République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb Blida 1

Faculté de technologie

Département d'Électronique

Filière: **Electronique**

Spécialité : **instrumentation**

Mémoire de Master

Thème

Système d'authentification du locuteur basé sur un Réseau neuronal résiduel (ResNet)

Encadré par : Farid Ykhlef

Présenté par :

KALLA AMIRA
&
ERRIRI RYM

Année Universitaire : 2023/2024

Remerciement

Tout d'abord, nous remercions Dieu de nous avoir donné la foi, la force et la persévérance nécessaires pour atteindre cet objectif. Sans Sa bénédiction, ce travail n'aurait pas été possible.

Nous exprimons notre gratitude particulière à notre encadrant, monsieur Yekhlef.F , pour son soutien indéfectible, ses conseils avisés et sa disponibilité tout au long de cette année universitaire. Sa confiance en notre potentiel et son dévouement à notre projet ont été une source inestimable d'encouragement.

Nous tenons à remercier chaleureusement nos enseignants, en particulier monsieur

Benselama.Z et madame Bougherira.N , qui ont bien voulu accepter d'examiner notre travail. Leurs commentaires constructifs et leurs suggestions précieuses ont grandement contribué à l'amélioration de ce mémoire.

Nos remerciements vont également à l'ensemble des professeurs de notre université « SAAD DEHLEB BLIDA » qui ont contribué de manière significative à notre formation académique et à notre développement intellectuel au cours de nos années d'études. Leurs enseignements et leur soutien ont été fondamentaux pour notre réussite.

Nous remercions également nos collègues et amis, qui ont été une source constante de motivation et de soutien. Leurs encouragements et leur camaraderie ont rendu ce parcours plus enrichissant et agréable.

Un grand merci à nos familles, pour leur amour, leur compréhension et leur patience tout au long de cette période. Leur soutien moral et matériel a été essentiel pour surmonter les défis rencontrés.

Enfin, nous adressons nos sincères remerciements à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail, que ce soit par leurs conseils, leur aide pratique ou leurs encouragements.

Merci à tous.

ملخص

أصبحت مصادقة المتحدث، والتي تسمى غالباً التعرف على الصوت، تقنية مهمة في العديد من المجالات، بما فيذلك الأمن. يهدف مشروعنا، الذي يحمل عنوان "نظام مصادقة المتحدث استنادًا إلى الشبكة العصبية المتبقية) ResNet("، إلى تطوير نموذج قادر على التعرف على أصوات أربعة قادة بارزين: جينس ستولتنبرغ، وجوليا جيلارد، ومارغريت تاتشر، ونيلسون مانديلا. استخدام معالجة الإشارات المتقدمة وتقنيات التعلم العميق. لهذا، قمنا بتصميم نموذج يعتمد على بنية تسمى ResNet)أو "الشبكة العصبية المتبقية"(. يعمل هذا النموذج على تحسين دقة المصادقة الصوتية للمتحدث ،ويقلل من مخاطر الاحتيال ويمكن استخدامه لمراقبة الاتصالات الحساسة، مما يوفر حلاً فعالاً وغير تدخلي للتحقق من الهوية.

Abstract

Speaker authentication, often called voice recognition, has become a crucial technology in many areas, including security. Our project, titled "Speaker Authentication System based on a Residual Neural Network (ResNet)", aims to develop a model capable of identifying the voices of four prominent leaders: Jens Stoltenberg, Julia Gillard, Margaret Thatcher and Nelson Mandela. Using advanced signal processing and Deep Learning techniques. For this, we designed a model based on an architecture called ResNet (or "Residual Neural

Network"). This model improves the accuracy of a speaker's voice authentication, reduces the risk of fraud and can be used to monitor sensitive communications, providing a nonintrusive and effective solution for identity verification.

Résumé

L'authentification du locuteur, souvent appelée la reconnaissance vocale, est devenue une technologie cruciale dans de nombreux domaines, notamment pour la sécurité. Notre projet, intitulé "Système d'authentification du locuteur basé sur un Réseau neuronal résiduel (ResNet)", vise à développer un modèle capable d'identifier les voix de quatre dirigeants éminents : Jens Stoltenberg, Julia Gillard, Margaret Thatcher et Nelson Mandela. En utilisant des techniques avancées de traitement du signal et de Deep Learning. Pour cela, nous avons conçu un modèle basé sur une architecture nommée ResNet (ou "Residual Neural Network"). Ce modèle améliore la précision de l'authentification vocale d'un locuteur, réduit les risques de fraude et peut être utilisé pour surveiller des communications sensibles, offrant ainsi une solution non intrusive et efficace pour la vérification d'identité.

Liste des figures

Liste des abréviations et acronymes

IA : Intelligente artificielle

ML : Machine learning

NLP : Naturel languageprocessing

GPS : Global Positioning System

CNN : Convolution neurial network

CAH : Classification Ascendante Hiérarchique

MLP : Multi-Layer Perceptron

ReLU : RectifiedLinear Unit

Tanh : Hyperbolic Tangent

ResNet : Residual Network

RAL : reconnaissance automatique du locuteur

TF : Transformée de fourier

RAP : Reconnaissance automatique de la

parole F0 : Fréquence fondamentale dB :

décibel

VAL : Vérification automatique du locuteur

IAL : Identification automatique du locuteur

IFT : Inverse fourier transformation

PCM : Pulse Code Modulation

GPU : Graphics Processing Unit

TPU : TensorProcessing Unit

FFT : Fastfourier transformation

Table de matière

Introduction générale

La reconnaissance vocale s'est imposée comme une technologie incontournable, tissant sa toile dans divers aspects de notre vie quotidienne, des assistants virtuels répondant à nos requêtes aux systèmes de transcription automatique facilitant la prise de notes. Parmi ses applications critiques, la sécurité se distingue comme un domaine prioritaire. Les systèmes de reconnaissance vocale, en authentifiant les utilisateurs, en surveillant les communications et en ajoutant une couche de vérification supplémentaire dans les environnements sensibles, contribuent à renforcer la sécurité. En conséquence, le développement de modèles de reconnaissance vocale robustes et précis s'avère crucial pour garantir l'efficacité et la fiabilité de ces systèmes.

Au cœur de notre projet réside le développement d'un modèle de reconnaissance vocale capable d'identifier les voix de quatre personnalités éminentes : Jens Stoltenberg, Julia Gillard, Margaret Thatcher et Nelson Mandela. S'appuyant sur des techniques de pointe en traitement du signal et en apprentissage profond, nous avons conçu un modèle basé sur l'architecture ResNet, réputée pour sa gestion efficace des gradients et sa capacité à capturer des caractéristiques complexes des signaux audio.

Ce modèle revêt une importance capitale dans le domaine de la sécurité, et ce à plusieurs égards. Premièrement, il permet une authentification vocale de haute précision, réduisant ainsi considérablement les risques de fraude et d'accès non autorisés. Deuxièmement, il offre la possibilité de surveiller et d'analyser des communications sensibles, contribuant ainsi à la détection d'anomalies ou de menaces potentielles. Enfin, dans des contextes où la discrétion est primordiale, un système de reconnaissance vocale performant constitue une solution non intrusive et efficace pour la vérification d'identité.

ChapitreI: Réseau résiduel

I.1 Introduction

Le deep learning est une branche de l'intelligence artificielle qui cherche à imiter le fonctionnement du cerveau humain pour traiter des données et accomplir des tâches complexes. Il tire son nom de l'utilisation de réseaux de neurones artificiels composés de plusieurs couches (d'où le terme "deep" signifiant profond en anglais), permettant d'apprendre des représentations de plus en plus abstraites des données.

Ces dernières années, le domaine du deep learning a connu un essor considérable, notamment grâce à l'augmentation de la puissance de calcul des ordinateurs et à la disponibilité de vastes quantités de données. Le deep learning a révolutionné de nombreux secteurs tels que la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale, et les systèmes de recommandation de contenu.

Les réseaux de neurones profonds sont capables d'apprendre à partir de données non structurées en identifiant des modèles complexes et en effectuant des prédictions précises. Leur flexibilité et leur capacité à généraliser les connaissances en font des outils puissants pour résoudre une grande variété de problèmes, allant de la classification d'images à la traduction automatique.

Dans ce chapitre, nous avons exploré en profondeur les concepts du deep learning, en nous concentrant particulièrement sur les réseaux de neurones convolutifs (CNN) et les architectures avancées telles que ResNet. Nous avons examiné les principes fondamentaux du deep learning, analysé le fonctionnement des CNN et détaillé les innovations et avantages spécifiques des réseaux résiduels comme ResNet. En particulier, nous avons appliqué ces techniques à l'authentification du locuteur, démontrant comment les architectures ResNet peuvent être utilisées pour améliorer la précision et la robustesse de l'identification et de la vérification des locuteurs. Ces discussions approfondies fournissent une base solide pour comprendre les techniques modernes de traitement des signaux vocaux et leur application dans le domaine de l'authentification du locuteur.

I.2 L’intelligence artificielle

L'intelligence artificielle (IA) représente l'une des avancées les plus fascinantes et potentiellement transformatrices de notre époque. En bref, l'IA désigne la capacité des machines à exécuter des tâches qui nécessitent traditionnellement l'intelligence humaine. Ces tâches peuvent inclure la reconnaissance de motifs, la prise de décision, la résolution de problèmes complexes et même la créativité.

L'histoire de l'IA remonte à plusieurs décennies, mais ces dernières années ont été marquées par des avancées spectaculaires grâce à des progrès dans les domaines de l'apprentissage automatique, de l'apprentissage profond et du traitement du langage naturel. Ces avancées ont permis à l'IA de transformer divers secteurs, de la santé à la finance, en passant par la technologie et les sciences.

I.3 Comment fonctionne l’IA

En général, les systèmes d'IA fonctionnent en ingérant de grandes quantités de données d'entraînement étiquetées, en analysant les données pour repérer des corrélations et des motifs, et en utilisant ces motifs pour faire des prédictions sur des états futurs. Ainsi, un chatbot qui est alimenté avec des exemples de texte peut apprendre à générer des échanges réalistes avec des personnes, ou un outil de reconnaissance d'images peut apprendre à identifier et décrire des objets dans des images en examinant des millions d'exemples. De nouvelles techniques d'IA générative, en constante amélioration, peuvent créer du texte, des images, de la musique et d'autres médias réalistes [1].

I.4 L’importance de l’IA

L'IA est importante pour son potentiel à changer notre manière de vivre, de travailler et de nous divertir. Elle a été efficacement utilisée dans les entreprises pour automatiser des tâches effectuées par des humains, notamment le travail de service client, la génération de leads, la détection de fraudes et le contrôle qualité. Dans de nombreux domaines, l'IA peut effectuer des tâches bien mieux que les humains. Notamment en ce qui concerne les tâches répétitives et détaillées, telles que l'analyse de grands nombres de documents juridiques pour s'assurer que les champs pertinents sont remplis correctement, les outils d'IA accomplissent souvent les tâches rapidement et avec relativement peu d'erreurs. En raison des vastes ensembles de données qu'elle peut traiter, l'IA peut également donner aux entreprises des aperçus sur leurs opérations dont elles pourraient ne pas avoir été conscientes. La population en expansion rapide des outils d'IA générative sera importante dans des domaines allant de l'éducation et du marketing à la conception de produits

I.5 Machine Learning

Le machine learning, ou apprentissage automatique en français, est une branche de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données sans être

explicitement programmés. En d'autres termes, au lieu de suivre des instructions spécifiques pour effectuer une tâche, les machines utilisant le machine learning peuvent apprendre à reconnaître des modèles dans les données et à prendre des décisions autonomes basées sur ces modèles.

Cette capacité à apprendre de manière autonome à partir de données est au cœur du machine learning. Elle permet aux machines de s'améliorer et de s'adapter à de nouvelles situations sans intervention humaine constante. Le machine learning est devenu omniprésent dans notre vie quotidienne, des recommandations de produits sur les sites de commerce électronique à la reconnaissance vocale sur nos smartphones, en passant par les systèmes de détection de fraude dans les services financiers.

I.6 Machine learning et analyse de données

Le Machine Learning est largement employé data science et en analyse de données. Il offre la possibilité de concevoir, expérimenter et mettre en œuvre des algorithmes d'analyse prédictive sur diverses données pour anticiper l'avenir. Le Machine Learning permet d'accélérer l'analyse de données et de rendre l'analyse plus précise en automatisant le développement de modèles analytiques. [2]

Il offre la possibilité d'attribuer aux machines des tâches essentielles dans l'analyse de données telles que la classification, le clustering ou la détection d'anomalies. Les algorithmes assimilent les informations et produisent des données statistiques, et peuvent progresser de manière autonome au fil du temps. Quand ils remarquent une modification des données, ils peuvent prendre des décisions sans avoir besoin d'intervention humaine.

Actuellement, il est nécessaire d'avoir un humain pour examiner les résultats des analyses réalisées par les algorithmes de Machine Learning. Son rôle consiste à donner une signification à ces résultats, ou encore à garantir que les données traitées par l'algorithme ne soient ni biaisées ni altérées.

I.7 Deep Learning

L'apprentissage profond, également appelé Deeplearning, est l'une des technologies majeures du Machine learning. Dans le domaine du Deep Learning, il s'agit d'algorithmes qui peuvent reproduire les actions du cerveau humain en utilisant des réseaux de neurones artificiels. Les réseaux comprennent plusieurs dizaines voire centaines de "couches" de neurones, chacune recevant et interprétant les données de la couche précédente.[3]

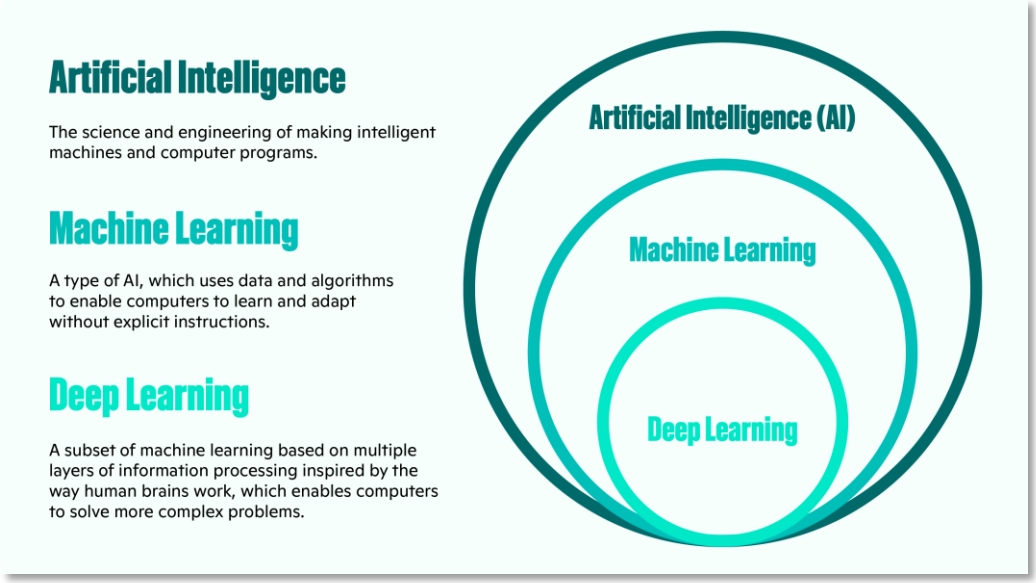


Figure 1 : AI , Machine learning , Deeplearning

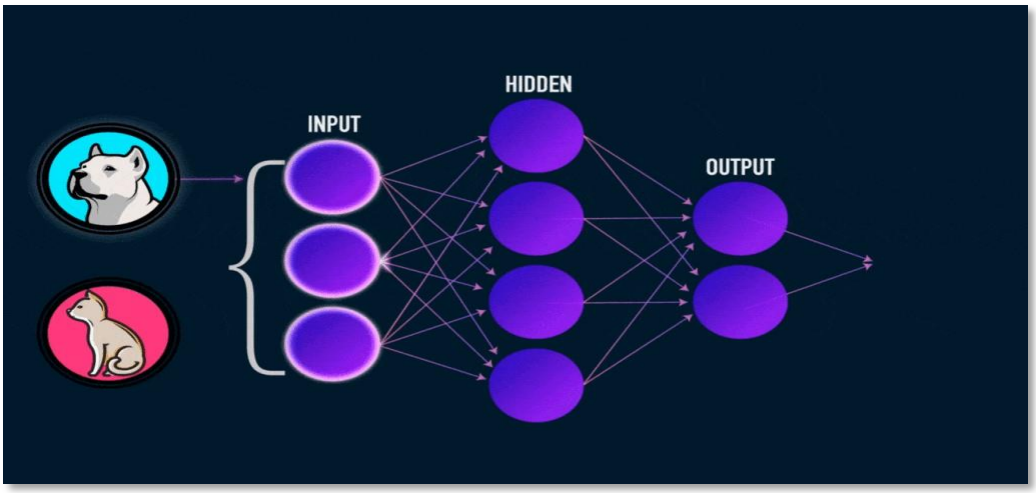


Figure 2 : Deep learning

Chaque neurone artificiel, symbolisé par un rond dans l'image précédente, peut être considéré comme un modèle linéaire. En reliant les neurones en couche, nousdéveloppons un modèle non-linéaire extrêmement complexe pour notre réseau de neurones. [3]

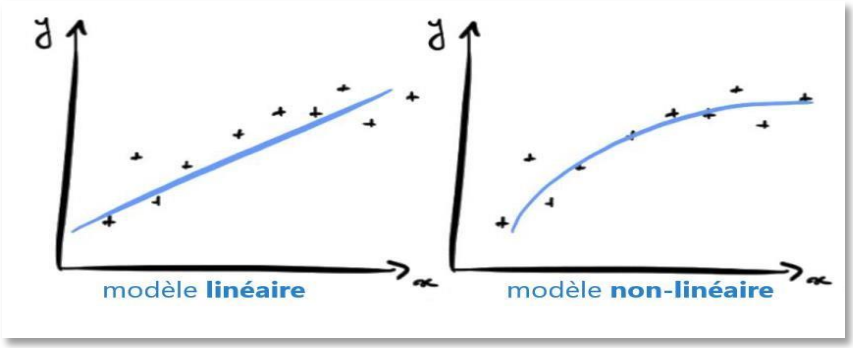


Figure 3 : Modèle linéaire et non-linéaire

Les modèles d'apprentissage profond sont généralement efficaces avec une grande quantité de données, tandis que les modèles d'apprentissage automatique plus traditionnels cessent de progresser après un point de saturation.

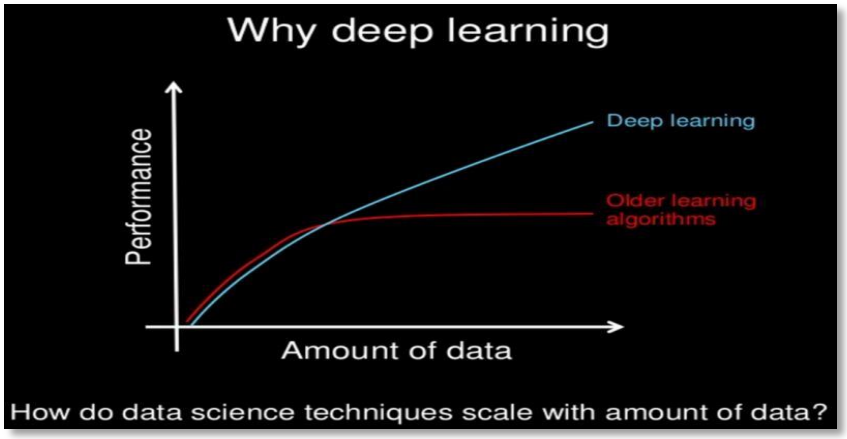


Figure 4 : Avantage de deeplearning

Avec l'avènement du big data et des composants informatiques de plus en plus puissants, les algorithmes de Deep Learning, qui nécessitent une grande puissance et des données, ont surpassé la plupart des autres techniques. Ils apparaissent disposés à résoudre de nombreux problèmes : repérer des visages, battre des joueurs de go ou de poker, autoriser la conduite de voitures autonomes ou encore identifier des cellules cancéreuses. [3]

I. 7.1 Principe de l'apprentissage profond

L'apprentissage profond est un paradigme d'apprentissage automatique inspiré de l'anatomie du cerveau humain. Cet apprentissage est associé à une structure algorithmique que l'on appelle un réseau de neurones. Le neurone biologique est une cellule complexe, mais seul son fonctionnement basique a servi d'inspiration au neurone informatique. [6]

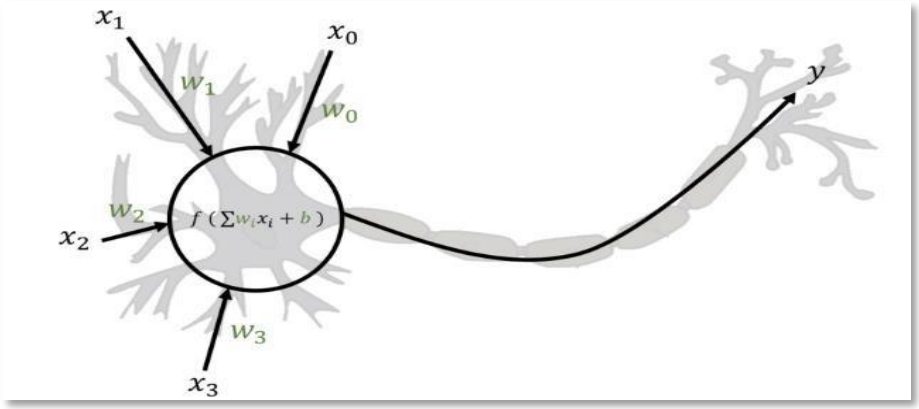


Figure 5 : Schéma d'un neurone informatique superposé à un schéma de neurone biologique

Un neurone biologique reçoit des signaux électriques d'autres neurones en amont via des points d'entrées (x0, x1, x2 et x3). Ces signaux sont accumulés à l'intérieur du corps du neurone, et si la somme de ces signaux dépasse un certain seuil, le neurone s'active et envoie à son tour un signal à des neurones en aval. Un neurone informatique approxime ces principes. Il accepte en

entrée un nombre fixe de nombres réels (représentant les signaux des neurones en amont) et produit en sortie (y) une valeur réelle (représentant le signal envoyé aux neurones en aval).

Cette sortie est calculée à partir des entrées via l'équation :

$$y = f\left(\sum_{i=0}^{n} w_i x_i + b\right)$$

qui représente l'accumulation de signaux électriques.

Les entrées sont multipliées par des valeurs w_i que l'on appelle les poids, qui représentent la force de la connexion entre ce neurone et les neurones en amont. La fonction f est une fonction dite d'activation. Il s'agit d'une fonction non-linéaire croissante.

Les premières implémentations de réseaux définissaient f comme une fonction seuil : f renvoie 1 si son argument est strictement positif et 0 sinon, mais d'autres fonctions furent proposées au fil des années comme la fonction unité linéaire rectifiée (RectifiedLinear Unit en anglais, ou ReLU).

Cette fonction représente l'activation du neurone si celui-ci a accumulé suffisamment de potentiel électrique. La valeur b, que l'on appelle le biais, représente l'appétence ou la résistance du neurone à s'activer. Les poids et le biais (souvent désignés collectivement sous le nom de « poids ») sont les paramètres du neurone : ce sont ces valeurs qui sont modifiées au cours d'un entraînement.

Un neurone informatique peut constituer à lui seul le support d'un algorithme d'apprentissage pour certaines tâches bien définies. En constituant un jeu d'entraînement composé de paires d'entrées réelles et des sorties binaires attendues, et en présentant séquentiellement ces exemples au neurone, il existe un algorithme d'entraînement qui définit comment modifier les poids de celui-ci afin de converger vers la classification attendue.
Cependant, la capacité d'un neurone unique est trop faible pour qu'il soit appliqué ailleurs que sur des cas jouets. Les algorithmes d'apprentissage profond sont basés sur des ensembles de neurones que l'on appelle des réseaux. On appelle « architecture » la structure selon laquelle les neurones sont reliés entre eux.

Les premières architectures s'appelaient les perceptron multicouche (Multi-Layer Perceptron en anglais, ou MLP). Dans un MLP, les neurones sont connectés à la fois parallèlement et séquentiellement selon une organisation en couches . La première couche est constituée d'un certain nombre de neurones qui prennent en entrée les données. Les sorties des neurones de cette couche servent d'entrée à une deuxième couche de neurones, et ainsi de suite, jusqu'à une dernière couche dont on identifie la sortie à proposition faite par le réseau pour l'étiquetage de l'entrée. Cette structure en couches est inspirée de l'architecture neuronale du cerveau humain .

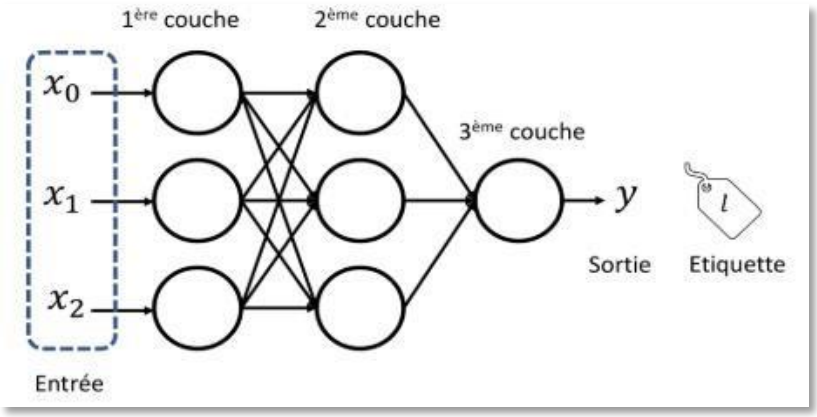


Figure 6 : Un perceptron multi-couche ou MLP composé de trois couches

I. 7.2 Méthode d'apprentissage

Comme pour le cerveau humain, les réseaux de neurones artificiels ne peuvent pas être programmés directement, mais doivent apprendre en étudiant et en analysant des exemples. Il existe trois méthodes d'apprentissage, sont [16] :

- **Apprentissage supervisé :** en ce qui concerne l'apprentissage supervisé, l'algorithme s'entraîne à partir de données étiquetées**.** Pour effectuer la tâche, il se modifie jusqu'à arriver à traiter le jeu de données pour obtenir le résultat attendu. Un résultat concret doit être défini pour chaque option d'entrée. L'apprentissage supervisé permet d'apporter des modifications au système afin d'optimiser le fonctionnement de l'algorithme.

  L'algorithme est guidé avec des connaissances préalables de ce que devraient être les valeurs de sortie du modèle. Par conséquent, le modèle ajuste ses paramètres de façon à diminuer l'écart entre les résultats obtenus et les résultats attendus. La marge d'erreur se réduit ainsi au fil des entraînements du modèle, afin d'être capable de l'appliquer à de nouveaux cas.

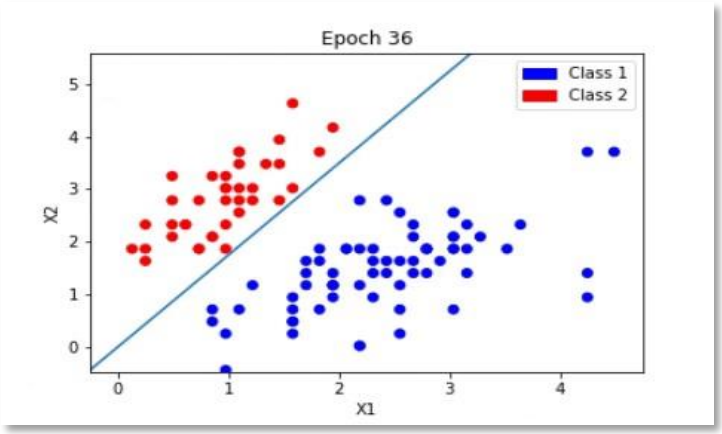


Figure 7 : Convergence d'un modèle - Apprentissage supervisé

- **Apprentissage non supervisé :** en apprentissage non supervisé, le réseau de neurones doit analyser un ensemble de données qui ne sont pas étiquetées**.** Une fonction spécifique lui indique à quel degré il s'éloigne ou s'approche du résultat attendu. Le réseau s'adapte ensuite. Le résultat de la tâche n'est donc pas déterminé à l'avance, mais le système pose lui-même son diagnostic à partir des informations obtenues. Le système s'appuie entre autres sur la théorie de la résonance adaptative.L'apprentissage non supervisé n'utilise pas de données étiquetées. Il est alors impossible à l'algorithme de calculer de façon certaine un score de réussite. Son objectif est donc de déduire les regroupements présents dans nos données.

Prenons l'exemple, d'un jeu de données de fleurs, on recherche à les regrouper en classes. Ici, nous ne connaissons pas l'espèce de la plante, mais nous voulons essayer de les regrouper, par exemple, si les formes des fleurs sont similaire alors elles sont en rapport avec une même plante correspondante.

Il existe deux principaux domaines de modèles dans l'apprentissage non-supervisées pour retrouver les regroupements : Les méthodes par partitionnement (les algorithmes des kmeans) , les méthodes de regroupement hiérarchique : classification ascendante hiérarchique (CAH).

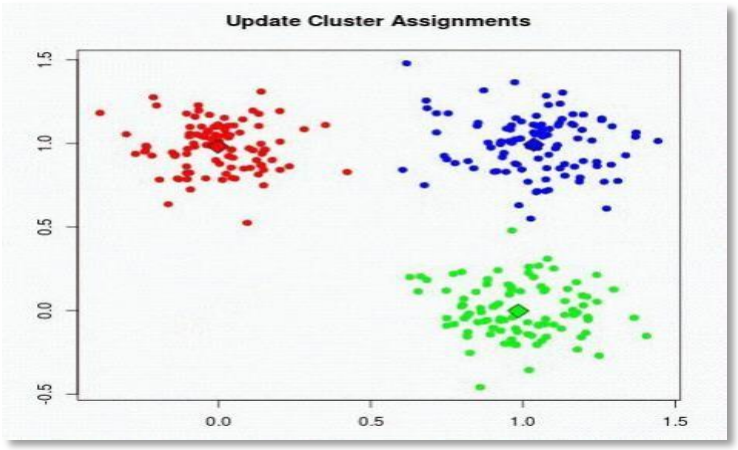


Figure 8 : Apprentissage non supervisé

- **Apprentissage renforcé :** Quant à l'apprentissage renforcé, il s'agit d'une méthode par laquelle on procède par renforcements et sanctions selon que les résultats sont positifs ou négatifs. Comme le cerveau humain qui apprend par essais et erreurs, le réseau de neurones apprend progressivement à mesure qu'il traite les données qu'on lui soumet.

I.8 Réseau de neurones convolutifs

Les réseaux de neuronesconvolutifs (CNN) sont des architectures spécifiquement conçues pour la vision par ordinateur. Ils sont composés de couches de convolution, de pooling et de

classification. Les couches de convolution filtrent l'entrée pour extraire des caractéristiques importantes, tandis que les couches de pooling réduisent la dimensionnalité de la sortie. Les CNN sont largement utilisés pour la classification d'images, la détection d'objets et d'autres tâches de traitement d'images [5].

I.8. 1 Architecture d'un CNN

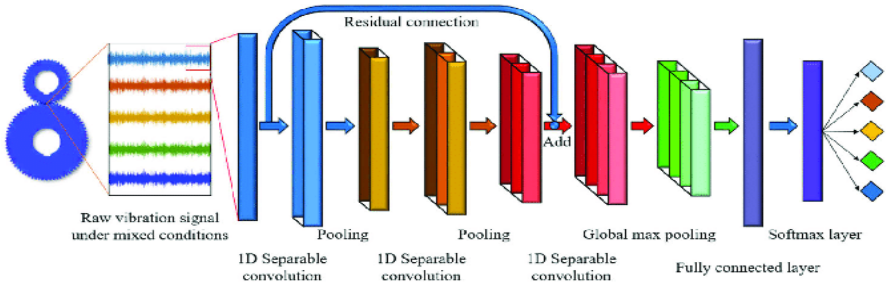
Les CNN désignent une sous-catégorie de réseaux de neurones et sont à ce jour un des modèles de classification d'images réputés être les plus performant. Leur mode de fonctionnement est à première vue simple : l'utilisateur fournit en entrée une image sous la forme d'une matrice de pixels.

Celle-ci dispose de 3 dimensions : deux dimensions pour une image en niveaux de gris ,une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert, Bleu).

Contrairement à un modèle MLP (Multi Layers Perceptron) classique qui ne contient qu'une partie classification, l'architecture du Convolutional Neural Network dispose en amont d'une partie convolutive et comporte par conséquent deux parties bien distinctes :

• **Une partie convolutive :** Son objectif final est d'extraire des caractéristiques propres à chaque image en les compressant de façon à réduire leur taille initiale. En résumé, l'image fournie en entrée passe à travers une succession de filtres, créant par la même occasion de nouvelles images appelées cartes de convolutions. Enfin, les cartes de convolutions obtenues sont concaténées dans un vecteur de caractéristiques appelé code CNN.

• **Une partie classification :** Le code CNN obtenu en sortie de la partie convolutive est fourni en entrée dans une deuxième partie, constituée de couches entièrement connectées appelées perceptron multicouche (MLP pour Multi Layers Perceptron). Le rôle de cette partie est de combiner les caractéristiques du code CNN afin de classer l'image. Pour revenir sur cette partie



.

Figure 9 : Schéma représentant l'architecture d'un CNN

I.8. 2 Parie convolutive

La convolution est une opération mathématique simple généralement utilisée pour le traitement et la reconnaissance d'images. Sur une image, son effet s'assimile à un filtrage dont voici le fonctionnement :

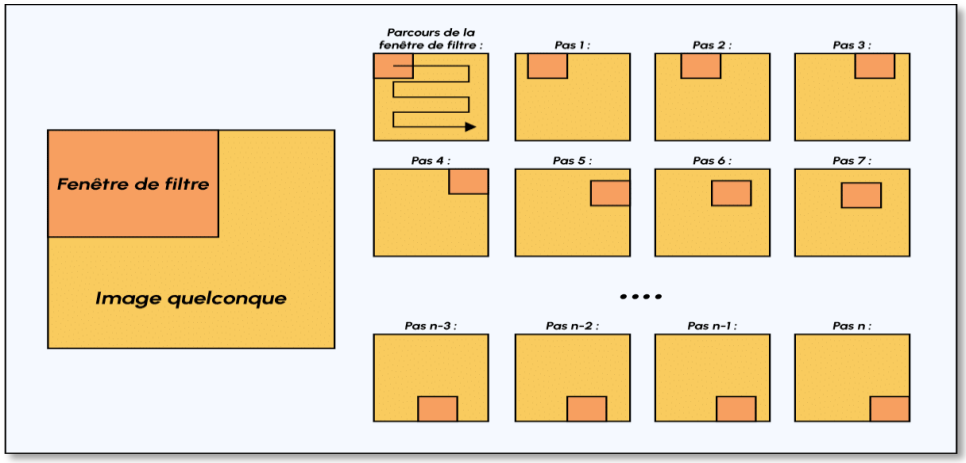


Figure 10 : Schéma du parcours de la fenêtre de filtre sur l'image

Dans un premier temps, on définit la taille de la fenêtre de filtre située en haut à gauche.

La fenêtre de filtre, représentant la feature, se déplace progressivement de la gauche vers la droite d'un certain nombre de cases défini au préalable (le pas) jusqu'à arriver au bout de l'image.

À chaque portion d'image rencontrée, un calcul de convolution s'effectue permettant d'obtenir en sortie une carte d'activation ou featuremap qui indique où est localisées les features dans l'image : plus la featuremap est élevée, plus la portion de l'image balayée ressemble à la feature.

I.8. 3 Méthode de sous échantillonnage « Le max pooling »

Le Max-Pooling est un processus de discrétisation basé sur des échantillons. Son objectif est de sous-échantillonner une représentation d'entrée (image, matrice de sortie de couche cachée, etc.) en réduisant sa dimension. De plus, son intérêt est qu'il réduit le coût de calcul en réduisant le nombre de paramètres à apprendre et fournit une invariance par petites translations (si une petite translation ne modifie pas le maximum de la région balayée, le maximum de chaque région restera le même et donc la nouvelle matrice créée restera identique).

Pour rendre plus concret l'action du Max-Pooling, voici un exemple : imaginons que nous avons une matrice 4×4 représentant notre entrée initiale et un filtre d'une fenêtre de taille 2×2 que nous appliquerons sur notre entrée. Pour chacune des régions balayées par le filtre, le maxpooling prendra le maximum, créant ainsi par la même occasion une nouvelle matrice de sortie où chaque élément correspondra aux maximums de chaque région rencontrée.

Illustrons le processus :

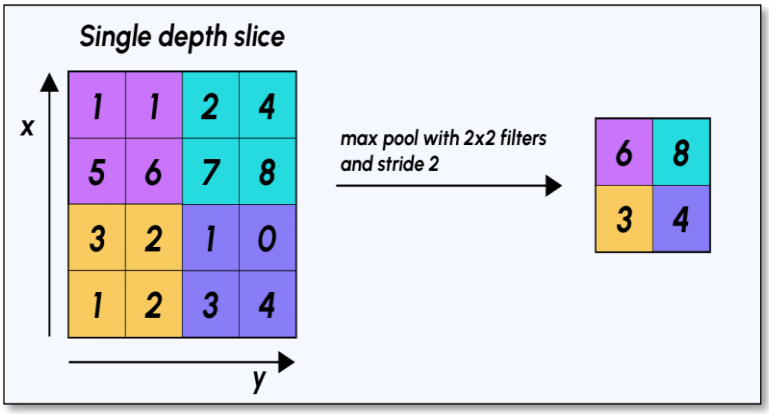


Figure 11 : Processus de Max-Pooling

La fenêtre de filtre se déplace de deux pixels vers la droite (stride/pas = 2) et récupère à chaque pas “l’argmax” correspondant à la valeur la plus grande parmi les 4 valeurs de pixels.

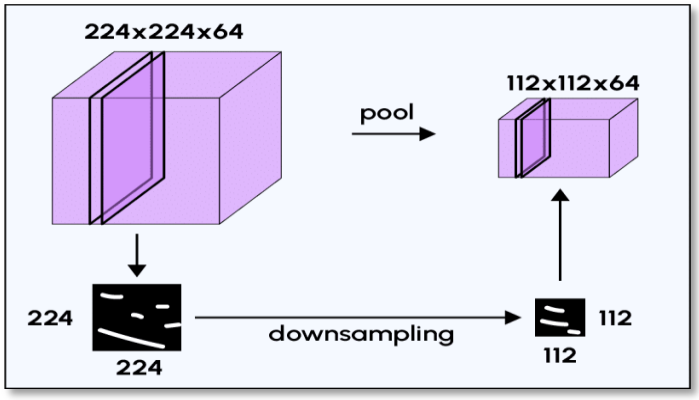


Figure 12 : Exemple d’effet du Max-Pooling

I.9 Réseau résiduel neuronal

Un réseau de neurones artificiels appelé RésNet est un réseau de neurones qui utilise des connexions de saut ou raccourcis pour traverser certaines couches. Dans ce ResNet, on met en place un saut double et triple de couches pour intégrer des non-linéarités sous forme de couches d'activation ReLU.

I.9. 1 Block résiduel

Un bloc résiduel est une accumulation de couches qui est définie de manière à ce que la sortie d'une couche soit intégrée et ajoutée à une autre couche plus profonde dans le bloc.

Après l'avoir ajoutée avec la sortie de la couche correspondante dans le chemin principal, on applique la non-linéarité. On appelle cette connexion by-pass le raccourci ou le saut de connexion.

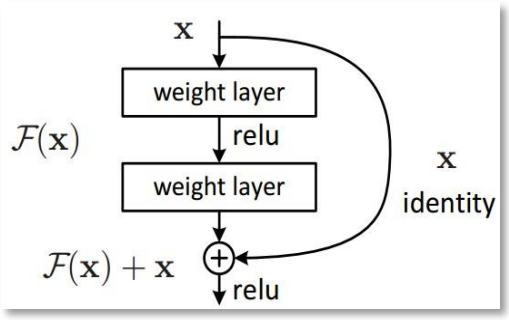


Figure 13 : Bloc résiduel de l'architecture ResNet

- **x (entrée)** : C'est l'entrée du bloc résiduel. Elle passe par deux chemins parallèles dans ce bloc.

- **Chemin principal (F(x)**:

Weightlayer : Il s'agit d'une couche de neurones (généralement une couche entièrement connectée ou une couche de convolution) qui applique des poids à l'entrée.

ReLU : Une fonction d'activation RectifiedLinear Unit (ReLU) est appliquée pour introduire la non-linéarité après la première couche de poids.

Le chemin principal peut être décrit comme F(x) , qui représente une transformation nonlinéaire de l'entrée x.

- **Chemin d'identité (x)** : Parallèlement au chemin principal, l'entrée x est acheminée directement vers la sortie sans aucune modification (identité).

- **Addition (F(x)+x)** : La sortie du chemin principal F(x) est ajoutée à l'entrée originale x. Cela permet de combiner les informations transformées par les couches de poids avec l'entrée originale.

- **ReLU final** : Après l'addition, une fonction d'activation ReLU est appliquée à la somme F(x)+x pour introduire une non-linéarité supplémentaire.

la sortie sera H (x) = F (x) + x

Ainsi, lorsque F (x) = 0, alors H (x) = x qui est connu sous le nom de mappage d'identité.

C’est-à-dire lorsque l'entrée du réseau est égale à sa sortie.

Afin d'ajouter F (x) + x, la forme des deux doit être exactement la même. Si la forme n'est pas la même, nous multiplions une matrice Ws avec l'entrée x.

I.9. 2 Connections de saut

Les connexions de saut sont une forme de raccourci qui permet de relier la sortie d'un calque à l'entrée d'un autre calque qui n'est pas adjacent. Prenons l'exemple d'un CNN comprenant quatre couches, A, B, C et D. Une connexion skip pourrait relier la couche A à la couche C, ou la couche B à la couche D, ou les deux. Il est possible de mettre en place des connexions d'omission de diverses façons, comme l'ajout, la concaténation ou la multiplication des sorties des couches ignorées. [17]

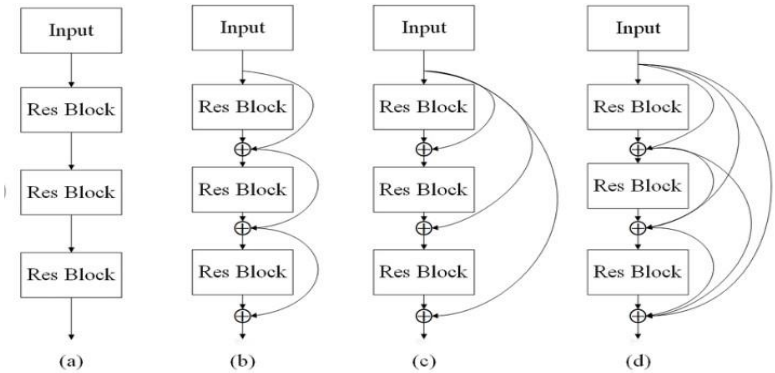


Figure 14 : Les différent schémas de connections de saut (a)

Pas de connexion sautée .

(b) Connexion de saut de source distincte.

(c) Connexion de saut de source partagée.

(d) Connexion de saut dense.

Les connections de saut fonctionne en facilitant la circulation des informations et des gradients sur le réseau. Les données d'entrée et les caractéristiques extraites par les couches sont appelées informations, tandis que les gradients sont les signaux qui modifient les poids des couches lors de la rétro propagation. Il est possible de sauter les connexions afin de préserver les informations et les gradients qui pourraient autrement être perdus ou dilués en traversant plusieurs couches. Ils ont aussi la capacité de rassembler des fonctionnalités de divers niveaux d'abstraction et de résolution, ce qui peut augmenter la capacité de représentation du réseau.

I.9. 3 Types de block résiduel

Il y a deux catégories de blocs résiduels :

• **Bloc d'Identité**

Un bloc d'identité est utilisé lorsque la dimension de l'entrée et de la sortie reste la même. Dans ce cas, l'entrée x est ajoutée directement à la sortie de la séquence de couches convolutives.

• **Bloc de Convolution**

Un bloc de convolution est utilisé lorsque la dimension de l'entrée et de la sortie diffère. Dans ce cas, une couche de convolution 1×1 est utilisée pour correspondre à la dimension de l'entrée à celle de la sortie avant d'ajouter l'entrée.

I.9. 4 Les fonctions d'activation

Les fonctions d'activation jouent un rôle crucial dans les réseaux de neurones et le deeplearning en introduisant de la non-linéarité dans le modèle, permettant ainsi au réseau de capturer des relations complexes dans les données. Voici quelques-unes des fonctions d'activation couramment utilisées :

• **Sigmoïde (logistique) :**

L'une des fonctions d'activation les plus couramment utilisées est la fonction sigmoïde (ou fonction logistique). La fonction est décrite comme suit :

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

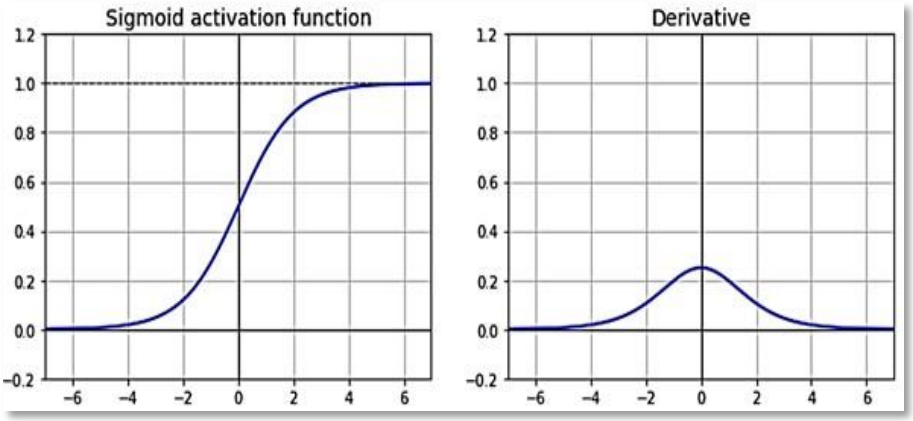


Figure 15 : le tracé de la fonction sigmoïde et de sa dérivée

• **La tangente hyperbolique :**

Il est défini comme:

$$\tanh = \frac{e^{x} - e^{-x}}{e^{-x} + e^{-x}}$$

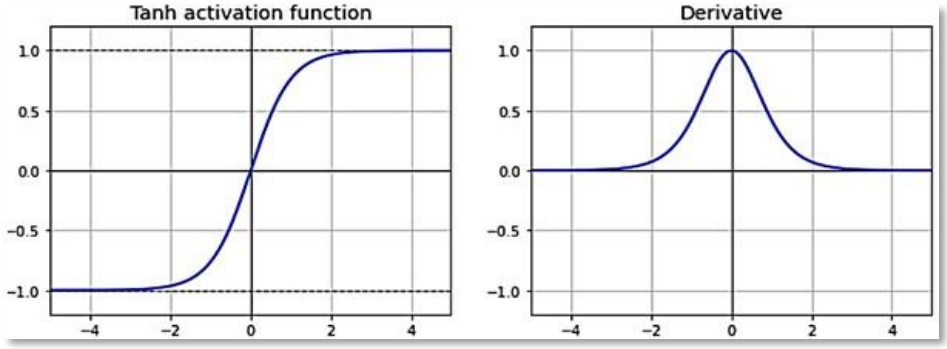


Figure 16 : le tracé de la fonction tanh et de sa dérivée

• L' unité linéaire rectifiée (ReLU) :

La fonction d'activation la plus fréquemment employée dans le domaine de l'apprentissage profond est l'unité linéaire rectifiée (ReLU). Si l'entrée est négative, la fonction renvoie 0, tandis que pour toute entrée positive, elle renvoie cette valeur. On définit la fonction comme suit :

$$\begin{cases} 0, & x < 0 \\ x, & x > 0 \end{cases}$$

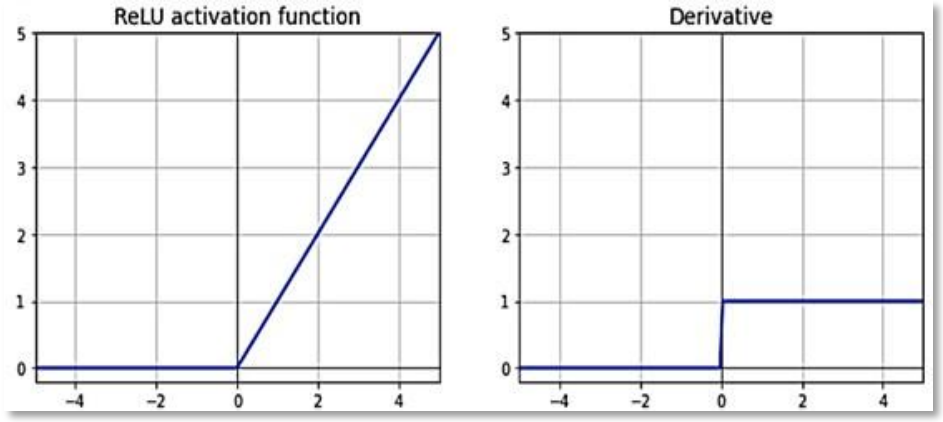


Figure 17 : le tracé de la fonction ReLU et de sa dérivée

I.9. 5 Le problème du gradient de disparition

Le gradient de disparition est une difficulté rencontrée lors de la création de réseaux de neurones artificiels, notamment des réseaux de neurones à réaction profonde et récurrente. Ce souci se produit pendant le processus de rétropropagation, qui est employé pour actualiser les poids du réseau neuronal en utilisant une descente de gradient. La règle de chaîne permet de calculer les gradients et de les propager à travers le réseau, en partant de la couche de sortie et en se dirigeant vers la couche d'entrée. Toutefois, lorsque les gradients sont très faibles, ils peuvent s'affaiblir au fur et à mesure qu'ils se propagent à travers le réseau, ce qui entraîne des mises à jour minimes, voire inexistantes, des pondérations dans les couches initiales. On nomme ce phénomène le problème du gradient de disparition [18].

On attribue souvent le problème du gradient de disparition à la sélection des fonctions d'activation et à l'organisation du réseau neuronal. Les gradients des fonctions d'activation telles que la tangente sigmoïde ou hyperbolique (tanh) varient de 0 à 0,25 pour la sigmoïde et de -1 à 1 pour la tanh. Les gradients de la fonction de perte par rapport aux paramètres peuvent devenir très faibles lorsque ces fonctions d'activation sont utilisées dans des réseaux profonds, ce qui empêche les poids de modifier leurs valeurs pendant l'entraînement.

L'initialisation des poids peut également être une autre cause du problème du gradient de disparition. L'initialisation de poids trop faibles peut entraîner une diminution exponentielle des gradients à mesure qu'ils se propagent à travers le réseau, ce qui entraîne la disparition des gradients [18].

I.9. 6 Quels problèmes les ResNets résolvent-ils?

Le problème du gradient qui disparaît est l'un des problèmes résolus par les ResNets. C'est le cas lorsque le réseau est trop profond, car les gradients, à partir desquels est calculée la fonction de perte, ont tendance à baisser rapidement jusqu'à zéro après plusieurs applications de la règle de la chaîne. Il en découle que les poids ne sont jamais actualisés et, par conséquent, aucun apprentissage n'est réalisé.

Grâce aux ResNets, les gradients ont la possibilité de se déplacer directement à travers les connexions de saut en arrière, des couches au-delà des filtres initiaux.

I.10 Conclusion

Dans ce chapitre, nous avons exploré les fondements et les avancées du deeplearning, en mettant un accent particulier sur les réseaux de neurones, les réseaux de neurones convolutifs (CNN), et les architectures résiduelles (ResNet). Nous avons discuté de la manière dont ces technologies ont transformé le paysage de l'intelligence artificielle et de l'apprentissage automatique, permettant de traiter et d'analyser des données complexes avec une précision et une efficacité sans précédent.

Les réseaux de neurones et les CNN se sont révélés extrêmement puissants pour apprendre des modèles complexes à partir de données brutes. Cette capacité a ouvert la voie à des solutions pour des problèmes considérés autrefois comme insolubles, et a permis des avancées significatives dans divers domaines, notamment la reconnaissance d'images, la détection d'objets et la traduction automatique.

Cependant, l'entraînement de réseaux de neurones très profonds a posé des défis importants, notamment le problème du gradient vanishing, qui rendait difficile l'ajustement des poids des couches profondes. L'introduction des architectures résiduelles, ou ResNet, a apporté une solution innovante à ce problème. Les connexions résiduelles (skip connections) de ResNet facilitent la propagation des gradients à travers le réseau, permettant ainsi d'entraîner des modèles très profonds sans dégradation des performances.

En conclusion, ce chapitre a mis en lumière l'impact significatif du deeplearning et des architectures avancées comme ResNet sur notre capacité à analyser, comprendre et agir sur des données complexes. Les avancées théoriques et pratiques dans ce domaine continuent d'inspirer de nouvelles innovations technologiques, transformant divers aspects de notre société et améliorant notre interaction avec la technologie.

ChapitreII: Introduction à la reconnaissance vocale

II.1 Introduction

La reconnaissance vocale, connue sous le nom de la reconnaissance automatique du locuteur (RAL), souvent désignée sous le terme de biométrie du locuteur, représente une branche cruciale de la biométrie vocale. Elle englobe divers processus allant de l'identification à la vérification, en passant par la classification, la segmentation, le suivi et la détection des locuteurs.

Alors que l'accent est souvent mis sur l'identification de l'utilisateur pour améliorer l'expérience utilisateur, la sécurité est généralement associée à la vérification du locuteur dans le domaine de la biométrie vocale. Cet aspect de la technologie vocale a gagné en maturité et en fiabilité, rendant les méthodes d'authentification basées sur la reconnaissance du locuteur de plus en plus prisées par les entreprises pour renforcer leur sécurité et améliorer leurs flux de travail.

Dans ce chapitre, nous avons abordé la définition de la parole, les tâches de reconnaissance automatique du locuteur (RAL) et les techniques utilisées pour améliorer cette reconnaissance.

Pour traiter efficacement les signaux vocaux, notamment en présence de bruit de fond, diverses techniques d'extraction des caractéristiques ont été développéesparmi les méthodes couramment utilisées pour l'extraction des caractéristiques on cite la transformée de Fourier (TF).

II.2 La parole

La parole est le principal moyen de communication dans toute société humaine. Son apparition peut être considérée comme concomitante à l'apparition des outils, l'homme ayant alors besoin de raisonner et de communiquer pour les façonner. Son abstraction par rapport à un support physique en fait un moyen de communication très simple à utiliser.

L'ère industrielle a par ailleurs permis de mettre en place des moyens d'enregistrement, et donc de sauvegarde, qui permettent à la parole de se hisser au rang de l'écrit pour la conservation de la connaissance. L'importance de la parole fait que toute interaction homme-machine devrait plus ou moins passer par elle. D'un point de vue humain, la parole permet de se dégager de toute obligation de contact physique avec la machine, libérant ainsi l'utilisateur qui peut alors effectuer d'autres tâches. Sans pour autant imposer la parole là où elle pourrait être un frein à l'interaction (il est par exemple difficile d'imaginer une application graphique où seule la parole serait utilisée), son utilisation permettrait de commencer à limiter l'emploi des claviers, tablettes graphiques et autres écrans tactiles ou gants de désignation.

Bien que plusieurs systèmes de reconnaissance de la parole soient aujourd'hui commercialisés par des sociétés plus ou moins spécialisées dans ce domaine, diverses études d'introduction d'interface vocale dans des applications existantes ont montré que les techniques actuellement mises en œuvre imposaient encore trop de contraintes.

La recherche en reconnaissance automatique de la parole, RAP, tente donc aujourd'hui de mieux comprendre le processus humain de génération et de compréhension de la parole, tant d'un point de vue mécanique par le biais de l'étude et de la modélisation des organes biologiques en charge de ces tâches, que d'un point de vue mathématique par le développement de méthodes de classification toujours plus fines et exactes [19].

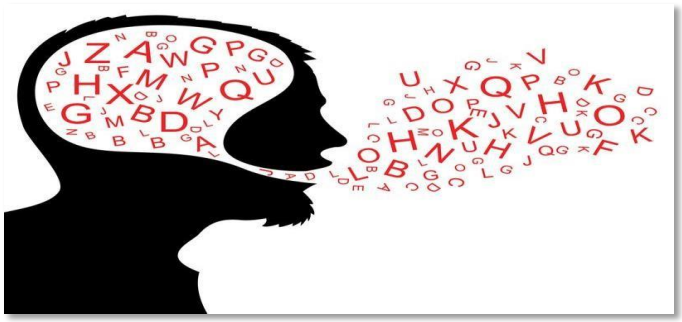


Figure 18 : La parole

II.2.1 Reconnaissance de la parole

La reconnaissance automatique de la parole revient à faire interpréter par unemachine un mot ou une phrase prononcés par un locuteur, en vue d'une action :
saisie de données, commande, etc. Au début des années 1970, les méthodes decomparaison « élastique » de Formes fondées sur des techniques de programmationdynamique ont été largement utilisées.

Ces méthodes sont désormais abandonnéesau profit d'une approche probabiliste du problème faisant appel à des modèlesstochastiques. Les performances des systèmes de reconnaissance actuels sontbonnes dans des conditions d'utilisation bien contrôlées. En revanche, le taux d'erreur s'accroît beaucoup lorsque les conditions d'apprentissage et d'utilisationd'un système sont différentes.

Les sources de variabilité de la parole sont diverses : l'environnement du locuteur, lelocuteur luimême et les conditions d'enregistrement et de transmission. Denombreuses techniques ont été proposées pour augmenter la robustesse dessystèmes, mais le problème demeure.

II.2.2 Système de reconnaissance de parole

La reconnaissance automatique de la parole pose de nombreux problèmes d'un pointde vue théorique. Leurs complexités faites que seuls des sous-problèmes ont pu êtreà ce jour résolus. Ces solutions partielles correspondent à des contraintes plus oumoins fortes, et les systèmes existants supposent une coopération plus ou moins grande des utilisateurs.

Pour classer les systèmes de reconnaissance automatique, on a généralementrecours aux critères suivants :

- Le mode d'élocution : des syllabes ou mots isolés aux mots connectés, jusqu'à une parole dite, « continue » c'est-à-dire sans pauses artificielles.
- Taille du vocabulaire et difficulté de la grammaire (la complexité du langageautorisé).
- La dépendance plus ou moins grande vis-à-vis du locuteur.

L'environnement protégé ou non (la robustesse aux conditions d'enregistrement).

En outre, il n'est pas inintéressant de les différencier selon deux points qui ont aussi leur importance :

- La compréhension est-elle requise ou non ? (Un système decompréhension cherche à accéder à la signification de l'énoncé parlé).
- Le discours est-il naturel, ou la syntaxe des phrases doit-elle être contraignante ?

Les systèmes réalisés de la RAP, sont conçus pour des applications spécifiques .Cela conduit à une restriction de l'univers du dialogue homme-machine.

L'objectif essentiel des études sur la reconnaissance et la compréhension automatique de la parole est « de permettre, à terme, un dialogue le plus naturel possible entre l'homme et la machine, dans le cadre d'une application spécifique».[8]

II.2.3 Description accoustique de la parole

Du point de vue phonétique acoustique, les sons du langage humain sont constitués par des ondes en mouvement. Il s'agit d'un mouvement vibratoire régulier ou irrégulier par les articulateurs et les cordes vocales. Ces vibrations sont ensuite transmises par les milieux matériels (l'air, l'eau, le bois, le métal,…). Contrairement à la lumière, le son ne se propage pas dans le vide, il est à signaler que le son se propage dans l'air à une vitesse de 340 m/ s [9].

III.2.3.1 La mélodie

La mélodie de la voix résulte de la vibration des cordes vocales, et se traduit phonétiquement par l'évolution de la fréquence de vibration laryngienne, on utilise plutôt le terme fréquence fondamentale, qui correspond à une estimation de fréquence laryngienne réalisée à partir du signal de parole [10].

III.2.3.2 La fréquence fondamentale (F0)

La fréquence fondamentale ou F0 est également appelée pitch. Elle représente le nombre de vibrations par seconde des cordes vocales. La F0 n'est calculée que sur des parties voisées de la parole. La gamme de variation moyenne de la fréquence fondamentale dépend, essentiellement, de l'âge, de l'état et du sexe du locuteur. Elle peut varier de :

- 70 à 250 Hz chez l'homme.
- 150 à 400 Hz chez la femme.
- 200 à 600 Hz chez l'enfant.

III.2.3.3 L'intensité

Par l'intensité, nous désignons la qualité qui nous fait distinguer un son fort d'un son faible. L'intensité augmente avec l'amplitude des vibrations sonores. On utilise une unité de mesure relative, le décibel (dB), pour rendre compte de l'intensité d'un son.

III.2.3.4 La durée

La durée est le paramètre acoustique le plus délicat à évaluer. La difficulté de mesure réside dans sa grande variabilité qui est due au contrôle quasi impossible du système phonatoire. Chaque phonème se caractérise par ses propres durés intrinsèques et extrinsèques.

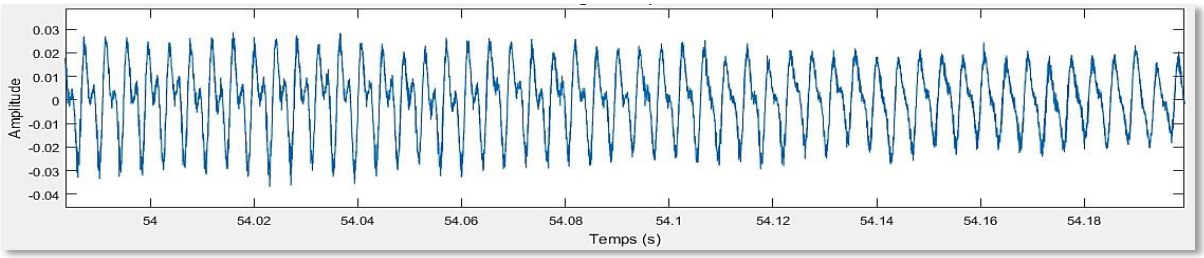


Figure 19 : Exemple d'un signal voisée

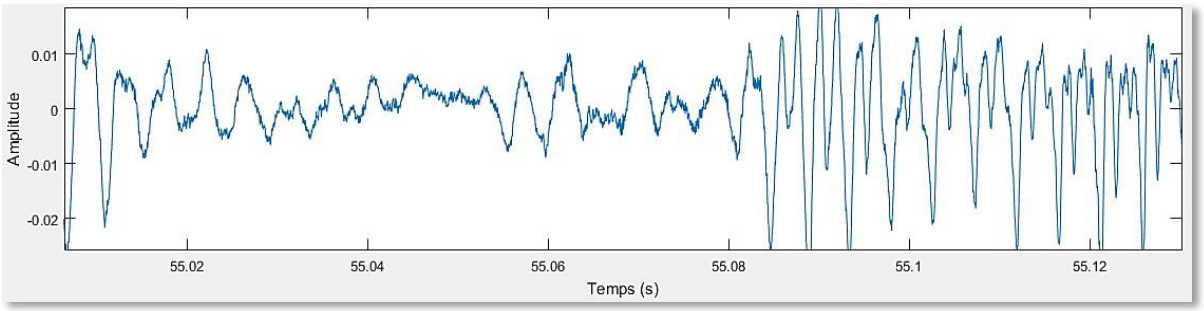


Figure 20 : Exemple d'un signal non voisée

III.2.3.5 Le timbre

Il représente la qualité particulière du son, indépendante de son intensité ou de sa hauteur, mais spécifique de l'instrument ou de la voix qui l'émet. Le timbre est la qualité qui nous permet de distinguer les différents instruments de musique ou de reconnaître une voix familière, par exemple.

III.2.3.6 Les formants

Les formants sont des zones fréquentielles de forte énergie, correspondent à une résonance dans le conduit vocal de la fréquence fondamentale produite par les cordes vocales. Ces formants représentent les maxima de la courbe de réponse enfréquences du conduit vocal.

II.3 Reconnaissance automatique du locuteur (RAL)

La caractérisation automatique du locuteur est un vaste domaine dans lequel la "machine" a pour tâche d'extraire du signal de parole les informations de nature à renseigner sur les spécificités d'un individu : identité, caractéristiques physiques, émotivité, état pathologique, particularités régionales, etc.

Elle s'applique à différents thèmes de recherche traitant des informations véhiculées par la voix tels que la classification d'individus, ou l'étude psychique ou physiologique d'une personne. La RAL est un sous-problème de la caractérisation automatique du locuteur. Son objectif est de reconnaître l'identité d'une personne à l'aide de sa voix. La variabilité de la parole entre locuteurs (variabilité inter-locuteur) est l'essence même de la RAL. Sans cette variabilité, il serait impossible de reconnaître une voix parmi plusieurs voix possibles.

La RAL, contrairement à la Reconnaissance Automatique de la Parole (RAP) s'intéresse tout particulièrement aux informations extra-linguistiques véhiculées par un signal vocal (signal de parole). Pourtant, la RAL a très souvent bénéficié des avancées de la RAP. Ainsi, de nombreuses techniques ont été appliquées en RAP avant d'être adaptées au domaine de la RAL.

Les applications de la RAL sont principalement liées aux problèmes d'authentification ou de confidentialité.

II.3.1 Différentes Tâches en RAL

La discipline de la reconnaissance automatique du locuteur comporte de nombreuses branches qui sont directement ou indirectement liées. En général, elle se manifeste de six manières différentes. Nous classifions ces branches en deux groupes différents, simple et composé. Les branches simples sont celles qui sont autonomes. D'autre part, les branches composées sont celles qui utilisent une ou plusieurs des branches simples, éventuellement avec des techniques supplémentaires.
Les branches simples de la RAL sont la vérification, l'identification, et la classification du locuteur. Selon la définition ci-dessus, les branches composées de la RAL sont la segmentation, la détection, et le suivi du locuteur. Actuellement, la vérification du locuteur est la branche la plus populaire en raison de la forte demande d'applications de sécurité et d'accès vocal.

II.3.1.1 Vérification du locuteur

La Vérification Automatique du Locuteur (VAL) est le processus décisionnel permettant de déterminer, au moyen d'un message vocal, la véracité de l'identité revendiquée par un individu . L'identité ainsi que le message vocal constituent les deux entrées du système de VAL. L'identité, nécessairement connue du système, désigne automatiquement la référence caractéristique d'un locuteur. Une mesure de similarité est calculée entre cette référence et le message vocal puis comparée à un seuil de décision. Dans le cas où la mesure de similarité est inférieure au seuil, l'individu est accepté, dans le cas contraire, l'individu est considéré comme un imposteur, est rejeté [11].

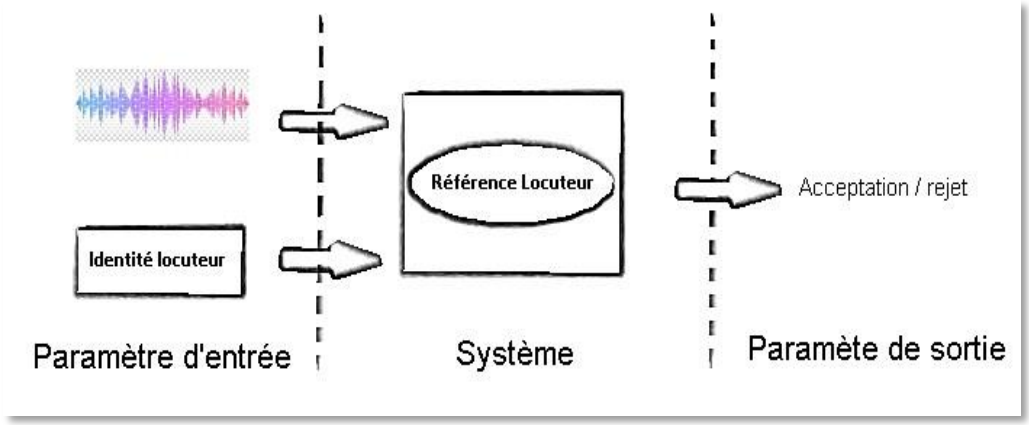


Figure 21 : Principe de base de la tâche de Vérification Automatique du Locuteur

- **Le fonctionnement :** La vérification du locuteur peut être dépendante ou indépendante du texte. La vérification en fonction du texte signifie que les locuteurs doivent choisir la même phrase de passe à utiliser pendant les phases d'inscription et de vérification. La vérification

indépendante du texte signifie que les locuteurs peuvent s'exprimer en langage courant dans les phrases d'inscription et de vérification.

Pour la vérification en fonction du texte, la voix du locuteur est enregistrée en prononçant une phrase de passe parmi un ensemble de phrases prédéfinies. Les caractéristiques de la voix sont extraites de l'enregistrement audio pour former une signature vocale unique, et la phrase de passe choisie est également reconnue. Ensemble, la signature vocale et la phrase de passe sont utilisées pour vérifier le locuteur.

La vérification indépendante du texte n'a aucune restriction sur ce que le locuteur dit pendant l'inscription, à part la phrase d'activation initiale pour activer l'inscription. Elle n'a aucune restriction sur l'échantillon audio à vérifier, car elle extrait uniquement les caractéristiques de la voix pour évaluer la similarité.

Les API ne sont pas destinées à déterminer si l'audio provient d'une personne vivante ou d'une imitation ou d'un enregistrement d'un locuteur inscrit.

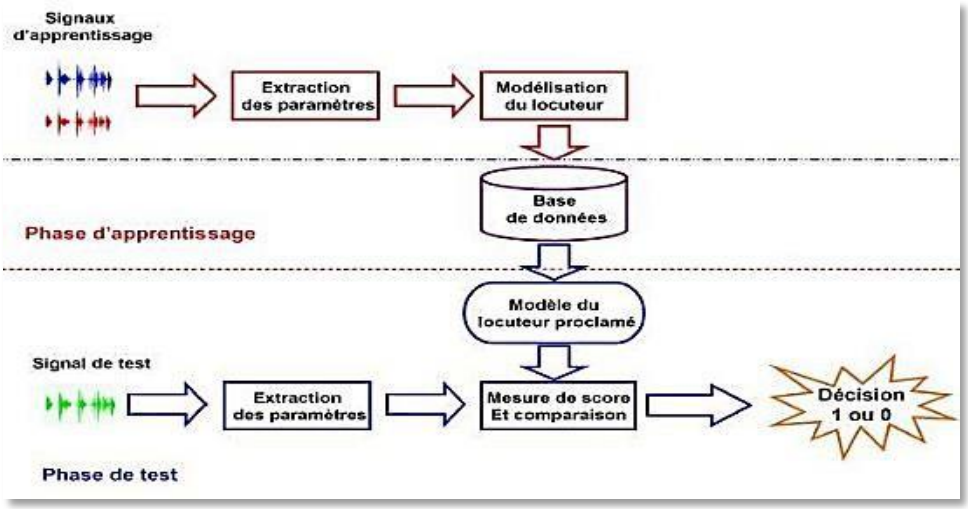


Figure 22 : Système de vérification du locuteur

II.3.1.2 Modules d'un système de vérification du locuteur

Les systèmes de vérification du locuteur sont composés de quatre modules principaux interdépendants.

1. Le prétraitement

Il s'agit d'extraire du signal de parole « pur » (valeurs de l'amplitude échantillonnées à des fréquences de l'ordre de 8 KHz) des caractéristiques(paramètres numériques et/ou symboliques). Un bon prétraitement fournit des paramètres dépendants des variations interlocuteurs et peu sensibles aux variations extrinsèques à l'identité du locuteur (conditions d'enregistrement ,variabilités intra locuteurs, etc.).

2. L'apprentissage

Il s'agit d'instancier des modèles à partir de paramètres extraits de locuteurs étiquettes ou non.

L'apprentissage se fait souvent par des méthodes d'entraînement itératives.

3. L’attribution de scores

Ce module est étroitement lié à la façon dont ont été conçus et entraînes les modèles dans le module d’apprentissage. Alors que ce dernier s’applique à des séquences d’entraînement, l’attribution de scores s’applique à des séquences test. Notons que l’apprentissage peut tenir compte de plusieurs séquences train pour l´élaboration d’un modèle, alors que le module de scoring traite les séquences test indépendamment les unes des autres.

4. La prise de décision

Ce petit module vient directement après l’attribution de scores. Typiquement, il s’agit de comparer les scores à un seuil (fixé lors de la phase de développement) pour renvoyer une décision binaire. Un seuil pour la prise de décision joue un rôle très important dans la performance d’un système vérification du locuteur .

II.3.1.3 Identification

L’Identification Automatique du Locuteur (IAL) est le processus qui consiste à déterminer, parmi une population de locuteurs connus, la personne ayant prononcé un message donné. D'un point de vue schématique (une séquence de parole est donnée en entrée du système d'IAL. Pour chaque locuteur connu du système, la séquence de parole est comparée à une référence caractéristique du locuteur : identité du locuteur dont la référence est la plus proche de la séquence de parole est donnée en sortie du système d'IAL.

Deux modes sont proposés en IAL, l’identification en ensemble :

- fermé pour lequel on suppose que la séquence de parole est effectivement prononcée par un locuteur connu du système .
- ouvert pour lequel le locuteur peut ne pas être connu.

En mode "ensemble ouvert", le système d'IAL doit décider de la fiabilité de son jugement en acceptant ou rejetant l’identité qu'il a trouvée. De par son principe - déterminer une identité parmi les identités potentielles - les performances des systèmes d'IAL se dégradent généralement au fur et à mesure que la population de locuteurs augmente [10].

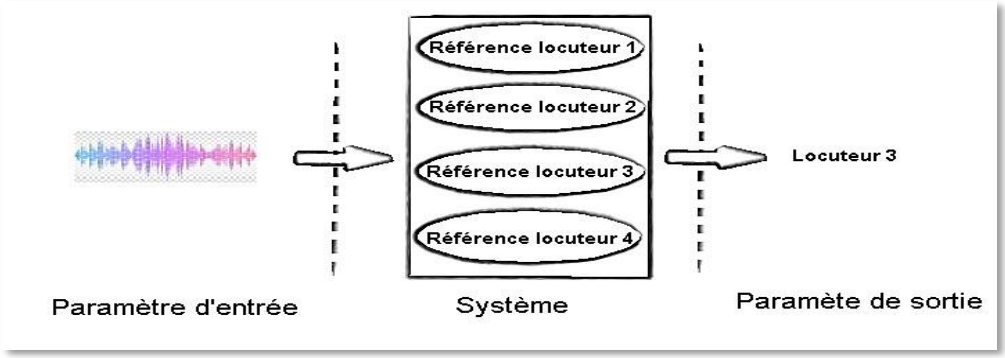


Figure 23 : Principe de base de la tâche d'Identification Automatique du Locuteur

- **Le fonctionnement :**L'inscription pour l'identification du locuteur est indépendante du texte. Il n'y a pas de restrictions sur ce que le locuteur dit dans l'audio, à part la phrase d'activation initiale pour activer l'inscription. Comme pour la vérification du locuteur, la voix du locuteur est enregistrée lors de la phase d'inscription, et les caractéristiques de la voix sont extraites pour former une signature vocale unique. Dans la phase d'identification, l'échantillon de voix

d'entrée est comparé à une liste spécifique de voix enregistrées (jusqu'à 50 dans chaque demande) [12].

II.3.2 Problèmes rencontrées en RAL

Les systèmes de RAL souffert des difficultés liées au domaine applicatif, comme l'utilisation des systèmes dans des conditions difficiles, les tentatives d'imposture.

III.3.2.1 Variabilité due au locuteur

Le signal parole varie pour un même individu parce que la voix d'une personne peut évoluer entre le début et la fin de la journée. Cette variabilité intra-locuteur est induite par l'évolution naturelle ou volontaire de la voix d'une personne.

III.3.2.2 Variabilité due au matériel

Cette variabilité est due aux: microphone, combiné téléphonique, ligne de transmission (ex : lignes téléphoniques), convertisseurs. Ces informations apparaissent le plus souvent sous la forme de déformations/dégradations du signal de parole.

III.3.2.3 Robustesse en environnements et tentatives d'imposture

Les systèmes de RAL doivent être robuste face au bruit ambiant et les environnements des canaux digitaux (téléphone, réseaux mobile, internet...). Dans le chapitre suivant on évoquera les réseaux et ses dégradations pour un objectif de développer un system de RAL à travers les canaux de transmission. Un système de RAL peut faire l'objet d'attaques d'individus envahissant l'identité de quelqu'un d'autre. Ces attaques peuvent, par exemple, avoir pour dessein des transactions frauduleuses sur le compte bancaire d'un client ou l'accès à des données confidentielles. Un système de RAL doit par conséquent être robuste [14].

II.4 Caractéristique d’extraction

L'authentification du locuteur est une technologie qui permet d'identifier ou de vérifier l'identité d'une personne basée sur sa voix. Pour ce faire, il est crucial d'extraire des caractéristiques de la parole. Voici une des principales techniques d'extraction de caractéristiques utilisées dans ce domaine :

II.4.1 Transformée de Fourier

II.4.1.1 Définition

La transformée de Fourier (TF) est un outil mathématique qui permet de décomposer une fonction ou un signal en ses composantes de fréquences. Elle transforme un signal temporel en un signal fréquentiel, ce qui est très utile pour analyser les propriétés des signaux périodiques et non périodiques. En termes simples, elle permet de voir quelles fréquences sont présentes dans un signal et quelle est leur amplitude.

Le son présente une ondulation. Il s'agit d'une vibration qui s'étend dans le temps. Cependant, lorsqu'on écoute un instrument de musique, on ne perçoit pas une vibration (fonction du temps), mais une note, c'est-à-dire une fréquence. Le poids relatif de chaque fréquence dans le signal temporel a été pesé par notre oreille, qui a calculé la transformée de Fourier du signal initial.

Formellement, la transformée de Fourier d'une fonctionf(t) est donnée par :

$$f(\omega) = \int_{-\infty}^{+\infty} e^{-j2\pi\omega t} f(t)dt$$

Où ω est la fréquence angulaire et j est l’unité imaginaire (avec j2=−1).

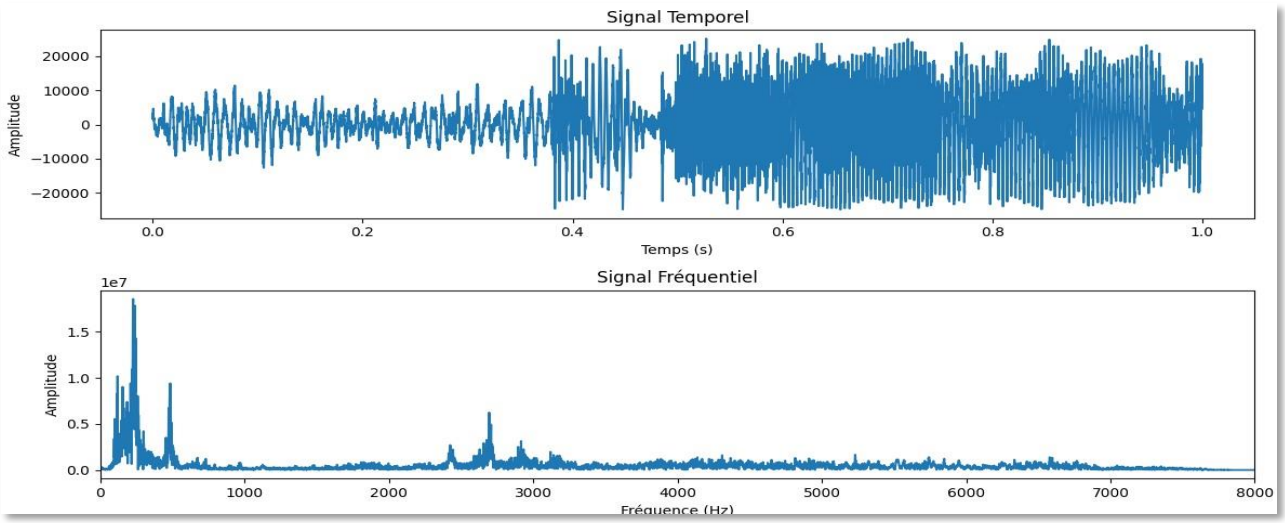


Figure 24 : signal audio en mode temporelle et fréquentielle

II.4.1.2 Transformée de fourier inverse

La transformée de Fourier inverse, également connue sous le nom d'IFT (inverse fouriertransform), est une méthode mathématique qui permet de transformer un signal de son domaine fréquentiel (spectre de fréquences) de retour dans le domaine temporel. Ce procédé revêt une importance capitale dans le domaine du traitement des signaux, car il permet de reconstruire le signal initial à partir de ses éléments fréquentiels.

La définition mathématique de la transformée de Fourier inverse est la suivante :

$$f(t) = \int_{-\infty}^{+\infty} e^{-j2\pi\omega t} f(\omega) dt$$

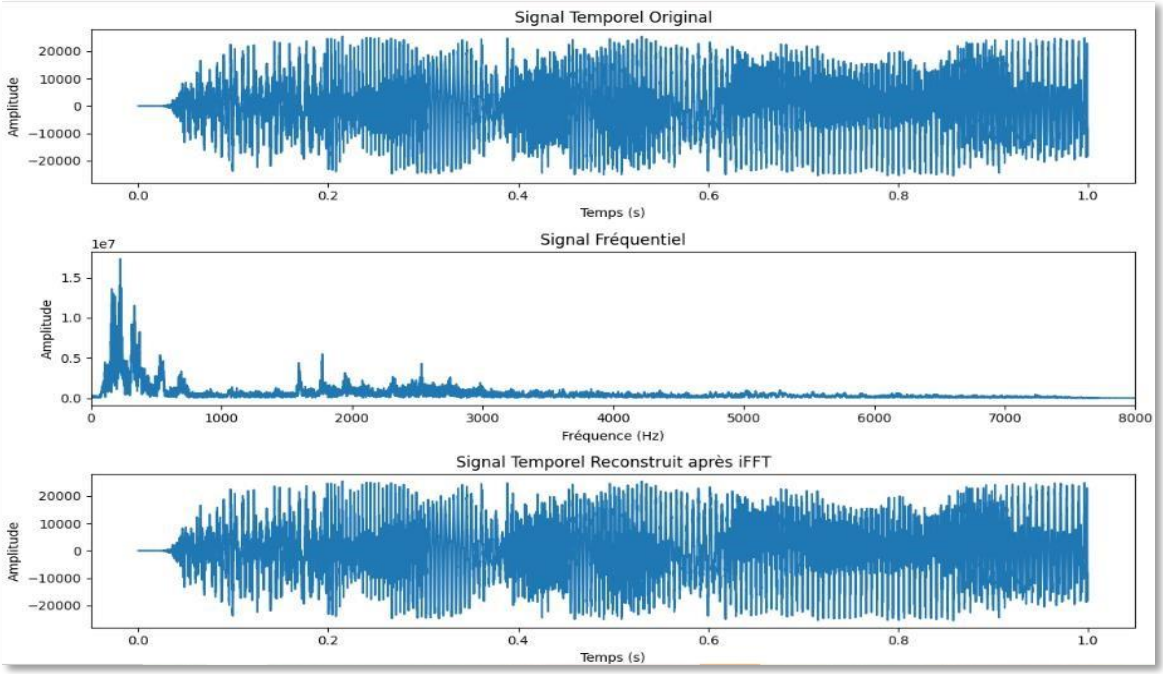


Figure 25 : inverser la TF et revenir au domaine temporel

II.4.1.3 L'importance de la TF

L'utilisation de la transformée de Fourier est cruciale dans de nombreux domaines, tels que les mathématiques, la physique, l'ingénierie et le traitement du signal. Elle offre la possibilité de diviser une fonction en une somme de fonctions sinus et cosinus de diverses fréquences, ce qui est pratique pour analyser et appréhender les signaux complexes. Elle est utilisée dans le domaine du traitement des images, de la compression de données, de la théorie de l'information, de la reconnaissance du locuteur , et bien d'autres domaines encore. Pour récapituler, la transformée de Fourier représente un outil puissant pour appréhender le fonctionnement des signaux et des systèmes dans différents secteurs.

On note aussi parmi ces avantages la rapidité et l'efficacité de ces résultats .

II.4.1.4 L’utilité de la TF dans le domaine de la reconnaissance du locuteur

La transformée de Fourier est une méthode utilisée dans le domaine de la reconnaissance du locuteur pour étudier les propriétés spectrales du signal vocal. Les composantes fréquentielles du signal vocal permettent d'obtenir des informations sur les caractéristiques propres à la voix d'un locuteur, comme le timbre, la fréquence fondamentale et les harmoniques. Par la suite, ces données sont exploitées afin de concevoir des modèles qui permettent d'identifier et de reconnaître un locuteur particulier. La transformée de Fourier joue donc un rôle crucial dans le traitement du signal vocal pour la reconnaissance du locuteur.

II.4.2 Le bruit

En général, un signal est perçu comme une grandeur physique qui contient des informations. Un bruit est donc une grandeur physique de la même nature que le signal, mais qui perturbe l'extraction ou l'exploitation de l'information. Les signaux sonores sont des signaux indésirables qui perturbent les signaux d'intérêt dans un système de communication ou de traitement de signal. Différentes sources peuvent être impliquées et peuvent avoir un impact sur la qualité et la précision des données transmises ou reçues.

II.4.2.1 Le bruit de fond (background noise)

Dans le domaine du traitement du signal, le bruit de fond désigne toute perturbation non souhaitée qui affecte la sortie d'un dispositif, indépendamment du signal à l'entrée. L'environnement sonore se divise en son propre, produit par le dispositif lui-même, et en perturbations provenant de l'extérieur qu'il capte par erreur. [18]

Dans le domaine de l'audio, de l'enregistrement et des systèmes de diffusion, le terme "bruit de fond" fait référence à toutes les perturbations indésirables qui impactent la sortie du système. Cela inclut les bruits ambiants du local de captation tels que les bruits de ventilation, les perturbations électromagnétiques causées par des défauts de compatibilité électromagnétique, les bruits d'origine vibratoire appelés bruits microphoniques, ainsi que le bruit propre généré par les appareils eux-mêmes, dont le bruit thermique est la composante principale, ainsi que le bruit de grenaille et le bruit de scintillation. [19]

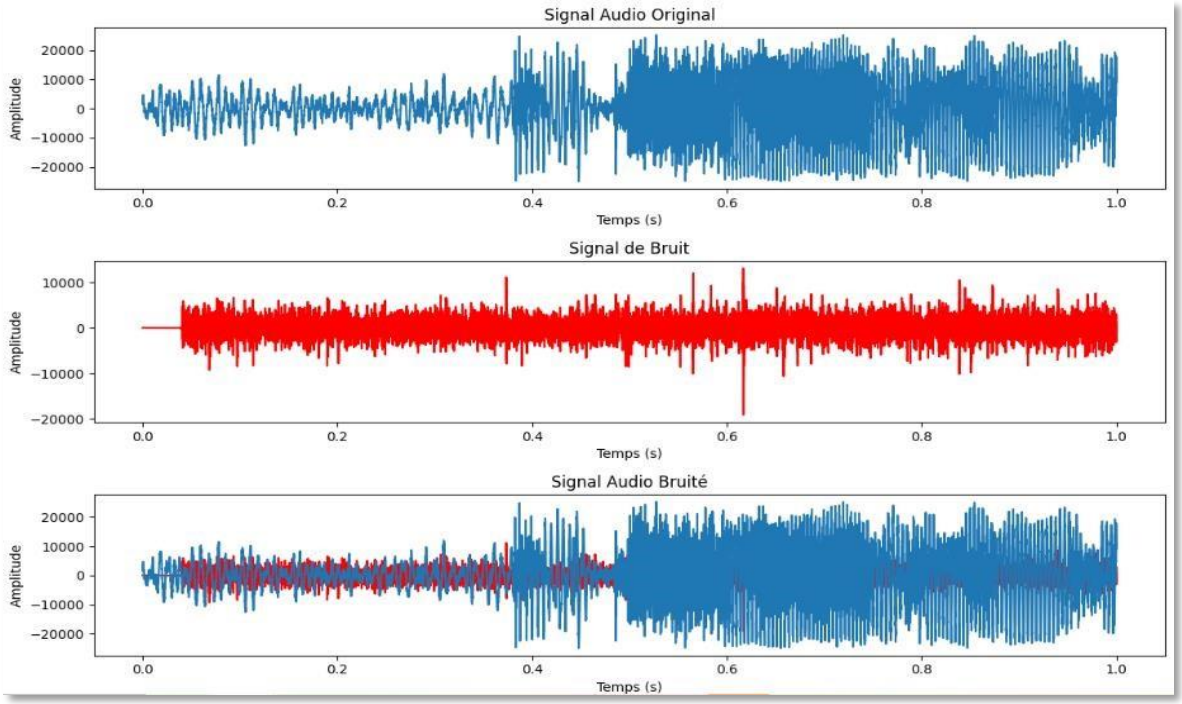


Figure 26 : signal audio avec un bruit de fond

II.4.2.2 Exemples des bruits de fond typiques

Imaginons une situation où un étudiant présente sa mémoire de fin d'études dans une salle de conférence de l'université. Voici quelques bruits de fond typiques que l'étudiant pourrait rencontrer :

- **Bruit de Ventilation** : Un bourdonnement constant provenant du système de ventilation ou de climatisation.
- **Bruits de Chaises** : Les chaises qui se déplacent lorsque le public s'assoit ou se lève.
- **Chuchotements** : Des membres du public qui chuchotent entre eux pendant la présentation.
- **Bruits de Portes** : Des portes qui s'ouvrent et se ferment lorsque des personnes entrent ou sortent de la salle.
- **Bruits de Fond Techniques** : Bruits provenant de l'équipement technique, comme le projecteur ou l'ordinateur.

II.5 Conclusion

Dans ce chapitre, nous avons exploré plusieurs aspects cruciaux de la technologie de reconnaissance de la parole. Nous avons commencé par distinguer la reconnaissance de la parole de la reconnaissance du locuteur : la première vise à comprendre ce qui est dit, tandis que la seconde cherche à identifier qui parle.

Ensuite, nous avons examiné l'importance de la Transformée de Fourier (TF) dans le traitement des signaux vocaux, en soulignant comment elle permet de convertir un signal temporel en ses composantes fréquentielles, facilitant ainsi l'analyse spectrale de la parole et l'extraction de caractéristiques pertinentes pour la reconnaissance vocale.

Enfin, nous avons abordé le problème du bruit de fond. En entraînant notre modèle avec des données contenant du bruit de fond, nous avons amélioré sa robustesse et sa capacité à fonctionner dans des environnements réels et bruyants. Cela montre l'importance d'intégrer des bruits de fond dans les données d'entraînement pour développer des systèmes de reconnaissance vocale fiables et efficaces.

ChapitreIII: Réalisations et expériences

III.1 Introduction

Dans ce chapitre, nous détaillons l'approche méthodologique et technique adoptée pour la réalisation de ce projet de reconnaissance vocale. L'utilisation de Python, avec ses vastes bibliothèques dédiées au traitement du signal et au deeplearning, a été au cœur de notre démarche. Nous explorerons comment des outils tels que NumPy, Librosa, Matplotlib, et TensorFlow ont été utilisés pour préparer les données, extraire des caractéristiques pertinentes, et construire des modèles de réseaux de neurones performants.

L'architecture de réseau de neurones convolutifs (CNN) que nous avons utilisée est basée sur le modèle ResNet (Residual Network), reconnu pour sa capacité à traiter efficacement des données complexes en profondeur grâce à ses connexions résiduelles. Ce modèle a été choisi pour sa robustesse et ses performances supérieures dans diverses tâches de classification.

En outre, nous décrirons la base de données utilisée, en détaillant sa composition, les types de données qu'elle contient, et les étapes de prétraitement effectuées pour assurer une qualité optimale des entrées du modèle. Nous aborderons également la division des données en ensembles d'entraînement, de validation et de test pour garantir une évaluation rigoureuse des performances du modèle.

Enfin, nous présenterons les résultats obtenus, en analysant les performances du modèle à travers diverses métriques et visualisations, et en discutant des défis rencontrés ainsi que des perspectives d'amélioration futures. Ce chapitre vise à fournir une compréhension complète de la mise en œuvre technique et des décisions méthodologiques prises pour atteindre les objectifs de ce projet.

III.2 La base de données

L'ensemble de données que nous avons utilisé est de source kaggle nommée « Speaker

Recognition Dataset » ,il comprend les discours de cinq dirigeants éminents : Benjamin Netanyahu, Jens Stoltenberg, Julia Gillard, Margaret Thatcher et Nelson Mandela. Les discours de Benjamin Netanyahu ont été supprimés de cette collection. Chaque discours est organisé en dossiers correspondant à chaque leader, et chaque fichier audio est standardisé par les caractéristiques suivantes : durée 1 seconde, encodé en PCM[1], et échantillonné à 16 000 Hz.

À l'origine, les discours de chaque leader existaient comme de longs fichiers audio. Pour améliorer l'utilisabilité et l'analyse, ces enregistrements ont été segmentés en intervalles d'une seconde, numérotés séquentiellement de 0.wav à 1500.wav. Cette segmentation facilite l'analyse

III.2.1 Les répertoires de données

```
[ ]  !cp -r "../input/speaker-recognition-dataset" ./
```

Copiez l'ensemble de données pour organiser l'audio et le bruit dans différents dossiers.

```
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)
```

- **Data directory :**Définition du répertoire principal du jeu de données
- **Audio_folder :** Définition du répertoire des fichiers audio
- **Noise_folder :** Définition du répertoire des fichiers de bruit
- **os.path.join() :**Construction du chemin vers le répertoire des fichiers audio / bruit

[1] **Pulse Code Modulation (PCM)** : est une technique de conversion des signaux analogiques en signaux numériques.

III.2.2 Définir les paramètres

```
valid_split = 0.1

shuffle_seed = 43

sample_rate = 16000

scale = 0.5

batch_size = 128
```

• **valid_split = 0.1 :** 10% des données seront utilisées pour la validation et 90% restants seront utilisées pour l'entrainement.

• **shuffle_seed = 43 :** Définir la graine de mélange pour les données avant de les diviser en ensemble d'entrainement et de validation.

• **sample_rate = 16000 :** Définition du taux d'échantillonnage , qui spécifie que les données audio seront échantillonnées à un taux de 16000 échantillons par seconde.

• **scale = 0.5 :** Définition du facteur d'échelle, pour réduire l'amplitude des données audio en les multipliant par 0.5.

• **batch_size = 128 :** Définition de la taille du lot (batch size), qui détermine que 128 exemples d'entraînement seront inclus dans chaque lot lors de l'entraînement du modèle.

III.2.3 Génération de base de données

```
for folder in os.listdir(data_directory):
    if os.path.isdir(os.path.join(data_directory, folder)):
        if folder in [audio_folder, noise_folder]:

            continue
        elif folder in ["other", "_background_noise_"]:

            shutil.move(
                os.path.join(data_directory, folder),
                os.path.join(noise_path, folder),
            )
        else:
            shutil.move(
                os.path.join(data_directory, folder),
                os.path.join(audio_path, folder),
            )
```

Ce code est conçu pour organiser les données en :

- Ignorant les dossiers déjà nommés pour les fichiers audio et de bruit.
- Déplaçant les dossiers spécifiquement nommés "other" et "background_noise" vers un répertoire de bruit.
- Déplaçant tous les autres dossiers vers un répertoire audio.

III.2.4 Divisez le bruit en morceaux de 16 000 pas chacun

```
command = (
    "for dir in `ls -1 " + noise_path + "`; do "
    "for file in `ls -1 " + noise_path + "/$dir/*.wav`; do "
    "sample_rate=`ffprobe -hide_banner -loglevel panic -show_streams "
    "$file | grep sample_rate | cut -f2 -d=`; "
    "if [ $sample_rate -ne 16000 ]; then "
    "ffmpeg -hide_banner -loglevel panic -y "
    "-i $file -ar 16000 temp.wav; "
    "mv temp.wav $file; "
    "fi; done; done"
)
```

 On utilise cette commande afin de garantir que tous les fichiers audio présents dans les sous-répertoires de `noise_path` sont échantillonnés à une fréquence de 16000 Hz. Elle les convertit au taux d'échantillonnage approprié en cas de non-conversion, remplaçant les fichiers originaux par les versions converties. Cela assure la cohérence des informations audio, ce qui revêt une importance capitale pour les opérations de reconnaissance vocale et de traitement audio.

III.2.5 Génération de la base de données

```
def paths_and_labels_to_dataset(audio_paths, labels):
    path_ds = tf.data.Dataset.from_tensor_slices(audio_paths)
    audio_ds = path_ds.map(lambda x: path_to_audio(x))
    label_ds = tf.data.Dataset.from_tensor_slices(labels)
    return tf.data.Dataset.zip((audio_ds, label_ds))


def path_to_audio(path):
    audio = tf.io.read_file(path)
    audio, _ = tf.audio.decode_wav(audio, 1, sample_rate)
    return audio
```

Paths_and_labels_to_dataset :

- **audio_paths**: une liste de chemins de fichiers audio.
- **labels :** une liste de labels correspondants à chaque fichier audio.

L'objectif de la fonction paths_and_labels_to_dataset est de transformer un ensemble de chemins de fichiers audio et de labels correspondants en un objet de datasetTensorFlow. Cet ensemble d'objets est employé pour entraîner ou évaluer des modèles d'apprentissage automatique, en particulier dans le domaine de la reconnaissance audio.

Path_to_audioest une fonction générale conçue pour lire un fichier audio à partir de son chemin spécifié et le décoder en tant que fichier WAV mono en utilisant TensorFlow. Cette fonction joue un rôle crucial dans la manipulation et le prétraitement des fichiers audio pour diverses tâches d'apprentissage automatique, comme la reconnaissance vocale, la classification audio, la génération de texte à partir de la parole, etc. Veillez simplement à indiquer la fréquence d'échantillonnage adéquate afin de décoder correctement le fichier audio.

III.2.6 Ajouter du bruit à la base de données

```
def add_noise(audio, noises=None, scale=0.5):
    if noises is not None:
        tf_rnd = tf.random.uniform(
            (tf.shape(audio)[0],), 0, noises.shape[0], dtype=tf.int32
        )
        noise = tf.gather(noises, tf_rnd, axis=0)

        prop = tf.math.reduce_max(audio, axis=1) / tf.math.reduce_max(noise, axis=1)
        prop = tf.repeat(tf.expand_dims(prop, axis=1), tf.shape(audio)[1], axis=1)

        audio = audio + noise * prop * scale

    return audio
```

 La fonction **add_noise** utilise un tensor audio ainsi qu'une liste de bruits (ou des bruits aléatoires générés) pour ajouter du bruit à l'audio d'entrée. Cette fonctionnalité permet d'intégrer du bruit dans des données audio, ce qui peut être avantageux dans certains scénarios d'entraînement de modèles afin de les rendre plus résistants aux conditions réelles où le bruit circule.

III.2.7 Divisé en formation et validation

```
# Split into training and validation
num_val_samples = int(valid_split * len(audio_paths))
train_audio_paths = audio_paths[:-num_val_samples]
train_labels = labels[:-num_val_samples]


valid_audio_paths = audio_paths[-num_val_samples:]
valid_labels = labels[-num_val_samples:]
```

Ces codes séparent un ensemble de données en ensembles d'entraînement et de validation en fonction d'un ratio précis (valid_split).

- **Calcul de la quantité d'échantillons de test :**Num_val_samples = entier(valid_split * nombre de chemins audio) En multipliant le ratio de validation (valid_split) par le nombre total d'échantillons dans le base de données, cette ligne permet de déterminer le nombre d'échantillons à utiliser pour l'ensemble de validation.

- **Création de l'ensemble de validation** : **Chemins audio = chemins audio[:-num_val_samples]** :En utilisant une découpe de liste, cette ligne choisit les chemins des fichiers audio pour l'ensemble d'entraînement. Son objectif est de prendre tous les éléments de la liste audio_paths, à l'exception de l'indice -num_val_samples. Cela implique que les derniers éléments num_val_samples sont supprimés de l'ensemble d'entraînement.

labels[:-num_val_samples] = train_labels : De la même manière, cette ligne choisit les labels correspondant aux chemins des fichiers audio pour l'ensemble de formation. Elle exclut les derniers labels num_val_samples.

- **Création de l'ensemble de validation** : **chemins audio valides = chemins audio[-num_val_samples:]** En utilisant les derniers éléments de la liste audio_paths, cette ligne choisit les chemins des fichiers audio pour l'ensemble de validation.

labels[-num_val_samples:] : valid_labels : De la même manière, cette ligne choisit les labels qui correspondent aux chemins des fichiers audio pour l'ensemble de validation en prenant en compte les derniers labels num_val_samples.

III.2.8 Créer des ensembles de données

```
# Create datasets, one for training and the other for validation
train_ds = paths_and_labels_to_dataset(train_audio_paths, train_labels)
train_ds = train_ds.shuffle(buffer_size=batch_size * 8, seed=shuffle_seed).batch(
    batch_size
)

valid_ds = paths_and_labels_to_dataset(valid_audio_paths, valid_labels)
valid_ds = valid_ds.shuffle(buffer_size=32 * 8, seed=shuffle_seed).batch(32)
```

 Ces lignes de code préparent les données audio pour l'entraînement et la validation en créant des ensembles de données tensorflow et en les configurant avec des opérations de mélange et de mise en lots. Cela permet de fournir des lots de données aléatoires à l'algorithme d'entraînement et de validation lors de l'entraînement du modèle.

III.3 Python

III.3.1 Introduction

Dans le cadre de ce projet de reconnaissance du locuteur , l'utilisation de Python, associée aux plateformes Kaggle et Google Colab, a joué un rôle crucial. Ces outils ont permis de tirer parti des bibliothèques avancées de traitement du signal et de deeplearning, tout en offrant un environnement de développement et de computation flexible et puissant. Cette section explore comment Python, Kaggle et Google Colab ont été intégrés pour construire et entraîner un modèle de reconnaissance vocale basé sur l'architecture ResNet.

III.3.2 Kaggle

Kaggle est une plateforme incontournable pour les data scientists et les chercheurs en Machine Learning. Elle offre un accès facile à une multitude de datasets de haute qualité, ainsi que des outils pour collaborer, partager du code et participer à des compétitions. Dans ce projet , Kaggle a été utilisé pour télécharger la base de données nécessaire à l'entraînement du modèle.

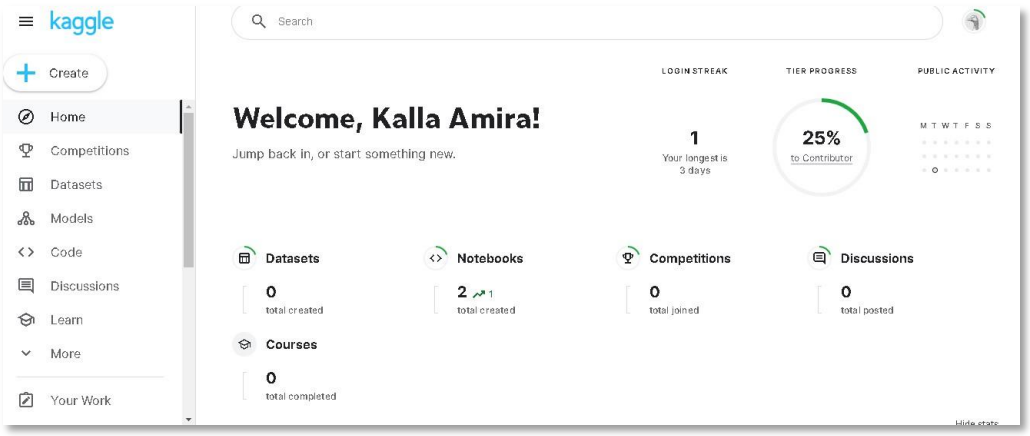


Figure 27 : Le menu de kaggle

Les avantages de Kaggle :

• **Accès à des Données de Qualité** : Kaggle propose une vaste collection de jeux de données publics couvrant divers domaines. Pour ce projet, Kaggle a été une source précieuse pour obtenir un ensemble de données de reconnaissance vocale.

• **Facilité d'Accès et de Téléchargement** : Grâce aux API et aux outils intégrés de Kaggle, le téléchargement et la gestion des datasets sont grandement simplifiés.

• **Communauté et Support** : Kaggle abrite une communauté active de data scientists et de développeurs qui partagent des notebooks, des solutions et des discussions, offrant un support précieux et des idées pour améliorer les projets.

III.3.3 Google Colab

Google Colab (Collaboratory) est un service de notebook basé sur le cloud qui permet aux utilisateurs de coder en Python directement depuis leur navigateur, en utilisant gratuitement des ressources de calcul avancées telles que des GPU et des TPU. Dans ce projet de reconnaissance vocale, Google Colab a été un outil essentiel pour le développement, l'entraînement et l'évaluation des modèles de deeplearning.

Voici une description détaillée de l'utilisation de Google Colab :

Figure 28 : le menu de googlecolab

Les avantages de googlecolab :

- **Accès Gratuit aux GPU et TPU** : Les tâches de deeplearning nécessitent souvent une puissance de calcul importante. Google Colab offre un accès gratuit aux GPU et TPU, permettant d'accélérer considérablement l'entraînement des modèles.
- **Partage et Collaboration** : Les notebooks Colab peuvent être facilement partagés et collaborés en temps réel, ce qui facilite le travail en équipe et la revue de code.
- **Environnement de Développement Flexible** : Colab intègre de nombreuses bibliothèques Python, simplifiant ainsi le processus d'installation et de gestion des dépendances.
- **Intégration avec Google Drive** : Colab permet de monter Google Drive, facilitant ainsi l'accès et le stockage des données et des modèles.

III.3.4 Intégration de google colabet kaggle

Google Colab et Kaggle, deux plateformes puissantes pour les data scientists et les chercheurs, offrent une synergie précieuse lorsqu'elles sont utilisées ensemble. Kaggle, avec sa vaste collection de jeux de données et ses compétitions, et Google Colab, avec ses capacités de calcul et son environnement de développement flexible, permettent de développer des projets de machine learning de manière fluide et efficace. Dans ce projet de reconnaissance vocale, l'intégration entre Kaggle et Google Colab a été exploitée pour télécharger les données et entraîner le modèle de deeplearning.

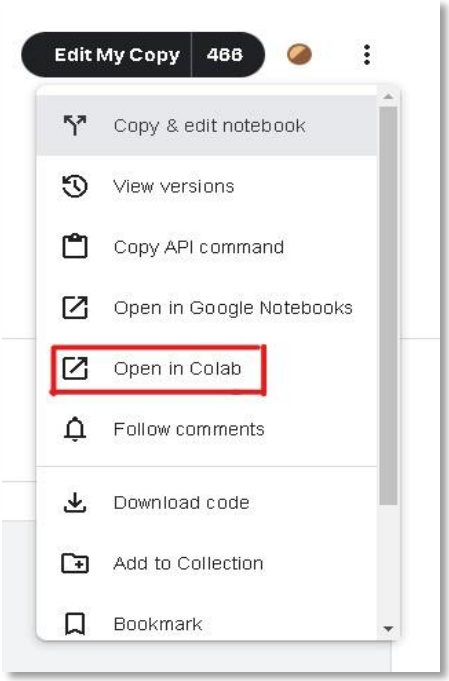


Figure 29 : Ouvrir des notebooks kaggle dans google colab

Kaggle propose une fonctionnalité pratique qui permet d'ouvrir directement des notebooks

dans Google Colab. Cette fonctionnalité simplifie grandement le processus de téléchargement et de manipulation des données, ainsi que le développement des modèles.

III.3.5 Les bibliothèques utilisées

III.3.5.1 Bibliothèques utilisées pour télécharger et gérer les données

```
import os
import sys
from tempfile import NamedTemporaryFile
from urllib.request import urlopen
from urllib.parse import unquote, urlparse
from urllib.error import HTTPError
from zipfile import ZipFile
import tarfile
import shutil
```

• **OS**

Description : La bibliothèque os fournit une manière simple d'utiliser les fonctionnalités dépendantes du système d'exploitation.

Utilisation : Création, suppression, et gestion des répertoires et fichiers.

• **SYS**

Description : La bibliothèque sys fournit un accès à des variables et des fonctions spécifiques au système d'exploitation.

Utilisation : Gestion des arguments de la ligne de commande et des chemins d'accès.

• **TEMPFILE**

Description : tempfile génère des fichiers et des répertoires temporaires.

Utilisation: Création de fichiers temporaires pour stocker les données téléchargées.

• ***URLLIB.REQUEST* et *URLLIB.PARSE***

Description : urllib.request permet d'ouvrir des URLs et urllib.parse fournit des fonctions pour manipuler les URLs.

Utilisation : Téléchargement des ensembles de données depuis Kaggle.

• **URLLIB.ERROR**

Description: Fournit des exceptions pour les erreurs rencontrées lors des requêtes HTTP.

Utilisation:Gestion des erreurs lors du téléchargement des données.

• **ZIPFILE**

Description : La bibliothèque zipfile permet de travailler avec des fichiers compressés au format ZIP.

Utilisation : Extraction des fichiers de données compressés.

• **TARFILE**

Description : La bibliothèque tarfile permet de lire et écrire des archives tar.

Utilisation : Extraction des fichiers de données compressés.

• **SHUTIL**

Description : La bibliothèque shutil offre des fonctions pour des opérations de fichiers de haut niveau.

Utilisation : Copie, déplacement et suppression de fichiers et répertoires.

III.3.5.2 Bibliothèques utilisées pour le traitement des signaux et l'entraînement du modèle

```
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
```

• **TENSORFLOW**

Description : TensorFlow est une bibliothèque open-source pour le calcul numérique et le machine learning.

Utilisation : Construction et entraînement du modèle ResNet.

• **KERAS**

Description: Keras est une API de haut niveau pour construire et entraîner des modèles de deeplearning, qui peut utiliser TensorFlow en backend.

Utilisation : Définition et compilation du modèle de deeplearning.

• **NUMPY**

Description: NumPy est une bibliothèque fondamentale pour le calcul scientifique en Python.

Utilisation : Manipulation des tableaux de données et calculs mathématiques.

• **PATHLIB**

Description : Pathlib offre une interface orientée objet pour travailler avec les chemins de fichiers.

Utilisation: Gestion et manipulation des chemins de fichiers.

• **IPYTHON**.DISPLAY

Description: Fournit des fonctions pour afficher des objets interactifs dans les notebooks.

Utilisation: Lecture et affichage des fichiers audio dans les notebooks.

• **SUBPROCESS**

Description : Permet de créer de nouveaux processus, de se connecter à leur entrée/sortie et d'obtenir leur retour d'exécution.

Utilisation: Exécution de commandes système pour gérer les fichiers et les téléchargements.

III.4 Caractéristique d'extraction

L'extraction des caractéristiques est une étape cruciale dans le traitement des signaux audio, particulièrement pour des tâches telles que la reconnaissance vocale. Dans ce projet, la Transformée de Fourier Rapide (FFT) a été utilisée pour extraire des caractéristiques pertinentes des signaux audio. La FFT permet de transformer un signal audio du domaine temporel au domaine fréquentiel, offrant ainsi une représentation plus utile pour l'analyse et la modélisation.

```
def audio_to_fft(audio):
    audio = tf.squeeze(audio, axis=-1)
    fft = tf.signal.fft(
        tf.cast(tf.complex(real=audio, imag=tf.zeros_like(audio)), tf.complex64)
    )
    fft = tf.expand_dims(fft, axis=-1)

    return tf.math.abs(fft[:, : (audio.shape[1] // 2), :])
```

```
# Add noise to the training set
train_ds = train_ds.map(
    lambda x, y: (add_noise(x, noises, scale=scale), y),
    num_parallel_calls=tf.data.experimental.AUTOTUNE,
)

# Transform audio wave to the frequency domain using `audio_to_fft`
train_ds = train_ds.map(
    lambda x, y: (audio_to_fft(x), y), num_parallel_calls=tf.data.experimental.AUTOTUNE
)

train_ds = train_ds.prefetch(tf.data.experimental.AUTOTUNE)

valid_ds = valid_ds.map(
    lambda x, y: (audio_to_fft(x), y), num_parallel_calls=tf.data.experimental.AUTOTUNE
)
valid_ds = valid_ds.prefetch(tf.data.experimental.AUTOTUNE)
```

• **Fonction add_noise** :

Cette fonction prend un exemple audio et ajoute du bruit à partir d'une collection de bruits préalablement définie. Le paramètre scale contrôle l'intensité du bruit ajouté.

• **Fonction audio_to_fft** :

Cette fonction transforme les données audio du domaine temporel au domaine fréquentiel en utilisant la transformée de Fourier (FFT). La fonction tf.signal.fft calcule la FFT du signal. Ensuite, elle prend la moitié des fréquences (jusqu'à la moitié de la longueur de l'audio), car la FFT d'un signal réel est symétrique.

• **Transformation des ensembles de données** :

Entraînement : Ajoute du bruit aux exemples audio et les transforme en domaine fréquentiel avant de les utiliser pour entraîner le modèle.

Validation : Transforme directement les exemples audio en domaine fréquentiel sans ajouter de bruit, pour évaluer le modèle sur des données non altérées.

• **Optimisation avec prefetch** :

Utiliser prefetch avec tf.data.experimental.AUTOTUNE permet de précharger les données en arrière-plan pendant que le modèle est en train de s'entraîner, ce qui peut augmenter la vitesse de l'entraînement.

III.5 Modèle

Ce modèle CNN utilise des blocs résiduels, une technique clé de l'architecture ResNet, pour améliorer la reconnaissance de motifs dans des données séquentielles comme des signaux audio. Les blocs résiduels facilitent l'entraînement en permettant une meilleure propagation des gradients, ce qui est particulièrement important dans les réseaux profonds. En combinant des couches convolutionnelles 1D et des couches entièrement connectées, le modèle est capable d'extraire des caractéristiques complexes des données d'entrée et de les classifier avec précision. Des techniques de régularisation et d'optimisation sont employées pour maximiser la performance tout en évitant le surapprentissage.

III.5.1 Blocks résiduels

Les blocs résiduels sont des sous-structures du réseau qui permettent d'apprendre des fonctions résiduelles par rapport aux entrées originales des blocs. Cette technique, introduite par les réseaux ResNet, aide à prévenir le problème de la dégradation dans les réseaux très profonds.

```
def residual_block(x, filters, conv_num = 3, activation = "relu"):
    s = keras.layers.Conv1D(filters, 1, padding = "same")(x)

    for i in range(conv_num - 1):
        x = keras.layers.Conv1D(filters, 3, padding = "same")(x)
        x = keras.layers.Activation(activation)(x)

    x = keras.layers.Conv1D(filters, 3, padding = "same")(x)
    x = keras.layers.Add()([x, s])
    x = keras.layers.Activation(activation)(x)

    return keras.layers.MaxPool1D(pool_size = 2, strides = 2)(x)
```

- **Shortcut (s)** : Une connexion de raccourci qui contourne les couches intermédiaires et est ajoutée à la sortie du bloc convolutif.

- **Convolutions** : Le nombre de convolutions est défini par conv_num. Chaque convolution est

suivie d'une fonction d'activation ReLU.

- **Addition** : La sortie des convolutions est ajoutée au raccourci.
- **Activation et MaxPooling** : La sortie résultante passe par une autre fonction d'activation ReLU et une couche de mise en commun.

III.5.2 Construction du modèle

```
def build_model(input_shape, num_classes):
    inputs = keras.layers.Input(shape = input_shape, name = "input")

    x = residual_block(inputs, 16, 2)
    x = residual_block(inputs, 32, 2)
    x = residual_block(inputs, 64, 3)
    x = residual_block(inputs, 128, 3)
    x = residual_block(inputs, 128, 3)
    x = keras.layers.AveragePooling1D(pool_size=3, strides=3)(x)
    x = keras.layers.Flatten()(x)
    x = keras.layers.Dense(256, activation="relu")(x)
    x = keras.layers.Dense(128, activation="relu")(x)

    outputs = keras.layers.Dense(num_classes, activation = "softmax", name = "output")(x)

    return keras.models.Model(inputs = inputs, outputs = outputs)
```

- **Entrée** : Définie par input_shape.
- **Blocs Résiduels** : Cinq blocs résiduels avec des filtres croissants.
- **AveragePooling1D** : Une couche de mise en commun moyenne pour réduire la dimensionnalité.
- **Flatten** : Aplatie les données avant de les passer aux couches denses.
- **Couches Denses** : Deux couches denses avec des activations ReLU.
- **Sortie** : Une couche dense avec une activation softmax pour la classification.

III.5.3 Compilation et Callbacks

```
model = build_model((sample_rate // 2, 1), len(class_names))

model.summary()

model.compile(optimizer="Adam", loss="sparse_categorical_crossentropy", metrics=["accuracy"])

model_save_filename = "model.h5"

earlystopping_cb = keras.callbacks.EarlyStopping(patience=10, restore_best_weights=True)

mdlcheckpoint_cb = keras.callbacks.ModelCheckpoint(model_save_filename, monitor="val_accuracy", save_best_only=True)
```

- **Optimizer** : Adam, un optimiseur populaire et efficace.

L'optimiseur Adam (Adaptive Moment Estimation) est un algorithme d'optimisation populaire utilisé pour entraîner des réseaux de neurones. Il combine les avantages de deux autres méthodes : le momentum, qui accélère la convergence, et AdaGrad, qui adapte le taux d'apprentissage pour chaque paramètre. Adam maintien des estimations adaptatives des moments du premier ordre (moyenne des gradients) et du second ordre (moyenne des carrés des gradients), rendant l'entraînement plus efficace et stable. Il est largement utilisé car il est adaptatif, efficace et facile à utiliser avec peu de réglages.

- **Loss** : sparse_categorical_crossentropy, adapté pour les problèmes de classification avec des labels entiers.
- **Callbacks** :

  EarlyStopping : Arrête l'entraînement si la performance sur l'ensemble de validation ne s'améliore pas pendant 10 epochs consécutives.

  ModelCheckpoint : Sauvegarde le meilleur modèle basé sur la précision de validation.

III.5.4 L'entrainement du modèle

Ce code entraîne notre modèle en utilisant l'ensemble de données d'entraînement, valide les performances à chaque epoch, et utilise les callbacks pour optimiser le processus d'entraînement.

```
epochs = 15
history = model.fit(
    train_ds,
    epochs=epochs,
    validation_data=valid_ds,
    callbacks=[earlystopping_cb, mdlcheckpoint_cb],
)
```

• **Hyperparamètres :**epochs = 15 : Le modèle passera 15 fois sur l'ensemble des données d'entraînement.

• **Entraînement :train_ds :** Ensemble de données d'entraînement. **valid_ds :** Ensemble de données de validation pour évaluer la performance du modèle pendant l'entraînement.

• **callbacks : earlystopping_cb :** Arrête l'entraînement si la performance sur l'ensemble de validation ne s'améliore pas pendant plusieurs epochs consécutifs, afin d'éviter le surapprentissage.

• **mdlcheckpoint_cb :** Sauvegarde le modèle avec la meilleure précision sur l'ensemble de validation.

III.5.5 Interprétation du résultat

Accuracy

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

19/19 [==============================] - 31s 2s/step - loss: 0.0218 - accuracy: 0.9900
Accuracy of model: [0.021825481206178665, 0.9900000095367432]
```

- **Evaluation** : model.evaluate(valid_ds) : Cette méthode évalue le modèle sur l'ensemble de données de validation valid_ds.

• **Résultats** :[0.021825481206178665, 0.9900000095367432]

0.0218 : C'est la valeur de la fonction de perte sur l'ensemble de validation.

0.9900 : C'est la précision (accuracy) du modèle sur l'ensemble de validation.

La précision de notre modèle est de 0.9900 .Cela signifie que notre modèle a correctement classé 99% des exemples de l'ensemble de validation .

• **Calcul de la Précision (Accuracy)**

La précision est définie comme le ratio des prédictions correctes par rapport au nombre total de prédictions. Voici la formule :

$$\text{Accuracy} = \frac{\text{nombres de précision correctes}}{\text{Nombre totale de précisions}}$$

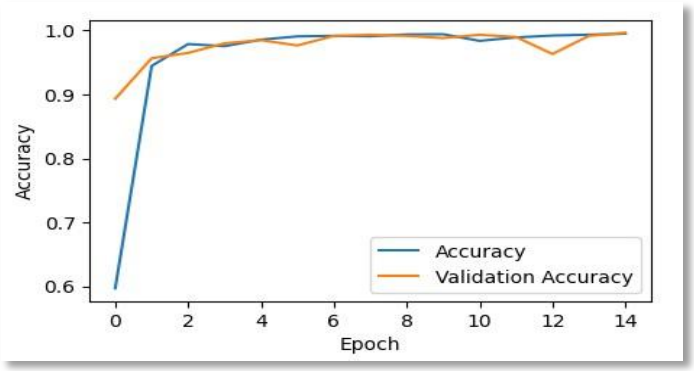


Figure 30 : Précision d'entrainement et de validation

En TensorFlow/Keras, cette métrique est calculée automatiquement en utilisant model.evaluate() ou model.fit().

• **Calcul de la Perte (Loss)**

La perte mesure combien les prédictions du modèle sont éloignées des vraies valeurs. Le choix de la fonction de perte dépend du type de problème (classification, régression, etc.). Pour une classification multi-classes, on utilise généralement l'entropie croisée (cross-entropy). Voici la formule de l'entropie croisée :

$$\text{Loss} = -\sum_{i} y_i \log(\widehat{y_i})$$

51

Où :

y_i est la véritable étiquette (0 ou 1).

$\hat{y}_i$ est la probabilité prédite par le modèle.

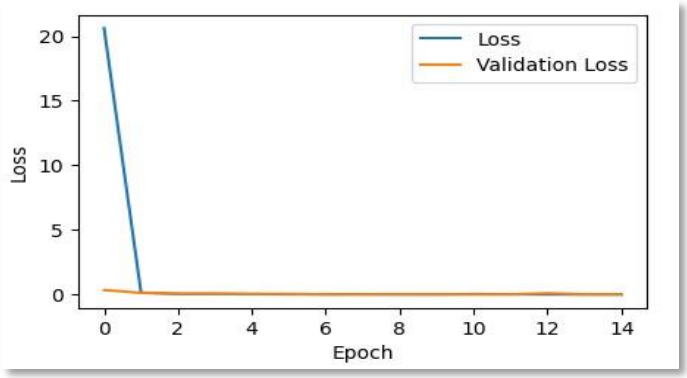


Figure 31 : Perte d'entrainement et de validation

III.5.6 Tester le modèle

```
[ ]  path = ["../input/speaker-recognition-dataset/16000_pcm_speeches/Jens_Stoltenberg/1013.wav"]
     labels = ["unknown"]
     try:
         predict(path, labels)
     except:
         print("Error! Check if the file correctly passed or not!")
```

```
1/1 [==============================] - 0s 212ms/step
Speaker: 3      Predicted: 3
Speaker Predicted: Jens_Stoltenberg
```

 Le modèle a correctement prédit que l'orateur est Jens Stoltenberg.

III.6 Organigramme

Le schéma suivant illustre l'organisation d'un modèle de reconnaissance vocale utilisant ResNet, qui vise à repérer des locuteurs particuliers à partir de leurs enregistrements vocaux. Pour extraire les caractéristiques des signaux audio, l'architecture utilise des couches de convolution et des blocs résiduels, puis des couches denses pour la classification finale. Ce modèle allie des méthodes de traitement du signal sophistiquées et des techniques de deep learning afin de garantir une précision et une solidité exceptionnelles dans la reconnaissance vocale.

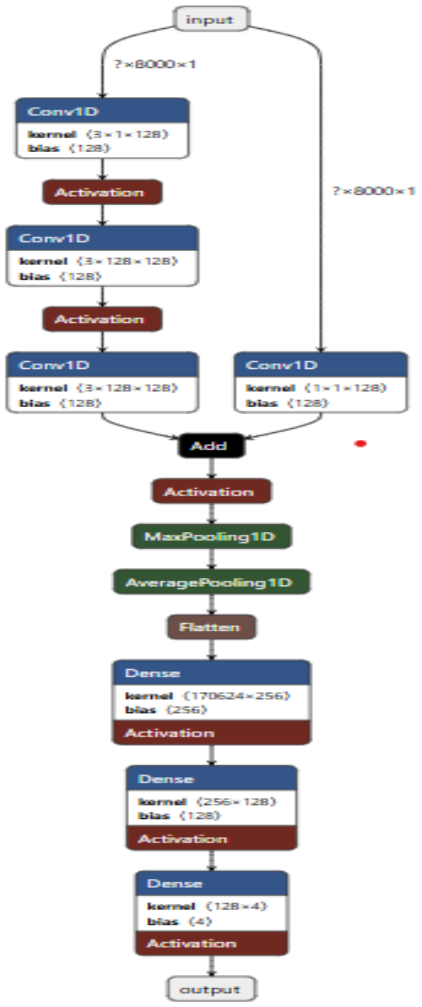


Figure 32 :organigramme d'un Modèle de Reconnaissance Vocale Basé sur ResNet

III.7 L’interface graphique

Pour faciliter l'interaction avec notre modèle de reconnaissance vocale, nous avons développé une interface graphique intuitive et conviviale. Cette interface permet aux utilisateurs de : télécharger des fichiers audio , tracer le spectrogramme de l’audio , identifier le locuteur et afficher sa photo .

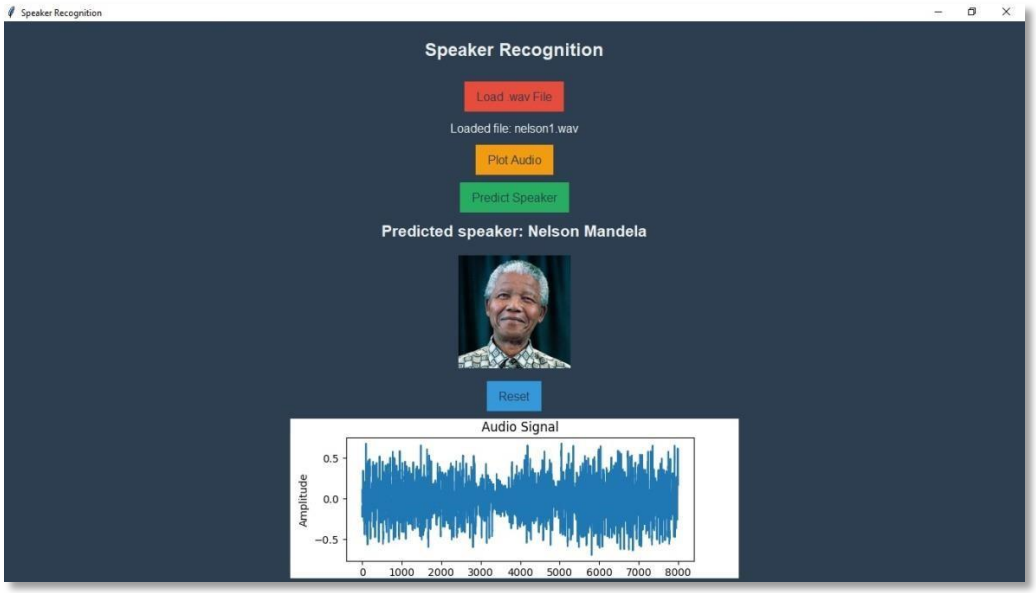


Figure 33 : Identifier le premier locuteur

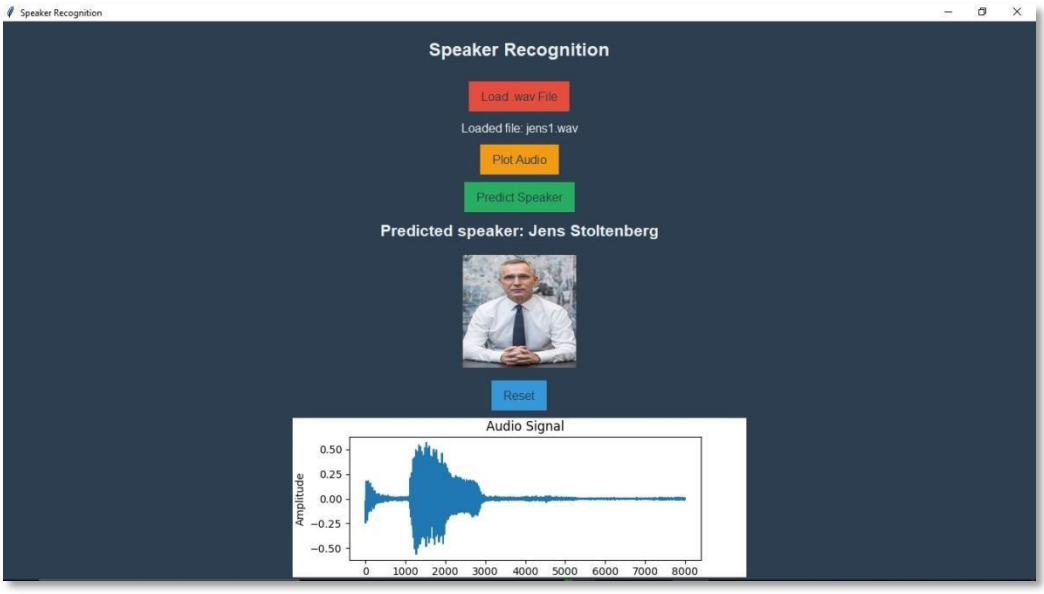


Figure 34:Identifier le deuxième locuteur

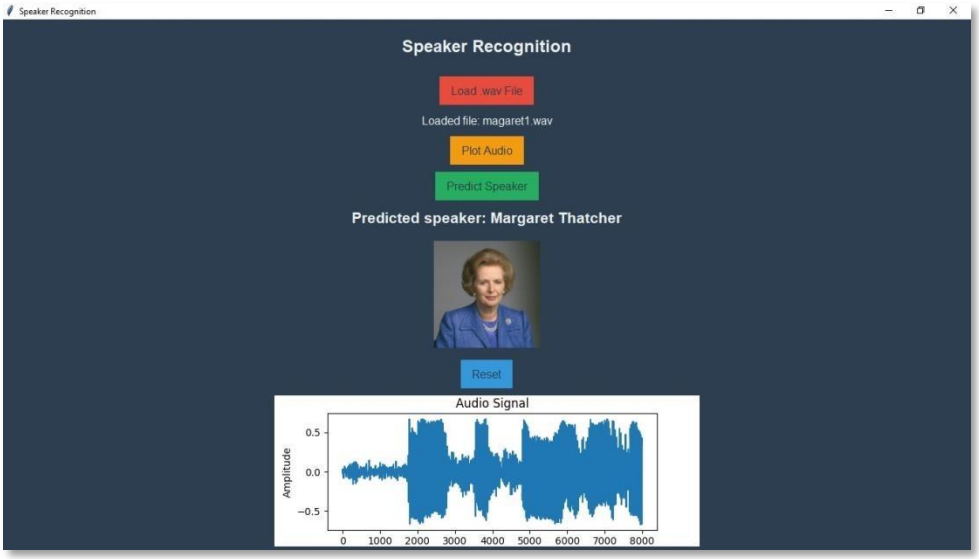


Figure 35 : Identifier le troisième locuteur

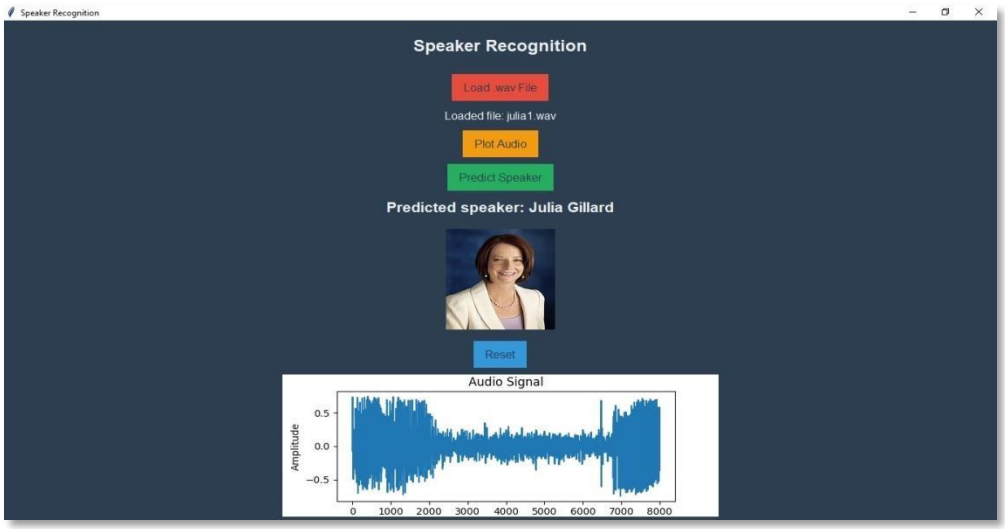


Figure 36 : Identifier le denier locuteur

III.8 Conclusion

Ce projet de reconnaissance vocale visait à développer un modèle capable d'identifier les discours de quatre éminents dirigeants : Jens Stoltenberg, Julia Gillard, Margaret Thatcher et Nelson Mandela. En utilisant un ensemble de données audio standardisé, chaque fichier audio a été converti en format PCM avec une fréquence d'échantillonnage de 16000 Hz, garantissant ainsi la qualité et la cohérence des données d'entraînement. Pour améliorer la robustesse du modèle face aux variations et aux interférences audio, nous avons ajouté du bruit aux données d'entraînement. Les ondes audio ont été transformées en domaine fréquentiel à l'aide de la transformée de Fourier (FFT), capturant ainsi les caractéristiques essentielles des signaux audio.

Le modèle reposait sur l'architecture ResNet, utilisant des blocs résiduels pour une meilleure performance. Les connexions de saut (shortcuts) dans ces blocs maintiennent des gradients stables pendant l'entraînement, évitant les problèmes de dégradation. L'optimiseur Adam, réputé pour sa capacité d'adaptation et son efficacité, a été utilisé pour ajuster les poids du modèle.

Le modèle a été entraîné sur plusieurs epochs, avec des callbacks pour l'arrêt anticipé et la sauvegarde du meilleur modèle. Les résultats ont montré une précision impressionnante de 99% sur l'ensemble de validation, démontrant l'efficacité du modèle à généraliser sur des données non vues.

En conclusion, ce projet a démontré comment combiner les techniques de deeplearning et de traitement du signal pour créer un système de reconnaissance vocale performant et précis. Avec une précision de 99%, le modèle est prometteur pour des applications réelles variées, allant de la sécurité à l'assistance vocale, en passant par l'archivage de discours. Cette méthodologie peut être étendue pour inclure plus d'orateurs et de langues, ouvrant la voie à des systèmes de reconnaissance vocale encore plus avancés.

Conclusion générale

En conclusion, notre projet de reconnaissance vocale a atteint ses objectifs en démontrant la capacité de la technologie à répondre aux besoins critiques de sécurité. En développant un modèle capable d'identifier les discours de Jens Stoltenberg, Julia Gillard, Margaret Thatcher et Nelson Mandela, nous avons prouvé que des techniques avancées comme le deeplearning et les architectures de réseaux de neurones convolutifs, telles que ResNet, peuvent offrir des solutions robustes et précises pour l'authentification vocale et la surveillance des communications.

Les résultats obtenus montrent que notre modèle peut réduire les risques de fraude et d'accès non autorisé grâce à une authentification vocale fiable. De plus, il peut analyser et surveiller efficacement des communications sensibles, détectant ainsi les anomalies et les menaces potentielles. Dans des environnements nécessitant une discrétion absolue, notre système s'est avéré être une méthode de vérification d'identité non intrusive et efficace.

En somme, ce projet illustre l'importance et le potentiel de la reconnaissance vocale dans le renforcement de la sécurité. Il ouvre la voie à des applications futures, consolidant la position de cette technologie comme un pilier essentiel dans les systèmes de sécurité modernes. Nous sommes confiants que les avancées réalisées dans ce projet contribueront à l'évolution continue de la reconnaissance vocale, apportant des améliorations significatives dans divers secteurs.

Bibliographie

[1] https://fr.wikipedia.org/wiki/Intelligence_artificielle

[2] https://datascientest.com/machine-learning-tout-savoir

[3] https://datascientest.com/deep-learning-definition

[4] https://datascientest.com/deep-learning-definition

[5] https://datascientest.com/convolutional-neural-

[6] https://culturesciencesphysique.ens-lyon.fr/pdf/IA-apprentissage-Rousseau.pdf

[7] Laurent Buniet. Traitement automatique de la parole en milieu bruité : étude de modèles connexionnistes statiques et dynamiques. PhD thesis, Université Henri Poincaré-Nancy 1, France, 1997. 13

[8] BEGHA Chayma& ZIOUANE Abir .DeepIearning pour la reconnaissance automatique du locuteur Université SAAD DAHLAB de BLIDA 2021,2022

[9] Mlle F. Bendimerad et Mlle S. Siouane , Élaboration d'un système d'Identification Automatique du Locuteur par Quantification Vectorielle , Juin 2013.

[10] Calliope. La parole et son traitement automatique. Collection technique et scientifique des télécommunications, CNET - ENST, Masson, 718 pages, 1989

[11] S. Ouamour, Indexation automatique des documents audio en vue d'une classification par locuteurs-Application a l'archivage des émissions TV et Radio-, thèse de doctorat, Ecole Nationale Polytechnique, Alger/Algérie, 2009.

[12] BEGHA Chayma& ZIOUANE Abir ,DeepIearning pour la reconnaissance automatique du locuteur .Université SAAD DAHLAB de BLIDA 2021-2022

[13] AJGOU Riadh, Reconnaissance Automatique du Locuteur à Travers les Canaux Digitaux, Faculté des Sciences et de la Technologie . 14/02/2016

[14]Commission électrotechnique internationale, « Oscillations, signals et dispositifs en rapport », dans IEC 60050 Vocabulaire électrotechnique international, 2017 (lire en ligne [archive]), p.

702-08-15 « bruit de fond, souffle » ; aussi Commission électrotechnique internationale, « Enregistrement et reproduction de l'audio et de la vidéo », dans IEC 60050 Vocabulaire électrotechnique international, 2017 (lire en ligne [archive]), p. 806-12-15 « bruit de fond ».

[15] Mario Rossi, Audio, Lausanne, Presses polytechniques et universitaires romandes, 2007, 1re éd., p. 285-291 4.4.Bruit de fond

[16] https://ryax.tech/fr/deep-learning-comprendre-les-reseaux-de-neurones-artificiels-artificialneural-networks/

[17]https://www.linkedin.com/advice/0/what-benefits-drawbacks-using-skipconnections?lang=fr&originalSubdomain=fr

[18]https://deepai.org/machine-learning-glossary-and-terms/vanishing-gradient-problem

[19] Laurent Buniet. Traitement automatique de la parole en milieu bruité : étude de modèles connexionnistes statiques et dynamiques. PhD thesis, Université Henri Poincaré-Nancy 1, France, 1997. 13