectuer des tâches spécifiques. Ces exemples sont généralement étiquetés. Ce processus a permis aux ordinateurs de reconnaître des objets dans des images, parfois de manière plus performante que le cerveau humain lui-même.

Tout comme le cerveau humain, les réseaux de neurones artificiels ne peuvent pas être directement programmés, mais doivent apprendre en étudiant et en analysant des exemples [21].

2.9.2 Les types de réseaux de neurones artificiels

Les types de réseaux de neurones sont généralement classifiés en fonction du nombre de couches nécessaires entre l'entrée des données et la sortie finale. De plus, le type de réseau est déterminé en fonction du nombre de nœuds cachés présents dans chaque modèle. On tient également compte du nombre d'entrées et de sorties de chaque nœud [21].

a) Réseaux de neurones à propagation avant

Le réseau de neurones de base est connu sous le nom de Feedforward. Les données de ce type de réseau se propagent directement depuis les entrées jusqu'aux nœuds de traitement. Ensuite, ils vont directement aux sorties [21].

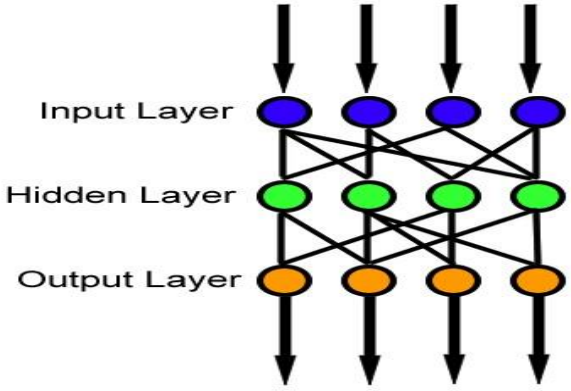

Figure 2.10. Réseaux de neurones à propagation avant.

b) Réseaux de neurones convolutives

Les réseaux de neurones convolutifs également connus sous le nom de CNN, sont utilisés pour détecter des motifs simples à l'intérieur d'une image afin d'identifier son contenu en effectuant des recoupements. Leur usage est de plus en plus répandu dans une variété de domaines, comme la reconnaissance faciale et la numérisation de texte. Les CNN comprennent au moins cinq couches et le résultat s'étend d'une couche à l'autre [21].

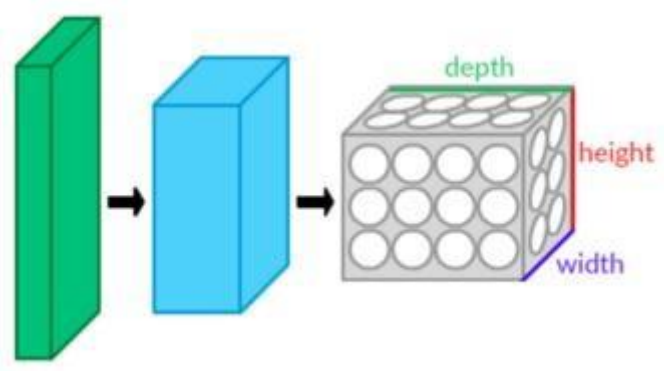

Figure 2.11. Réseau neuronal convolutif.

c) Réseaux de neurones récurrents

Les réseaux neuronaux récurrents (RNN) sont une variante très importante de réseaux neuronaux, largement utilisés dans le traitement du langage naturel. Ce qui distingue les RNN,

c'est leur capacité à effectuer la même tâche pour chaque élément d'une séquence, où la sortie dépend des calculs précédents. On peut également dire que les RNN possèdent une "mémoire" qui capture des informations sur ce qui a été calculé jusqu'à présent. En théorie, les RNN peuvent utiliser des informations provenant de séquences de longueur arbitraire, mais en pratique ils se limitent souvent à examiner uniquement les étapes récentes. Les RNN sont une classe de réseaux neuronaux qui permettent aux prédictions antérieures d'être utilisées comme entrées grâce à l'utilisation d'états cachés. La structure typique d'un RNN est représentée dans la Figure 2.12 [22].

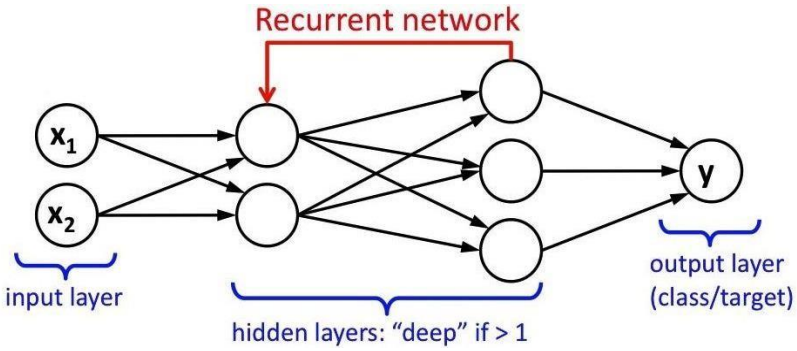

Figure 2.12. Architecture des réseaux de neurones récurrents.

2.10 Réseaux de neurones convolutionnels

2.10.1 Définition

Un réseau de neurones à convolution (CNN) est un type spécialisé de réseau neuronal qui utilise l'opération mathématique de convolution. Les CNN sont considérés comme l'un des meilleurs algorithmes d'apprentissage pour effectuer la convolution, ce qui permet d'extraire des fonctionnalités utiles à partir de données corrélées localement. Contrairement à la multiplication matricielle générale utilisée dans d'autres types de réseaux neuronaux, la convolution est employée dans au moins une couche des CNN.

Dans un CNN, les données sont traitées à travers des couches convolutives, des unités de traitement non linéaires (fonctions d'activation) et des couches de sous-échantillonnage. Les noyaux convolutifs effectuent la convolution pour extraire des caractéristiques des données d'entrée, qui sont ensuite envoyées aux unités de traitement non linéaires. Ces unités non

linéaires aident à apprendre des abstractions et introduisent de la non-linéarité dans l'espace des fonctionnalités. Cette non-linéarité permet d'obtenir différentes activations pour différentes réponses [23].

2.10.2 Architecture des CNN

Les réseaux de neurones convolutifs (CNN) comprennent différentes couches, chacune jouant un rôle spécifique dans le traitement des données d'entrée. Habituellement, la première couche est une couche de convolution qui extrait les caractéristiques de bas niveau. Par la suite, une couche de sous-échantillonnage réduit la taille spatiale de la carte d'entités. On empile alors plusieurs couches de convolution et de sous-échantillonnage pour en extraire des caractéristiques plus abstraites. Enfin, les caractéristiques extraites sont transmises à une couche totalement connectée afin de réaliser la classification finale. Les CNN peuvent également utiliser des techniques telles que la normalisation par lot, la régularisation et le dropout pour améliorer les performances du modèle et éviter le surapprentissage [18].

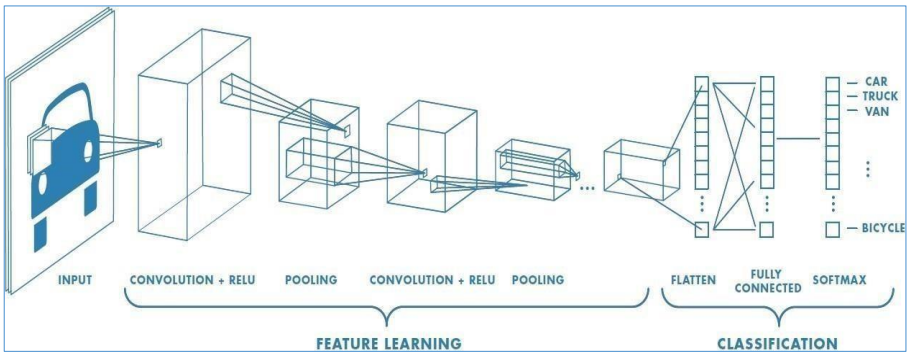

Figure 2.13. Réseau de neurones avec de nombreuses couches convolutives [23].

2.10.3 Les déférentes couches des CNN

Les réseaux de neurones convolutifs (Convolutional Neural Networks ou CNN) sont composés de plusieurs couches qui permettent d'extraire des caractéristiques significatives à partir des données d'entrée. L'architecture exacte des couches peut varier en fonction du problème et des objectifs spécifiques.

a) Couche de convolution

Couche principale du réseau CNN, elle joue un rôle vital dans l'extraction des caractéristiques. La figure 2.9 illustre l'opération de la convolution qui prend une image qui représente une matrice compose de pixels de 0 et 1, elle est une dimension de 7×7 pour appliquer

un calcul à l'aide d'un noyau ou d'un filtre (3x3), pour produire une matrice les Dimensions (5x5) ou ce que l'on appelle une carte des caractéristiques [24].

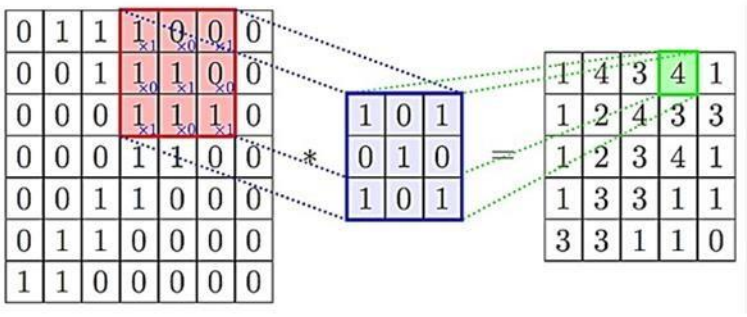

Figure 2.14. Couche de convolution [22].

b) Couche d'activation

Après les couches convolutionnels, une fonction d'activation est appliquée à chaque carte de caractéristiques pour introduire des non-linéarités dans le modèle. La fonction d'activation la plus couramment utilisée est la fonction ReLU (Rectified Linear Unit), qui remplace les valeurs négatives par des zéros et conserve les valeurs positives.

c) Couche de pooling

Dans la plupart des cas, une couche convolutive est suivie d'une couche de regroupement. L'objectif principal de cette couche est de réduire la taille des cartes d'entités convolutives pour réduire le coût de calcul. Ceci est réalisé en réduisant les connexions entre les couches et en fonctionnant indépendamment sur chaque carte d'entités. Selon la méthode utilisée, il existe différents types d'opérations de pooling [18].

Max pooling : Il sélectionne les éléments les plus importants de la fiche technique.

Les éléments importants de la carte des fonctionnalités sont stockés dans la couche maxpooling résultante. C'est la méthode la plus populaire car elle produit les meilleurs résultats [18].

Average pooling : Il s'agit de calculer la moyenne pour chaque région de la carte fonctionnalité. La somme des éléments de la partie prédéfinie est calculée dans Sum pooling [18].

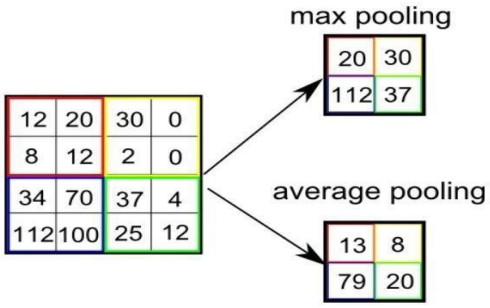

Figure 2.15. Pooling moyen & pooling maximal.

d) Couche entièrement connecter (Fully connected layer)

La couche entièrement connectée est similaire au réseau entièrement connecté dans les modèles conventionnels. La sortie de la première phase (comprenant la convolution et la mise en commun répétitive) est introduite dans la couche entièrement connectée, et le produit scalaire du vecteur de poids et du vecteur d'entrée est calculé afin d'obtenir la sortie finale [29].

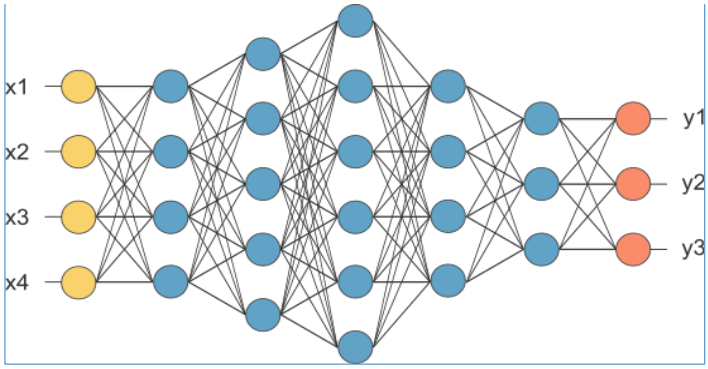

Figure 2.16. Couche fully-connected [22].

e) Couche de sortie :

La dernière couche d'un CNN est la couche de sortie, qui génère des prédictions ou des probabilités associées à différentes classes ou catégories. Selon le problème, cette couche peut utiliser une fonction d'activation appropriée, telle qu'une fonction softmax pour la classification multi-classes [18].

2.10.4 Les fonctions d'activation

Les fonctions d'activation d'un réseau neuronal convolutif (CNN) jouent un rôle important dans la capacité du réseau à introduire la non-linéarité et à capturer des relations complexes

entre les données. Le choix de la fonction de déclenchement dépend du problème spécifique et peut être testé pour de meilleurs résultats. Les fonctions d'activation couramment utilisées dans les CNN sont [25] :

a) Fonction d'activation ReLU (Rectified Linear Unit)

Cette fonction est définie comme f(x) = max (0, x), où x est l'entrée du neurone. Il conserve les valeurs positives et rejette les valeurs négatives, introduisant ainsi une non-linéarité dans le réseau.

b) Fonction d'activation Sigmoïde

La fonction sigmoïde est une fonction non linéaire qui transforme les valeurs d'entrée dans une plage de 0 à 1. Elle est définie comme f(x) = 1 / (1 + exp(-x)). Il est couramment utilisé dans les classes de sortie CNN pour la classification binaire ou probabiliste.

c) Fonction d'activation Softmax

La fonction softmax est utilisée pour classer plusieurs classes. Il convertit les valeurs d'entrée en une distribution de probabilité où la somme de toutes les sorties est égale à 1. Il est souvent utilisé dans la couche finale des CNN pour générer les probabilités de classe.

2.10.5 Les paramètres des CNN

Dans les réseaux de neurones à convolution (CNN), il est nécessaire de déterminer plusieurs paramètres pour chaque couche, tels que le nombre de couches de convolution, de couches de correction ReLU, de couches de pooling et de couches entièrement connectées. De plus, il est important de spécifier les paramètres pour chaque couche de convolution et de pooling.

Pour une couche de convolution, trois paramètres sont utilisés pour la dimensionner. Le premier est le nombre de noyaux de convolution, qui détermine combien de filtres seront appliqués à l'entrée. Le deuxième paramètre est le pas de chevauchement, qui détermine le décalage entre l'application des filtres. Le troisième paramètre est la marge à zéro (zéro padding), qui ajoute des zéros autour des bords de la carte de caractéristiques afin de maintenir sa taille identique à celle de l'image d'entrée. Cependant, il peut être souhaitable de ne pas ajouter de zéros et de réduire ainsi la taille de la carte des caractéristiques.

En ce qui concerne la couche de pooling, celle-ci est définie par la taille de la fenêtre de traitement et le pas de chevauchement. En pratique, une fenêtre de (2x2) avec un pas de 1 est souvent choisie, ce qui signifie que la fenêtre se déplace d'un pixel à la fois lors du pooling [25].

2.10.6 L'entraînement des CNN

L'entraînement d'un réseau neuronal, tel qu'un réseau de neurones à convolution (CNN), consiste à ajuster les poids du réseau afin qu'il puisse effectuer des prédictions précises. Initialement, les poids du CNN sont définis de manière aléatoire. Pendant l'entraînement, le CNN est alimenté avec un ensemble de données étiquetées, où chaque entrée est associée à une classe spécifique. Le CNN traite chaque entrée en utilisant les valeurs actuelles des poids, puis compare la sortie obtenue avec l'étiquette de classe réelle de cette entrée. Si la sortie ne correspond pas à l'étiquette de classe, des ajustements sont effectués sur les poids du réseau pour améliorer la correspondance entre la sortie et l'étiquette de classe. Cette correction des poids est réalisée grâce à une technique appelée rétropropagation [25].

La rétropropagation est un mécanisme qui optimise le processus d'ajustement des poids en calculant les gradients de l'erreur entre la sortie prédite et l'étiquette réelle, puis en propageant ces gradients de manière rétrograde à travers le réseau pour mettre à jour les poids de manière appropriée. Ce processus de rétropropagation facilite les ajustements nécessaires pour améliorer la précision du CNN. L'entraînement du CNN se déroule sur plusieurs itérations appelées '' epochs''. À chaque epoch, le CNN parcourt l'ensemble de données d'entrée à plusieurs reprises, effectuant des ajustements progressifs des poids pour améliorer sa capacité de classification et de prédiction correcte des classes [25].

Une fois l'entraînement terminé, un ensemble de données de test est utilisé pour évaluer les performances du CNN. Cet ensemble de données de test est composé des entrées étiquetées qui n'ont pas été utilisées dans le processus d'entraînement. Chaque entrée du jeu de test est introduite dans le CNN, et sa sortie est comparée à l'étiquette de classe réelle. Cela permet de mesurer la précision et la capacité de généralisation du CNN. Il est important de noter que si un CNN présente une précision élevée sur les données d'apprentissage mais une mauvaise précision sur les données de test, cela peut indiquer un surapprentissage. Le surapprentissage se produit lorsque le réseau s'est adapté de manière trop spécifique aux données d'entraînement et ne généralise pas bien sur de nouvelles données. Cela peut se produire lorsque la taille de l'ensemble de données est limitée [25].

2.11 Les avantage des CNN dans le domaine de RAL

Les réseaux de neurones convolutifs (CNN) présentent plusieurs avantages dans le domaine de la reconnaissance du locuteur [18].

Capacité à extraire des caractéristiques pertinentes : Les CNN sont particulièrement efficaces pour extraire des caractéristiques discriminantes à partir de données audios. Les couches de convolution des CNN sont capables de détecter des motifs et des structures spécifiques dans les signaux vocaux, ce qui permet de capturer des informations importantes pour l'identification du locuteur.

Invariance aux variations : Les CNN sont conçus pour rester inchangés pour certaines variations dans les données. Dans le cas de la reconnaissance du locuteur, cela signifie qu'ils peuvent reconnaître le locuteur même si l'enregistrement diffère par la hauteur, la vitesse de la parole, le stress, etc. Cette immuabilité améliore la robustesse du modèle et permet d'obtenir de meilleurs résultats.

Réduction de la dimensionnalité : CNN utilise des couches de sous-échantillonnage (Agrégation) pour réduire la taille des entités extraites. Cela réduit la dimensionnalité des données, facilitant le traitement et la classification ultérieurs. Cette réduction de taille est particulièrement intéressante dans le cas de la reconnaissance du locuteur où les enregistrements vocaux peuvent être de durée variable.

Apprentissage automatique des caractéristiques : Contrairement aux méthodes traditionnelles qui nécessitent une extraction manuelle des caractéristiques, CNN peut automatiquement apprendre les caractéristiques distinctives à partir des données brutes. Cela évite de dépendre de l'expérience et des connaissances d'experts, et permet au modèle d'identifier les caractéristiques les mieux adaptées à la reconnaissance du locuteur.

2.12 Machine learning VS deep learning

Le Machine Learning et le Deep Learning sont deux concepts liés à l'intelligence artificielle, mais ils présentent des différences importantes. Voici les principales distinctions entre le Machine Learning et le Deep Learning :

Machine learning	**Deep learning**
-Utilise des algorithmes pour apprendre à partir de données structurées.	-Utilise des réseaux de neurones artificiels pour apprendre à partir de données non structurées.
-Peut être supervisé, c'est-à-dire qu'il nécessite des données étiquetées, ou non supervisé, où les données ne sont pas étiquetées.	- Fait partie intégrante du Machine Learning, en se concentrant sur des modèles d'apprentissage automatique profonds et complexes.
-Peut être utilisé pour résoudre une grande variété de problèmes, tels que la classification, la régression, la prédiction, la détection d'anomalies, etc.	- Convient aux problèmes de reconnaissance d'images, de reconnaissance vocale, de traduction automatique, etc….
-Convient au traitement de données structurées et semi-structurées.	- Traite des données non structurées, telles que des images, des vidéos, des textes, des sons, etc.
-Peut être utilisé pour traiter de grandes quantités de données.	- Requiert une puissance de calcul plus élevée et des ressources matérielles spécialisées, telles que les unités de traitement graphique (GPU).
-Nécessite des compétences en extraction de fonctionnalités et en sélection d'algorithmes appropriés.	- Peut automatiquement extraire des fonctionnalités utiles à partir des données, ce qui réduit la nécessité d'une extraction manuelle de fonctionnalités.

Tableau 2.2. La différence entre machine et deep learning.

Pour résumer l'apprentissage profond implique des algorithmes qui peuvent apprendre plusieurs niveaux de représentation afin de modéliser les rapports complexes entre les données. Bien que l'apprentissage automatique fonctionne avec des fonctionnalités associées qui sont généralement extraites manuellement des entrées, ensuite, ces fonctionnalités sont utilisées pour créer un modèle qui exécute par exemple une tâche de classification. C'est pourquoi cette

extraction manuelle de caractéristiques est à la fois difficile et coûteuse. Pour l'apprentissage profond, les caractéristiques associées sont automatiquement extraites des données saisies, comme le montre la figure 2.12 [31].

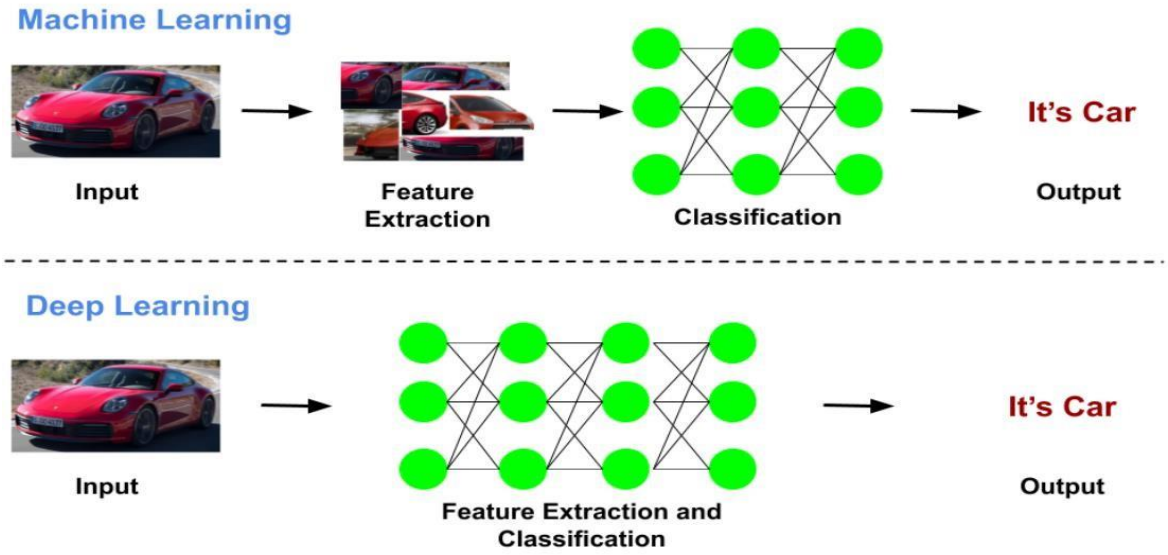

Figure 2.17. La différence entre ML et DL.

2.13 Conclusion

3 Dans ce chapitre Nous avons présenté le domaine de l'apprentissage profond, commencerons par une introduction générale à l'intelligence artificielle et à l'apprentissage automatique, soulignant l'importance de l'apprentissage profond et des CNN. Nous aborderons ensuite les fondements des réseaux de neurones artificiels, en mettant l'accent sur les différentes couches qui composent les CNN et leur fonction respective dans le traitement des données d'entrée. Ensuite, nous mettrons en évidence les avantages distincts des CNN dans le domaine de la reconnaissance du locuteur et leur capacité à extraire automatiquement des caractéristiques pertinentes à partir de données audios. Ainsi que les architectures et les méthodes d'entraînement spécifiques qui contribuent à l'efficacité et à la précision de ces modèles dans le domaine de la reconnaissance du locuteur. Enfin, on à vue les distinctions entre la machine et le deep learning.

3.1 Introduction

Le deep learning, également connu sous le nom d'apprentissage profond, est une branche de l'intelligence artificielle (IA) qui exploite les réseaux de neurones pour analyser des facteurs variés en utilisant une structure similaire au système neuronal humain. Le deep learning joue un rôle essentiel dans le domaine des technologies de l'information et de la communication en raison de ses vastes applications. Dans ce chapitre on va présenter un système de reconnaissance du locuteur à l'aide des réseaux de neurones convolutives (CNN).

3.2 Objectif de travail

L'objectif de notre recherche est de développer un système de reconnaissance du locuteur utilisant les réseaux de neurones convolutifs (CNN). Nous espérons tirer parti de la puissance de cette architecture d'apprentissage en profondeur pour identifier avec précision et automatiquement une personne à partir de sa voix. Nous avons rassemblé un ensemble de données composé d'enregistrements de plusieurs locuteurs, couvrant une variété de voix, de styles de parole et de conditions d'enregistrement. Chaque enregistrement est étiqueté avec l'identité de l'orateur correspondant. On a divisé notre ensemble de données en ensembles d'apprentissage qui sera utilisé pour l'apprentissage du modèle, ensemble validation qui sera utilisé pour ajuster les hyperparamètres et ensemble de test pour évaluer les performances finales du modèle.

Nous avons choisi d'utiliser une architecture CNN connue pour son efficacité dans le traitement des données séquentielles telles que les signaux audio. Notre réseau neuronal convolutif se compose de plusieurs couches convolutives et de sous-échantillonnage suivies de couches entièrement connectées pour la classification.

3.3 Environnement utilisés

3.3.1 Langage python

Python est l'un des langages de programmation les plus couramment utilisés par les professionnels de la donnée, il a été inventé par Guido van Rossum, la première version de python est sortie en 1991. Ses applications ne sont pas limitées à la Data Science mais peuvent également être utilisées pour développer des logiciels, écrire des algorithmes ou encore gérer l'infrastructure web d'un réseau social [24].

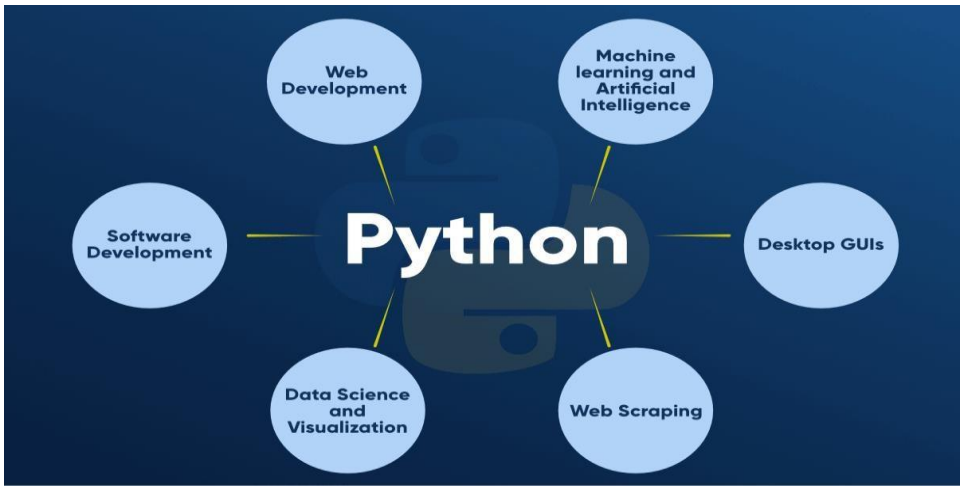

Figure 3.1. Les domaines d'applications de python.

Python est à la fois simple et puissant, il nous permet de créer des scripts très faciles à écrire et possédant de nombreuses bibliothèques, nous pouvons nous attaquer à des projets plus ambitieux. C'est un code de programmation qui se déroule en ligne, c'est-à-dire qu'il n'est pas nécessaire de le compiler avant de l'exécuter. Il est polyvalent, ça veut dire qu'il operate sur différents systèmes d'exploitation : Raspberry Pi, Mac OS X, Linux, Android, iOS et même sur les mini-ordinateurs [24].

- **Pour quoi python**

Python est également très populaire dans le secteur de l'IA en raison de la qualité de ses bibliothèques et de la puissance de calcul qu'elles possèdent. De nombreux chercheurs et développeurs utilisent Python pour exécuter et tester de nouveaux algorithmes d'apprentissage automatique et de deep learning, en particulier grâce aux plateformes telles que Tensorflow et PyTorch. En utilisant le Python, il est possible de créer rapidement des configurations complexes et performantes, qui sont donc un choix parfait pour les initiatives d'IA de grande échelle [24].

De plus, le codage en Python permet aux développeurs de se concentrer sur l'objectif de leurs programmes sans avoir à déboguer constamment leur code pour corriger les erreurs de syntaxe. Il permet de réaliser tout type de projet avec un niveau d'exigence élevé. C'est pourquoi de grandes entreprises comme Google, la Nasa, Microsoft ou Instagram (pour n'en citer que quelques-unes) utilisent Python [25].

3.3.2 La plateforme Kaggle

Kaggle est une plateforme web fondée en 2010 par Anthony Gold bloom et Ben Hamner. Il a ensuite été acquis par Google le 8 mars 2017 et la communauté compte plus de 536 000 membres actifs dans 194 pays. Kaggle est devenue la plus grande communauté de science des données au monde, recevant près de 150 000 soumissions par mois. La plate-forme est devenue un moyen extrêmement populaire pour les data scientistes de mettre en valeur leurs compétences et d'être reconnus sur le terrain.

La plateforme a gagné la confiance de grandes entreprises de science des données telles que Wal-Mart et Facebook. Il offre aux professionnels des données et autres développeurs la possibilité de participer à des compétitions et des défis d'apprentissage automatique, d'écrire et de partager du code et d'héberger des ensembles de données [25].

Figure 3.2. Le Logo de la plateforme Kaggle.

En outre, dans les notebooks Kaggle, nous avons la possibilité d'activer un GPU à tout moment et nous avons l'autorisation d'utiliser activement le GPU pendant un maximum de 30 heures par semaine. Le GPU mis à disposition par Kaggle est le Nvidia Tesla P100, doté de 16 Go de mémoire. Les ensembles de données de Kaggle sont en open source, cependant, afin de connaître les utilisations autorisées de ces ensembles de données, nous devons vérifier leur licence respective. Il se peut que certains ensembles de données ne puissent pas être utilisés dans des publications académiques ou à des fins commerciales [26].

3.3.3 Matlab

Matlab est un environnement de programmation orienté vers le calcul numérique. Avec son langage de script éponyme, il offre un éditeur permettant d'exécuter des séquences de commandes encapsulées dans des fonctions. Matlab est spécifiquement conçu pour des tâches telles que l'analyse de données, la visualisation de graphiques, la génération de matrices, le développement d'algorithmes et la création d'applications. Il est interopérable avec des langages tels que Python, C/C++, Java et Fortran, et est optimisé pour le calcul en parallèle.

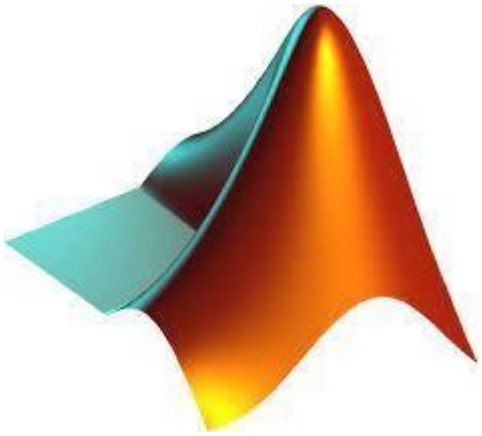

Figure 3.3. Le Logo de Matlab.

3.4 Définition des bibliothèques utilisées

3.4.1 Tensorflow

TensorFlow est un Framework d'apprentissage automatique open source développée par l'équipe Google Brain. Initialement publié en 2015, il est devenu l'un des Framework les plus populaires pour créer et déployer des modèles d'apprentissage automatique. TensorFlow fournit une plate-forme flexible et efficace pour créer des modèles d'apprentissage en profondeur, y compris la prise en charge du calcul CPU et GPU, du calcul distribué et de la différenciation automatique.

L'une des principales caractéristiques de TensorFlow est sa capacité à définir et à exécuter des graphes de calcul, permettant une parallélisation et une optimisation efficace des algorithmes d'apprentissage automatique. TensorFlow fournit également une API de haut niveau pour la création et la formation de modèles, et une API de bas niveau qui permet un contrôle plus précis du graphe de calcul.

3.4.2 keras

Keras est une bibliothèque open source d'apprentissage automatique et d'apprentissage en profondeur écrite en Python. Il fournit une interface de haut niveau facile à utiliser pour la création, la formation et la notation de modèles d'apprentissage automatique.

L'une des principales caractéristiques de Keras est sa facilité d'utilisation. Il simplifie le processus de développement de modèles d'apprentissage automatique et fournit un haut niveau d'abstraction qui masque des détails techniques complexes. Cette bibliothèque permet aux développeurs de créer des réseaux de neurones artificiels, des réseaux de neurones convolutionnels (CNN), des réseaux de neurones récurrents (RNN) et d'autres architectures de modèles populaires avec seulement quelques lignes de code.

3.4.3 Numpy

Le terme NumPy est en fait un acronyme pour "Numerical Python". C'est une bibliothèque open source pour le langage Python. NumPy fournit des fonctions puissantes pour effectuer des opérations mathématiques et statistiques en Python, y compris des opérations sur des matrices et des tableaux multidimensionnels. La bibliothèque est particulièrement appréciée pour sa capacité à effectuer efficacement des opérations telles que la multiplication matricielle.

3.5 Description et caractéristiques de la base de données

La base de données utilisée pour l'entraînement de notre modèle de reconnaissance automatique des locuteurs est “Speaker Recognition Dataset", elle contient des enregistrements audio de discours de différents locuteurs. Les caractéristiques de la base de données sont :

- Fréquence d’échantillonnage : Les enregistrements audio ont une fréquence d'échantillonnage de 16 000 Hz.
- Structure des fichiers : Les fichiers audio sont au format WAV.
- Répertoire des données : Les données sont stockées dans un répertoire spécifié («. /speaker-recognition-dataset/16000_pcm_speeches/audio").
- Structure des répertoires : Les enregistrements audio sont organisés dans des répertoires par locuteur. Chaque locuteur a un répertoire séparé contenant ses enregistrements audio.

- Étiquettes : La base de données contient 5 locuteurs (Magaret Tarcher, Benjamin Netanyau, Jens Stoltenberg, Julia Gillard et Nelson_Mandela). Chaque locuteur est associé à une étiquette unique pour l'entraînement et la classification.

3.6 Prétraitement des données

a. Chargement des dépendances

Les bibliothèques nécessaires sont importées pour le prétraitement des données, telles que TensorFlow, NumPy, os, shutil, pathlib, IPython.display et subprocess.

```
import tensorflow as tf
import os
from os.path import isfile, join
import numpy as np
import shutil
from tensorflow import keras
from pathlib import Path
from IPython.display import display, Audio
import subprocess
```

Figure 3.4. Importation des bibliothèques nécessaires.

Cela permet d'avoir accès aux fonctionnalités requises pour manipuler et traiter les données.

b. Copie des données

Nous avons utilisé la commande ‘CP’ pour copier les données audio du répertoire "/input/speaker-recognition-dataset" vers le répertoire courant "./speaker-recognitiondataset/16000_pcm_speeches/audio".

c. Définition des paramètres

Valid split

Cette variable représente la proportion de données qui sont utilisées comme ensemble de validation lors de l'entraînement de notre modèle. Dans notre cas, 10% des données seront utilisées pour la validation.

- **Shuffle seed**

C'est la valeur de la graine utilisée pour le mélange aléatoire des données lors de l'entraînement est égale 43.

- **Sample rate**

La variable ‘sample rate’ spécifie le taux d'échantillonnage des enregistrements audio de notre base de données qui est 16000 échantillons par seconde. Cela indique la fréquence à laquelle le signal audio a été enregistré.

- **Scale**

La variable `scale` définie par 0.5 représente un facteur de mise à l'échelle qui peut être appliqué aux données audios, généralement utilisé pour normaliser ou ajuster les amplitudes des échantillons audio.

- **Batch size**

Cette variable spécifie la taille des lots (Batches) utilisés lors de l'entraînement du modèle. Ici, la taille du lot est de 128. Cela signifie que 128 échantillons sont traités simultanément avant la mise à jour des poids.

- **Epochs**

La variable `epochs` définit le nombre d'epochs (itérations complètes) pendant les quelle nous avons entraîné le modèle. Un epoch correspond à une passe complète sur l'ensemble des données d'entraînement. Notre l'entraînement du modèle se fera sur 15 epochs.

```
valid_split = 0.1

shuffle_seed = 43

sample_rate = 16000

scale = 0.5

batch_size = 128

epochs = 15
```

Figure 3.5. Les paramètres de notre modèle.

Ces variables sont utilisées pour contrôler divers aspects du processus d'entraînement de notre modèle de reconnaissance de locuteurs.

d. Organisation des données

Nous avons organisé les fichiers en déplaçant les dossiers des locuteurs dans le répertoire audio_path et les dossiers de bruit dans le répertoire noise_path.

```
data_directory = "./speaker-recognition-dataset/16000_pcm_speeches"
audio_folder = "audio"
noise_folder = "noise"

audio_path = os.path.join(data_directory, audio_folder)
noise_path = os.path.join(data_directory, noise_folder)
```

Figure 3.6. Data directories.

Cette organisation permet de structurer les données de manière appropriée pour la suite du traitement.

e. Chargement des chemins des fichiers de bruit

Nous avons parcouru les fichiers de bruit et les avons chargés en tant qu'échantillons de bruit, Ce qui permet d'avoir accès aux échantillons de bruit pour les utiliser ultérieurement lors de l'ajout de bruit aux données d'entraînement.

f. Conversion des fichiers audio de bruit

Les fichiers audios de bruit sont convertis pour s'assurer qu'ils ont tous le même taux d'échantillonnage (16 000 Hz) en utilisant la commande FFmpeg. Cela garantit la cohérence du taux d'échantillonnage pour tous les échantillons de bruit.

g. Chargement des échantillons de bruit

Les échantillons de bruit sont chargés en utilisant la fonction ‘load_noise_sample(Path)’. Les échantillons sont stockés dans la liste noises après vérification du taux d'échantillonnage correct. Cela permet d'avoir accès aux échantillons de bruit pour les utiliser lors de l'ajout de bruit aux données d'entraînement.

h. Conversion des chemins et des étiquettes en un ensemble de données

Les chemins des fichiers audios et les étiquettes correspondantes sont convertis en un jeu de données TensorFlow. Cette étape permet de préparer les données pour l'entraînement du modèle en associant chaque chemin de fichier audio à son étiquette correspondante.

i. Ajout de bruit aux données d'entraînement

Le bruit est ajouté aux données d'entraînement en utilisant la fonction ‘add_noise’ (audio, noises, scale). Un échantillon de bruit aléatoire est sélectionné pour chaque échantillon audio, et la valeur maximale de l'échantillon audio est adaptée à la valeur maximale du bruit. Le paramètre scale contrôle l'intensité du bruit ajouté.

j. Conversion des données audio en domaine fréquentiel

Les signaux audios sont convertis du domaine temporel au domaine fréquentiel en utilisant la transformation de Fourier (FFT). Cette étape permet de représenter les données audios dans un format adapté à l'entraînement du modèle basé sur des caractéristiques fréquentielles.

3.7 Construction du modèle

Couche d'entrée

Une couche d'entrée prenant des séquences de taille (8000, 1) (8000 échantillons avec 1 dimension). Cela correspond probablement à un signal ou une séquence unidimensionnelle.

Couches convolutives ○ Conv1d_15, conv1d_16, conv1d_17

Ce sont des couches de convolution 1D avec 128 filtres chacune. Elles sont utilisées pour extraire des caractéristiques des données d'entrée.

○ Activation_10, activation_11

Ce sont des couches d'activation (ReLU) qui introduisent de la non-linéarité dans les sorties des couches convolutives.

Connexions résiduelles (Conv1d_14)

Une autre couche de convolution 1D avec 128 filtres appliquée sur l'entrée initiale. Cela sert de connexion résiduelle et est ajouté aux sorties des couches convolutives précédentes (add_4).

Couche d'activation (Activation_12)

Une couche d'activation est appliquée sur la sortie de la connexion résiduelle.

Couche de pooling (Max_pooling1d_4)

Une couche de max pooling 1D réduisant de moitié la dimension des caractéristiques en conservant les valeurs maximales dans chaque fenêtre.

Couche de pooling moyenne (Average_pooling1d)

Une couche de pooling moyenne 1D qui réduit davantage la dimensionnalité des caractéristiques en prenant la moyenne des valeurs dans chaque fenêtre. Cela réduit le nombre total de caractéristiques.

Couche de mise en forme (Flatten)

Cette couche convertit les caractéristiques en une forme linéaire pour les entrées de la couche dense suivante.

Couches complètement connectées

- **Dense :** Une couche entièrement connectée avec 256 neurones et une fonction d'activation 'relu', cette couche apprend des combinaisons linéaires des caractéristiques précédentes pour effectuer une classification plus complexe.
- **Dense_1 :** Une autre couche dense avec 128 neurones qui réduit davantage la dimensionnalité des caractéristiques.

Couche de sortie

La couche de sortie est une couche dense avec une fonction d'activation softmax, avec 5 neurones correspondant au nombre de classes de sortie. Cela indique que le modèle est utilisé pour une classification multi classe avec 5 classes.

Layer (type)	Output Shape	Param #	Connected to
input (InputLayer)	[(None, 8000, 1)]	0	
conv1d_15 (Conv1D)	(None, 8000, 128)	512	input[0][0]
activation_10 (Activation)	(None, 8000, 128)	0	conv1d_15[0][0]
conv1d_16 (Conv1D)	(None, 8000, 128)	49280	activation_10[0][0]
activation_11 (Activation)	(None, 8000, 128)	0	conv1d_16[0][0]
conv1d_17 (Conv1D)	(None, 8000, 128)	49280	activation_11[0][0]
conv1d_14 (Conv1D)	(None, 8000, 128)	256	input[0][0]
add_4 (Add)	(None, 8000, 128)	0	conv1d_17[0][0] conv1d_14[0][0]
activation_12 (Activation)	(None, 8000, 128)	0	add_4[0][0]
max_pooling1d_4 (MaxPooling1D)	(None, 4000, 128)	0	activation_12[0][0]
average_pooling1d (AveragePooli	(None, 1333, 128)	0	max_pooling1d_4[0][0]
flatten (Flatten)	(None, 170624)	0	average_pooling1d[0][0]
dense (Dense)	(None, 256)	43680000	flatten[0][0]
dense_1 (Dense)	(None, 128)	32896	dense[0][0]
output (Dense)	(None, 5)	645	dense_1[0][0]

Total params: 43,812,869
Trainable params: 43,812,869
Non-trainable params: 0

Figure 3.7. Architecture du modèle CNN.

Le modèle contient un total de 43,812,869 paramètres entraînables, c'est-à-dire que tous les paramètres du modèle sont entraînables, ce qui signifie qu'ils seront mis à jour lors de l'entraînement.

3.8 L'entrainement et évaluation de modèle

L'entraînement du modèle commence par l'appel à la méthode model.fit (), qui prend en entrée les données d'entraînement (train_ds) et les données de validation (valid_ds). Le nombre d'époques est défini par la variable epochs.

```
history = model.fit(
    train_ds,
    epochs=epochs,
    validation_data=valid_ds,
    callbacks=[earlystopping_cb, mdlcheckpoint_cb],
)
```

Figure 3.8. Entraînement de modèle.

Pendant chaque époque, le modèle est entraîné sur les données d'entraînement en utilisant la méthode fit (). On peut voir que chaque époque est composée de plusieurs étapes (Steps) représentées par le nombre total de lot (batch) dans le jeu de données. Pour chaque étape, le modèle calcule la perte (loss) et l'exactitude (accuracy) sur les données d'entraînement.

```
Epoch 1/15
53/53 [==============================] - 888s 17s/step - loss: 14.7456 - accuracy: 0.5904
- val_loss: 0.1989 - val_accuracy: 0.9093

/opt/conda/lib/python3.7/site-packages/keras/utils/generic_utils.py:497: CustomMaskWarnin
g: Custom mask layers require a config and must override get_config. When loading, the cus
tom mask layer must be passed to the custom_objects argument.
  category=CustomMaskWarning)

Epoch 2/15
53/53 [==============================] - 883s 17s/step - loss: 0.2119 - accuracy: 0.9237 -
val_loss: 0.0776 - val_accuracy: 0.9747
Epoch 3/15
53/53 [==============================] - 873s 16s/step - loss: 0.1401 - accuracy: 0.9487 -
val_loss: 0.0919 - val_accuracy: 0.9627
Epoch 4/15
53/53 [==============================] - 859s 16s/step - loss: 0.1026 - accuracy: 0.9606 -
val_loss: 0.0714 - val_accuracy: 0.9680
Epoch 5/15
53/53 [==============================] - 861s 16s/step - loss: 0.0952 - accuracy: 0.9673 -
val_loss: 0.1074 - val_accuracy: 0.9653
Epoch 6/15
53/53 [==============================] - 871s 16s/step - loss: 0.0863 - accuracy: 0.9701 -
val_loss: 0.0592 - val_accuracy: 0.9733
Epoch 7/15
53/53 [==============================] - 861s 16s/step - loss: 0.0728 - accuracy: 0.9739 -
```

Figure 3.9. Evaluation du modèle.

Et après chaque époque, le modèle est évalué sur les données de validation. On peut observer la perte de validation (val_loss) et l'exactitude de validation (val_accuracy) pour chaque époque (Figure 3.7). De plus, deux rappels (callbacks) sont utilisés pendant l'entrainement earlystopping_cb et mdlcheckpoint_cb. 'Earlystopping_cb' permet d'arrêter l'entraînement prématurément si la performance du modèle sur les données de validation cesse de s'améliorer. Mdlcheckpoint_cb permet de sauvegarder le meilleur modèle obtenu pendant l'entraînement.

Après l'achèvement des époques, le modèle est évalué une dernière fois sur les données de validation en utilisant la méthode model. Evaluate ().

```
print("Accuracy of model:",model.evaluate(valid_ds))
```

```
24/24 [==============================] - 33s 1s/step - loss: 0.0662 - accuracy: 0.9840
Accuracy of model: [0.06619565933942795, 0.984000027179718]
```

Figure 3.10. La dernière évaluation de modèle.

On peut voir que la perte finale est de 0.0662 et l'exactitude finale est de 0.9840.

3.9 Test & prédictions du modèle

Pour effectuer les prédictions sur le jeu de données de test. Tout d'abord, nous créons le jeu de données de test en utilisant les chemins des fichiers audios de validation et leurs étiquettes correspondantes. Cela nous permet d'avoir un ensemble distinct d'échantillons pour évaluer les performances du modèle. Ensuite, nous mélangeons le jeu de données de test pour assurer une répartition aléatoire des échantillons. Cette étape est importante pour éviter tout biais potentiel dans l'ordre des échantillons lors de l'évaluation.

Nous divisons le jeu de données de test en lots de taille (batch size). Cela nous permet de traiter les échantillons par lots plutôt que tous en même temps, ce qui peut être plus efficace en termes de mémoire et de calcul. Pour augmenter la robustesse du modèle aux variations de bruit, nous appliquons une augmentation de bruit aux échantillons audio du jeu de données de test. Cela est réalisé en utilisant la fonction "add_noise", qui ajoute du bruit aux audios en se basant sur un ensemble prédéfini de bruits et une échelle spécifiée.

Ensuite, nous itérons sur le jeu de données de test et effectuons les prédictions. Pour chaque lot d'audios et d'étiquettes, nous convertissons les audios en représentations de transformée de Fourier à l'aide de la fonction audio_to_fft. La transformée de Fourier nous permet de représenter les signaux audios en termes de leurs composantes de fréquence. Une fois les audios convertis en représentations de transformée de Fourier, nous utilisons le modèle préalablement entraîné pour prédire les étiquettes des échantillons audio.

Ensuite, nous sélectionnons aléatoirement un certain nombre d'échantillons (défini par SAMPLES_TO_DISPLAY=10) à afficher. Pour chaque échantillon, nous comparons l'étiquette réelle avec la prédiction faite par le modèle. Si l'étiquette réelle et la prédiction sont identiques, nous affichons le message "Welcome", indiquant que le modèle a correctement identifié le locuteur. Si l'étiquette réelle et la prédiction diffèrent, nous affichons le message "Sorry", Indiquant que le modèle a fait une prédiction incorrecte. Ensuite, nous affichons le nom du locuteur prédit par le modèle.

```
Speaker: Jens_Stoltenberg       Predicted: Jens_Stoltenberg
Welcome
The speaker is Jens_Stoltenberg
Speaker: Magaret_Tarcher        Predicted: Magaret_Tarcher
Welcome
The speaker is Magaret_Tarcher
Speaker: Magaret_Tarcher        Predicted: Magaret_Tarcher
Welcome
The speaker is Magaret_Tarcher
Speaker: Jens_Stoltenberg       Predicted: Jens_Stoltenberg
Welcome
The speaker is Jens_Stoltenberg
Speaker: Magaret_Tarcher        Predicted: Magaret_Tarcher
Welcome
The speaker is Magaret_Tarcher
Speaker: Magaret_Tarcher        Predicted: Magaret_Tarcher
Welcome
The speaker is Magaret_Tarcher
Speaker: Jens_Stoltenberg       Predicted: Jens_Stoltenberg
Welcome
The speaker is Jens_Stoltenberg
Speaker: Julia_Gillard  Predicted: Julia_Gillard
Welcome
The speaker is Julia_Gillard
Speaker: Jens_Stoltenberg       Predicted: Jens_Stoltenberg
Welcome
The speaker is Jens_Stoltenberg
Speaker: Nelson_Mandela Predicted: Nelson_Mandela
Welcome
The speaker is Nelson_Mandela
```

Figure 3.11. Prédiction du modèle.

Nous pouvons voir que le modèle a fait des prédictions correctes pour plusieurs échantillons, ce qui est indiqué par le message "Welcome". Pour ces échantillons, le nom du locuteur prédit correspond à l'étiquette réelle. Cela suggère que le modèle a réussi à reconnaître ces locuteurs avec précision.

3.10 Résultat de classification

a. La précision (Accuracy)

La précision (Accuracy) est une mesure couramment utilisée pour évaluer les performances d'un modèle de classification. Elle représente le rapport entre le nombre d'échantillons correctement classés et le nombre total d'échantillons dans l'ensemble de validation.

Notre modèle a été entraîné sur un ensemble de données comprenant des échantillons audios de différentes classes. Lors de la phase d'évaluation sur les données de validation, Les résultats de classification indiquent que le modèle atteint une précision (Accuracy) de 98% sur les données de validation.

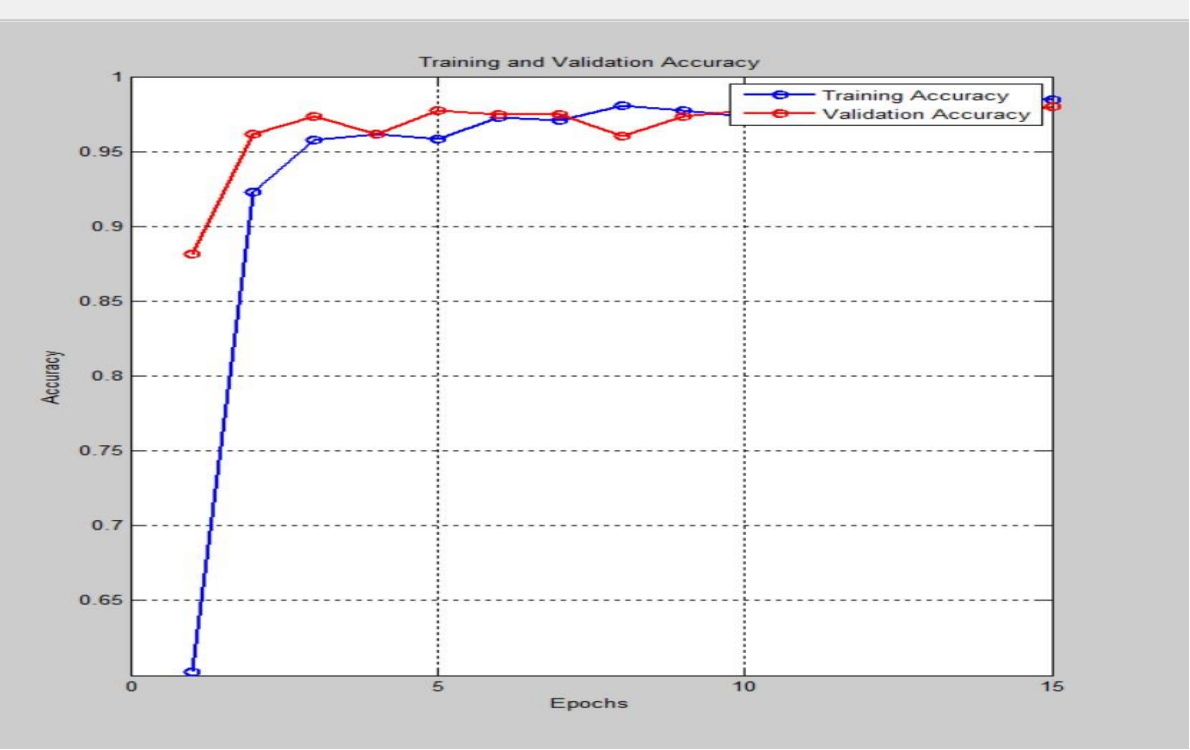

Figure 3.12. Graphe de l'entrainement et la validation de la précision (sur MATLAB).

Cela signifie que le modèle parvient à classifier correctement 98% des échantillons de la validation ce qui indique une performance élevée et une capacité du modèle à bien généraliser sur de nouvelles données.

b. La perte (Loss)

La perte est une mesure de l'erreur du modèle lors de la prédiction des étiquettes. L'objectif est de minimiser cette perte pendant l'entraînement afin d'améliorer les performances du modèle et d'obtenir des prédictions plus précises.

Après chaque époque d'entraînement de notre modèle, la perte (Loss) est diminuée, comme indiqué précédemment sa valeur est indiquée environ 0.0914.

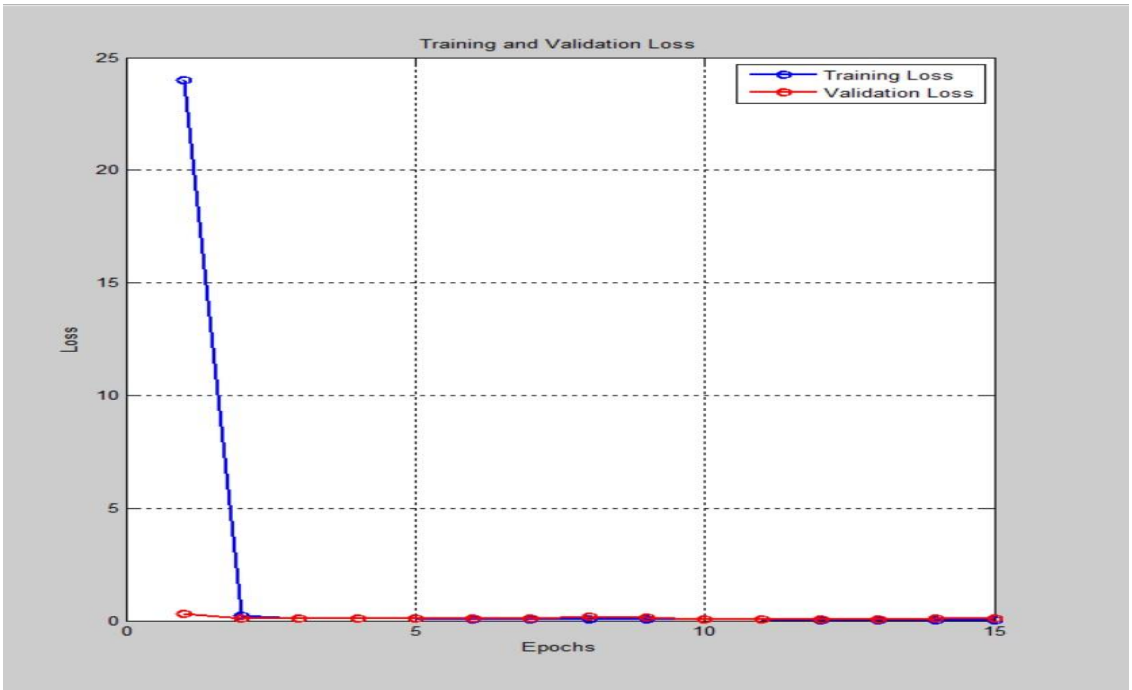

Figure 3.13. Graphe de l'entrainement et la validation de la perte(sur MATLAB).

Cela signifie que le modèle a réussi à minimiser l'erreur et à faire des prédictions précises.

3.11 Conclusion

Dans ce chapitre nous avons présenté notre étude sur la reconnaissance automatique des locuteurs avec un réseau de neurones convolutifs (CNN). Nous avons commencé par prétraiter les données audios afin de les représenter sous forme de données exploitables par le modèle CNN. Ensuite, nous avons entraîné le modèle sur un ensemble de données comprenant différentes voix et avons évalué ses performances sur un ensemble de test distinct.

Les résultats obtenus sont encourageants, avec une précision de 98% dans la prédiction des locuteurs. Cela signifie que notre modèle est capable de reconnaître avec précision l'identité des locuteurs à partir de leurs enregistrements.

Conclusion générale

Le travail que nous avons fait au long de cette mémoire de fin d'étude s'inscrit dans le cadre de la reconnaissance automatique du locuteur par Deep Learning en utilisant les réseaux de neurones convolutifs (CNN). Notre objectif était de développer un système capable d'identifier et de reconnaître les locuteurs à partir de leurs voix, en exploitant les puissantes capacités de traitement des réseaux de neurones convolutifs.

Dans le premier chapitre, nous avons introduit le contexte général de la reconnaissance automatique du locuteur, en analysant les différentes tâches impliquées dans ce domaine, telles que l'identification et la vérification du locuteur, mais aussi la production et la reconnaissance de la parole et du langage. Nous avons également présenté les applications de la reconnaissance du locuteur dans divers domaines.

Le deuxième chapitre s'est concentré spécifiquement sur l'intelligence artificielle et les réseaux de neurones artificiels. Nous avons abordé les différents types d'apprentissage ainsi que les types de réseaux des neurones en mettent l'accent aux réseaux de neurones convolutifs. Nous avons également expliqué en détail l'architecture des CNN, ses différentes couches et ses différents paramètres. Nous avons aussi présenté l'entraînement des réseaux de neurones convolutifs et leurs avantages dans le domaine de la reconnaissance du locuteur. Enfin, on a étudié la différence entre l'apprentissage machine et l'apprentissage profond.

Le dernier chapitre de notre mémoire a porté sur l'implémentation et l'évaluation du système de reconnaissance du locuteur basé sur les CNN. Nous avons décrit la méthodologie expérimentale utilisée, y compris la collecte des données audio, la création des ensembles d'apprentissage et de test et d'évaluation, ainsi que leurs paramètres.

Les résultats obtenus ont démontré l'efficacité des réseaux de neurones convolutifs dans la reconnaissance du locuteur. Avec une précision de 98%, notre système a atteint des performances remarquables, ouvrant ainsi de nombreuses perspectives d'application dans des domaines tels que la sécurité, l'authentification et la surveillance

Bibliographie

[1] A. Amehray, « Rehaussement de bruitage perceptuel de la parole », thèse de doctorat, école nationale supérieure des télécommunications de Bretagne, 2009.

[2] S. K. Singh, P.C.Pandey, « Featues and Technique For Speaker Recognition », Seminar Report, page 5,6, Novembre 03.

[3] M. MOUSS Mohamed Djamel, « Intégration D'un Module De Reconnaissance De La Parole Au Niveau D'un système Audiovisuel – Application Téléviseur », thèse de doctorat, Université Batna 2, AVRIL 2021.

[4] M. Denis Jouve, « Reconnaissance du locuteur en milieux difficiles », thèse de doctorat, UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE, 18 Juillet 2017

[5] Y. AZIZA, « modélisation Ar et arma de la parole pour une vérification robuste du locuteur dans un milieu bruité en mode dépendant du texte », Mémoire de Magister, Université Ferhat Abbas, Sétif, 2013.

[6] H. Satori, M. Harti and N. Chenfour, « Système de Reconnaissance Automatique de l'arabe basé sur CMUSphinx », mémoire master, Dhar Mehraz Fès Morocco.

[7] Othman Lachhab, « Reconnaissance Statistique de la Parole Continue pour Voix Laryngée et Alaryngée », Université Mohammed V de Rabat (Maroc), 2017.

[8] Vincent Jousse, « Identification nommée du locuteur : exploitation conjointe du signal sonore et de sa transcription », mémoire master, Université du Maine, 2011.

[9] Michael F McTear, « Spoken dialogue technology: toward the conversational user interface. Springer Science & Business Media », article, page 3, 2004.

[10] M. A. Wissmann et K. M. Béring, « Automatique Language Identification », Speech Communication, article, page 4, 2001.

[11] Mr. Haddab, « reconnaissance automatique du locuteur par la méthode du taux passage par zéro », mémoire master, université Mouloud mamri de Tizi-Ouzou, 2007/2008

[12] Jin, Minho, and Yoo, ChangD, « SPEAKER VERIFICATION AND IDENTIFICATION », Korê Institut Avancé des Sciences et Technologies, République de Corée, 2004.

[13] Siwar ZRIBI BOUJELBENE, « Identification du Locuteur par Système Hybride GMMSMO », thèse, TUNISIA, March 22/26/2009.

[14] Dr. Clint Slatton, « A Speaker Verification System », thèse, Université de Florida,2006.

[15] A. Preti, « Surveillance de reseaux professionnels de communication par la reconnaissance du locuteur », Thèse, Université d‟Avignon et des Pays de Vaucluse, France, 2008.

[16] (consulté le 12/06/2023), disponible sur : https://www.editionseni.fr/open/mediabook.aspx?idR=f6e7a7353a3574180124387fa03fdcl,

[17] Amine Abdaoui, « Machine Learning », article, page 5, 1/7/2019.

[18] La Ryax Team, « Deep learning : comprendre les réseaux de neurones artificiels (artificial neural networks) », article, page 3, 2020.

[19] Dr. Ouarda ZEDADRA, « Système de prédiction de la consommation d'énergie basé Deep Learning », Mémoire master, Université de 8 Mai 1945, Septembre 2021.

[20] Pr. BILAMI Azeddine, « Apprentissage Incrémental & Machines à Vecteurs Supports », Université HADJ LAKHDAR – BATNA, 18 /12 /2013

[21] Guillaume Saint-Cirgue, « Apprendre la machine learning en une semaine », 2019.

[22] Houcine Noura & Khelifa Nadia, « classification des textures par les réseaux de neurones convolutifs », mémoire master, université mouloud Mammri tizi-ouzou, 2018/2019.

[23] M. Abderrahmane Adjila, « Détection d'activité vocale utilisant l'apprentissage profond », Mémoire master, Université de Ghardaïa, 2019/2020.

[24] Apprendre programmation cours python 3, (consulté le 16/06/2023) disponible sur : https://python.doctor/ .

[25] consulté le 18/06/2023, disponible sur : https://www.data-bird.co/blog/langagepython#toc-que-peut-on-faire-avec-python-

[26] consulté le 22/06/2023 disponible sur : https://datascientest.com/kaggle-tout-ce-quil-asavoir-sur-cette-plateforme .