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

Université Larbi Tébessi

Faculté des Sciences Exactes et Sciences de la Nature et de la Vie

Département des Mathématiques et Informatique

Mémoire
Présenté en vue de l'obtention du diplôme de MASTER
Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Systèmes et multimédias

Par
Hacine Bouchra

Système de télécommandement vocal basé Deep Learning

Date de soutenance 22 Juin 2021

Devant le jury

Dr. Haouam Yassine	*MCB*	*Université Larbi Tébessi*	*Président*
Dr. Merzoug Soltane	*MCB*	*Université Larbi Tébessi*	*Examinateur*
Dr. Mohamed Gasmi	*MCB*	*Université Larbi Tébessi*	*Encadreur*
Mr. AbdEllatif Gahmousse	*MAA*	*Université Larbi Tébessi*	*Co-Encadreur*

Année universitaire : 2020/2021

ملخص

الكلام هو النمط الأكثر استخداما في التواصل الإنساني، من المتوقع أن يصبح الكلام على نحو متزايد جزءًا من واجهات الوسائط المتعددة بين المستخدم والنظام التلقائي، وذلك من ناحية بفضل تحسين قوة النظم التلقائية للتعرف على الكلام، ومن ناحية أخرى، بسبب الوعي المتزايد لعامة الجمهور بهذه التكنولوجيا التي لا تزال غير معروفة ـ

لطالما كان التعرف التلقائي على الكلام تحديًا علميًا.. في السنوات الأخيرة، بذلت جهود بحثية كبيرة لتطوير أنظمة وحلول داعمة لأداء مهام معينة كانت محفوظة حتى الآن للبشر ـ

نقترح في هذا العمل نظام تحكم صوتي عن بعد لموقع ويب يستخدم فقط اللغة العربية ـ يستخدم النظام مصنفًا قائمًا على التعلم العميق والذي يطبق خوارزمية الشبكات العصبية التلافيفية، اخترنا معاملات MFCC لاستخراج خصائص إشارة الكلام ـ

تقدم هذه الأطروحة دراسة بنية شبكة عصبية تلافيفية، تم اختبارها في قاعدة البيانات الخاصة بنا، حيث حصلنا على نتائج جيدة .

الكلمات المفتاحية: التعرف التلقائي على الكلام، التعلم العميق، الشبكات العصبية التلافيفية، MFCC، الإشارة.

Abstract

Speech is the most natural form of human communication. it is anticipated that the speech will increasingly part of multimedia interfaces between a user and an automatic system, firstly by improving the robustness of automatic recognition speech systems, secondly, because of the growing public awareness of this technology is still little known.

Automatic Speech Recognition (ASR) has always been a scientist challenge. Many research efforts have been made over recent years to offer solutions and aiding systems in order to carry out various tasks previously dedicated only to humans.

We propose in this work a remote-control system for a website using only the speech in Arabic language. The system uses a Deep Learning classifier that applies a convolutional neural network algorithm, MFCC coefficients were chosen for extracting speech signal characteristics.

This dissertation provides the study of an architecture of a convolutional neural network, tested on our dataset, where we obtained good results.

Keywords: Automatic Speech Recognition (ASR), Deep Learning, convolutional neural network, MFCC, Signal

Résumé

La parole est la forme la plus naturelle de communication humaine. On peut prévoir que la parole fera de plus en plus partie des interfaces multimédia entre un utilisateur et un système automatique, d'une part grâce à l'amélioration de la robustesse des systèmes de reconnaissance automatique de la parole et d'autre part, du fait de la sensibilisation croissante du grand public à cette technologie encore peu connue.

La Reconnaissance Automatique de la Parole (RAP) demeure depuis toujours un défi scientifique. Au cours de ces dernières années de grands efforts de recherche ont été concrétisés, afin de développer des systèmes d'aide et des solutions permettant d'effectuer certaines tâches jusqu'ici réservées aux humains.

Nous proposons dans ce travail un système de télé-commandement vocal pour un site web en utilisant seulement la parole en langue Arabe. Le système utilise un classifieur basé sur l'apprentissage profond et qui applique un algorithme de réseaux de neurones convolutifs, on a choisi les coefficients MFCC pour l'extraction des caractéristiques du signal de la parole.

Ce mémoire fournit l'étude d'une architecture d'un réseau de neurones convolutifs, testé sur notre base de données, où nous avons obtenu de bons résultats.

Mots clés : Reconnaissance Automatique de la Parole (RAP), Apprentissage approfondi, réseaux de neurones convolutifs, MFCC, Signal.

Annexe

ASR Automatic Speech Recognition

RAP Reconnaissance Automatique de la parole

MFCC Mel Frequency Cepstral Coefficients

LPC Linear Predictive Encoding

PLP Prediction Coefficient Extraction

DTW Dynamic Time Warping

HMMs Hidden Markov Models

DL Deep Learning

CNN convolutional neural network

Dédicaces

Je dédie ce mémoire à :

A mon très cher père ***Hacine Abdelatif,*** *vous avez toujours été pour moi un exemple du père respectueux, honnête, de la personne méticuleuse, je tiens à honorer l'homme que vous êtes.*

Grâce à vous papa j'ai appris le sens du travail et de la responsabilité. Je voudrais vous remercier pour votre amour, votre générosité, votre compréhension... Votre soutien fut une lumière dans tout mon parcours. Aucune dédicace ne saurait exprimer l'amour, l'estime et le respect que j'ai toujours eu pour vous.

Ce modeste travail est le fruit de tous les sacrifices que vous avez déployés pour mon éducation et ma formation. Je vous aime papa et j'implore le tout-puissant pour qu'il vous accorde une bonne santé et une vie longue et heureuse.

A ma chère maman ***Maina Nadia****, Aucune dédicace très chère maman, ne pourrait exprimer la profondeur des sentiments que j'éprouve pour vous, vos sacrifices innombrables et votre dévouement firent pour moi un encouragement, vous avez guetté mes pas, et m'avais couvé de tendresse, votre prière et votre bénédiction m'ont été d'un grand secours pour mener à bien mes études.*

Vous m'avais aidé et soutenu pendant de nombreuses années avec à chaque fois une attention renouvelée. Puisse Allah, tout puissant vous combler de santé, de bonheur et vous procurer une longue vie.

À Mes chères sœurs et ma belle-sœur, *pour leurs encouragements permanents, et leur soutien moral.*

À Mes chers frères, *pour leur appui et leur encouragement.*

À Mes chers neveux et nièces, *aucune dédicace ne saurait exprimer tout l'amour que j'ai pour vous, votre joie et votre gaieté me comblent de bonheur****.***

À Mes chères amies « Safa, Saoussene, Chaima, Ikrame et Meriem », *pour tous les souvenirs éternels dans nos cœurs, je les remercie pour leur soutien et pour m'accompagner.*

Remerciements

Remercier est d'autant plus facile que l'on est conscient de ne pas être arrivé là tout seul. Je tiens tout d'abord à remercier en premier lieu Allah le tout-puissant pour la volonté la santé et la patience, qu'il m'a donnée durant toutes ces langues années.

Toute ma reconnaissance et remerciement vont à mes encadrants Mr ***Gahmousse Abdelatif*** *et Mr* ***Gasmi Mohammed****, pour avoir accepté de diriger ce travail, leurs encouragements, leur disponibilité et pour leur rigueur scientifique, pour leurs nombreux conseils qui ont contribué à la réalisation de ce travail, pour la confiance qu'ils m'ont accordée, et leur soutien scientifique et moral.*

Je remercie également les membres du jury pour l'honneur qu'ils m'ont fait, en acceptant d'examiner ce travail et d'apporter leurs critiques enrichissantes, veuillez trouver ici l'expression de ma sincère reconnaissance.

Mes plus profonds remerciements vont à mes parents pour leurs présences et leur accompagnement pendant tout mon parcours et dans les moments de doute. Tout au long de mon cursus, ils m'ont toujours soutenue et encouragé dans la poursuite de mes études. Ils ont su me donner toutes les chances pour réussir. Ils m'ont donné le goût de la connaissance. Qu'ils trouvent, dans la réalisation de ce travail, l'aboutissement de leurs efforts ainsi que l'expression de ma plus affectueuse gratitude. Je leur exprime ici toute ma gratitude de m'avoir toujours écoutée et même souvent relue avec la plus grande attention et m'apporter chaque jour tant et plus.

Je remercie sincèrement tous les enseignants qui ont contribué de loin ou de près à ma formation du primaire au supérieur.

Liste de Tableaux

Liste de Figures

Table des matières

Introduction générale

Introduction générale

La parole est le moyen de communication le plus naturel. Grâce à elle, nous pouvons exprimer nos souhaits et nos idées. On peut l'utiliser pour exprimer des opinions, des pensées, des sentiments, des souhaits ou pour échanger, transmettre ou demander des informations. Aujourd'hui, nous l'utilisons non seulement pour communiquer avec d'autres personnes, mais aussi pour communiquer avec des machines.

La reconnaissance automatique de la parole est une technologie qui permet d'analyser les sons captés par un microphone et de les retranscrire en une série de mots utilisables par les machines. Depuis sa création dans les années 1950, avec l'aide de phonétiques, de linguistes, de mathématiciens et d'ingénieurs, qui ont défini les connaissances acoustiques et linguistiques nécessaires à la compréhension de la parole humaine, la reconnaissance automatique de la parole n'a cessé de s'améliorer.

Cependant, les performances obtenues ne sont pas parfaites et dépendent de nombreuses normes. Les avantages de la reconnaissance automatique de la parole incluent la parole native, appartenant à un seul locuteur avec des mots corrects (pas de pathologie de la parole), enregistrée dans un environnement calme et sans bruit, basée sur un vocabulaire commun (mots connus par le système). Lorsqu'il s'agit d'accents non natifs, de dialectes différents, de locuteurs mal prononcés, de mots inconnus du système (généralement des noms propres) et des signaux audios bruités (faible rapport signal/bruit), les performances du système se dégradent.

Les applications de la reconnaissance automatique de la parole sont très diversifiées, et chaque système a sa propre architecture et son propre mode de fonctionnement. Plus le champ d'application est grand, plus le modèle de reconnaissance doit être grand (pour comprendre la parole spontanée et la diversité des locuteurs). De nos jours, de nombreuses recherches sur la reconnaissance automatique de la parole sont faites pour imiter les assistants personnels : recherche d'informations sur Internet, prise de rendez-vous, envoi de SMS, contrôle de la domotique, etc.

Dans ce contexte, un de ces systèmes a été choisi d'être étudié dans ce mémoire, c’est un système de télé-commandement vocal basé Deep Learning pour un site web en utilisant seulement la parole en langue Arabe.

Nous avons choisi d'articuler notre étude autour de trois chapitres principaux :

Le premier chapitre intitulé « Contexte d'étude et état de l'art », dont on a présenté une brève description de la reconnaissance automatique de la parole, on rappelle leur historique et parlons en général à tout ce qui concerne la parole et la reconnaissance de la parole.

Le deuxième chapitre intitulé « Approche proposé », il contient la modélisation de l'approche proposé.

Le dernier chapitre intitulé « Implémentation et expérimentation », où l'on présente l'implémentation réalisée et les résultats obtenus, et proposer quelques perspectives sur les travaux futurs.

A la fin de ce travail nous concluons par une conclusion générale qui résume nos contributions.

Chapitre 01 :
Contexte d'étude et état de l'art

1. Introduction à la reconnaissance automatique de la parole

La reconnaissance automatique de la parole RAP « Automatic Speech Recognition ASR en anglais » est une technique informatique qui permet de décoder des informations fournies oralement par un utilisateur humain capté au moyen d'un microphone, pour les transcrire sous une forme exploitable par la machine.

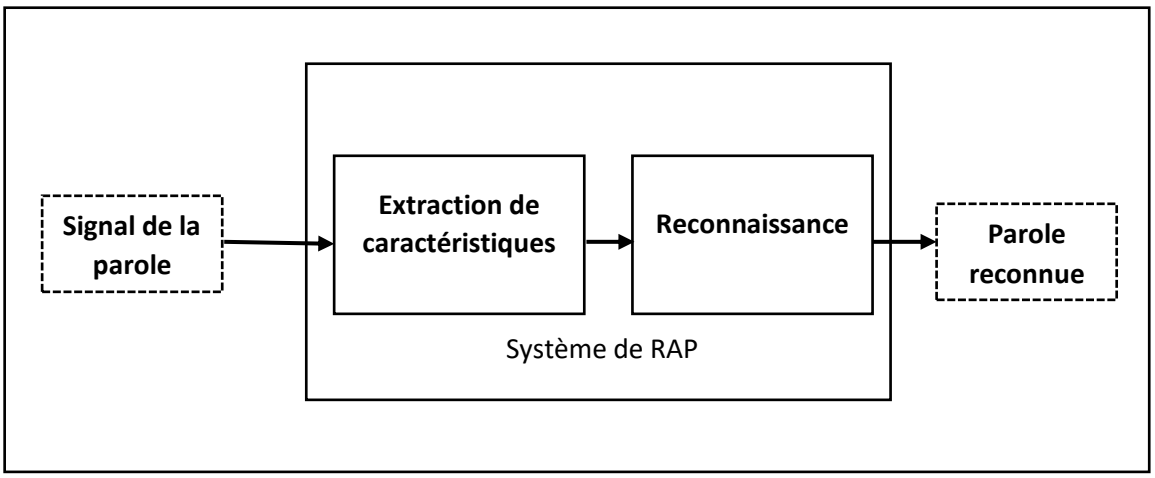

Figure 1.1 : Représentation du processus générale du RAP

Alors, la reconnaissance automatique de la parole a pour but de permettre à un utilisateur de s'adresser oralement à une machine (interface homme-machine) pour des taches diverses tel que : la commande (comme les ordinateurs, les automobiles, les appareils ménagers…), la traduction, la dictée, etc...

Selon l'information à extraire, on distingue deux types de reconnaissances [1] :

- La reconnaissance du locuteur : dont le but est de reconnaitre la personne qui parle parmi une population de locuteurs (identificateur) ou de vérifier son identité (vérificateur).
- La reconnaissance de la parole : dont le but est de transcrire l'information symbolique exprimée par le locuteur.

Le schéma ci-dessous présente les étapes de la reconnaissance de la parole. Les phrases enregistrées seront fournies au programme de reconnaissance automatique de la parole (RAP). Dans le formalisme RAP, les fonctions sont réparties comme suit :

- Données acoustiques : permet principalement d'extraire du signal vocal des vecteurs acoustiques. Le signal est numérisé et paramétré par une technique d'analyse fréquentielle utilisant la transformée de Fourier.
- L'apprentissage automatique : réalise une association entre les segments élémentaires de la parole et les éléments lexicaux. Cette association fait appel à

une modélisation statistique en utilisons les modèles de Markov cachés et/ou réseaux de neurones artificiels.

- Le décodage : concaténation des modèles élémentaires précédemment appris pour reconstituer le discours le plus probable

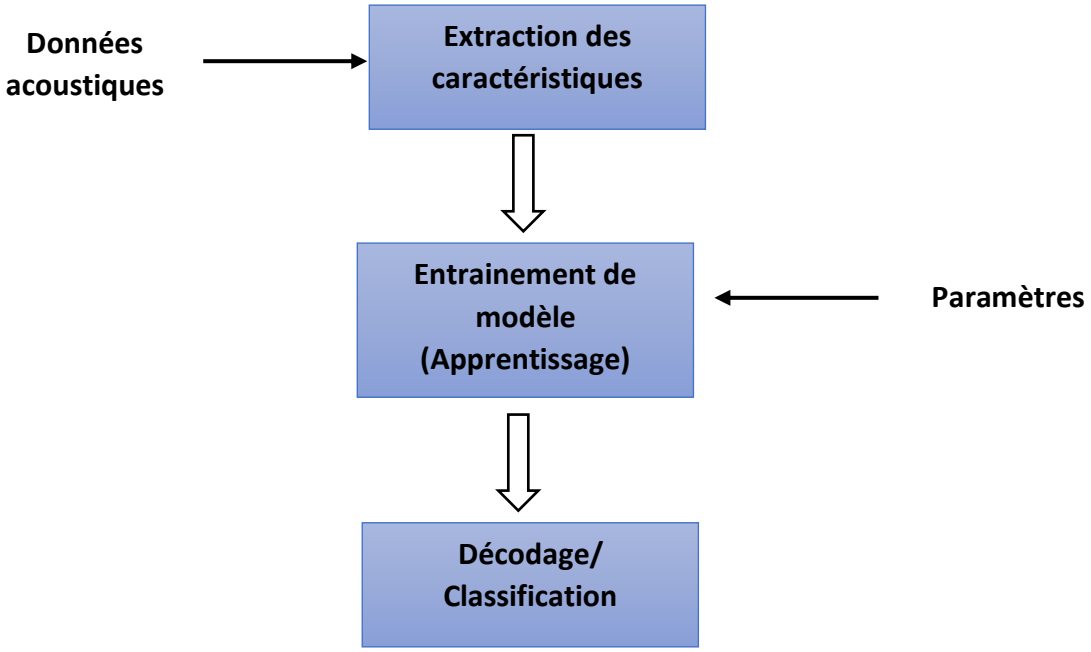

Figure 1.2 : Schéma général d'un système de reconnaissance de la parole.

Le schéma ci-dessous présente la reconnaissance du locuteur. La vérification du locuteur est le processus consistant à déterminer si l'identité déclarée d'un message vocal correspond à la véritable identité du locuteur. La réponse est binaire, accepter ou rejeter. Les éléments mis en jeu sont donc une identité proclamée et la référence associée à un échantillon connu de l'identité proclamée. [8]

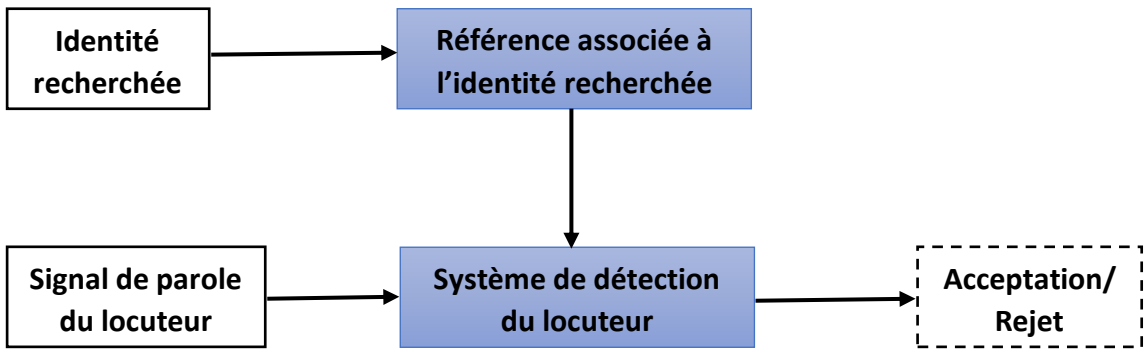

Figure 1.3 : Schéma général d'un système de reconnaissance du locuteur.

La RAP permet de contrôler-commander un outil et de dicter des mots. Il peut être utilisé aussi bien dans le domaine industriel que dans le domaine de l'automobile (par exemple, pour saisir une destination en GPS) et de la domotique (par exemple, pour programmer la température d'une maison). Il aide également les personnes handicapées (y compris la surdité partielle ou totale) ou les apprenants de langues à communiquer. Et voilà quelques applications connues de la RAP : **Cortana** (Microsoft), **Siri** (Apple), **Google Now** (Google), **Alexa** (Amazon), **Vocapia Research** (VoxSigma suite), **Vocon Hybrid et Dragon** (respectivement dictée par grammaire et dictée libre par Nuance Communications).

Les avantages de la reconnaissance automatique de la parole sont multiples. Contrairement à l'écran et au clavier, il libère complètement l'utilisation de la vue et des mains, et laisse l'utilisateur libre de ses mouvements.

De plus, la vitesse de transmission des informations vocales est naturellement plus rapide que celle de l'écriture manuscrite. Enfin, presque tout le monde peut parler, alors que peu de gens sont à l'abri des fautes de frappe et d'orthographe. Ces avantages sont : la commande de machines ou de robots, la dictée vocale, la messagerie, l'aide aux handicapés…

Ce chapitre est une introduction générale à la reconnaissance automatique de la parole caractérisé actuellement par le développement des applications et par un effort de recherche toujours important, pour augmenter la fiabilité et la robustesse des systèmes.

Cependant on va donner un bref historique des travaux et recherches antérieures et des difficultés qu'on peut rencontrer dans ce domaine et nous résumons par une petite conclusion.

2. Historique de la reconnaissance automatique de la parole

On a coutume de fixer l'origine des recherches en reconnaissance vocale aux années 1950[2]. C'est à cette époque en effet qu'IBM commence à investir dans ce domaine avec comme objectif de développer une nouvelle forme d'interaction entre l'homme et la machine.

Il est cependant intéressant de mentionner que, un siècle plus tôt, on s'intéressait déjà au problème connexe qu'est la synthèse vocale, c'est-à-dire aux possibilités de faire parler des machines. C'est ainsi qu'en 1846, un certain Joseph Faber construisait à Londres un "orgue vocal" capable de reproduire des phrases ordinaires et même de chanter la "God Save the Queen" ! Plus tard, en 1890, Edison mettait sur le marché une poupée parlante à 10$ (une somme équivalente au salaire de deux semaines de travail de l'époque).

A la fin des années cinquante, IBM développe le premier ordinateur entraîné à écouter des modèles spécifiques de sons et à dégager des corrélations statistiques entre ces sons et les mots qui y correspondent.

En 1964, IBM fait la première démonstration de reconnaissance vocale : le logiciel "ShoeBox" permet de reconnaître une série de chiffres dictés. Cette démonstration incite le ministère américain de la défense à financer un programme de recherche pour développer cette nouvelle technologie. C'est également ainsi que naît l'approche statistique dans le domaine de la reconnaissance vocale et que les techniques d'apprentissage voient le jour, techniques basées sur des algorithmes statistiques habituellement utilisés dans les théories de l'information.

En 1984, IBM présente le premier système de reconnaissance vocale au monde disposant d'un lexique de 5000 mots et bénéficiant d'un taux de reconnaissance de 95%. Ce logiciel nécessite 3 processeurs vectoriels et un grand système 4341 avec une interface utilisateur fonctionnant sur un ordinateur Apollo. Le logiciel permet à un utilisateur expérimenté de dicter ses textes en mode discret, c'est-à-dire en marquant une pause entre chaque mot. La même année, Philips commence le développement de "SPICOS", un logiciel de reconnaissance avec un vocabulaire de 1000 mots.

Dans les années suivantes, les développements vont s'accélérer. La puissance croissante des processeurs (et leur diminution de coût) va en effet permettre d'améliorer constamment les performances des algorithmes utilisés et également de traiter ces algorithmes par des logiciels et non plus par du hardware dédicacé. Plus tard encore, l'émergence de la carte son Soundblaster de Creative Labs comme standard de fait va encore favoriser le développement et la diffusion de ces logiciels sur les postes de travail PC compatibles.

A partir des années 90, de nouveaux acteurs se lancent dans ce marché et de nouveaux produits font leur apparition. Dragon Systems annonce la sortie de son premier logiciel de dictée en 1990 ; Apple lance en 1993 son produit "Plain Talk" ; en 1994, IBM commercialise "IBM Personal Dictation System" pour PC OS/2.

En 2008 Google lance une application de recherche sur Internet mettant en œuvre une fonctionnalité de reconnaissance vocale, Apple propose l'application Siri sur ses téléphones en 2011[9], puis en 2017 Microsoft annonce égaler les performances de reconnaissance vocale des êtres humains [10].

3. Les modules de SRAP

Les règles strictes pour la reconnaissance automatique de la parole sont souvent décomposées en plusieurs modules qui sont [3] :

- ✓ Un analyseur acoustique (sert à paramétriser le signal) : extraire les informations caractéristiques du signal de parole en éliminant au maximum les parties redondantes.
- ✓ Les modèles acoustiques : qui doivent représenter au mieux les unités acoustiques choisies (phonèmes, mots. . .).
- ✓ Les modèles linguistiques : qui doivent être une représentation la plus vraisemblable possible du langage.
- ✓ Le dictionnaire : qui doit contenir l'ensemble des mots que l'on souhaite pouvoir reconnaitre.
- ✓ Le système de reconnaissance.

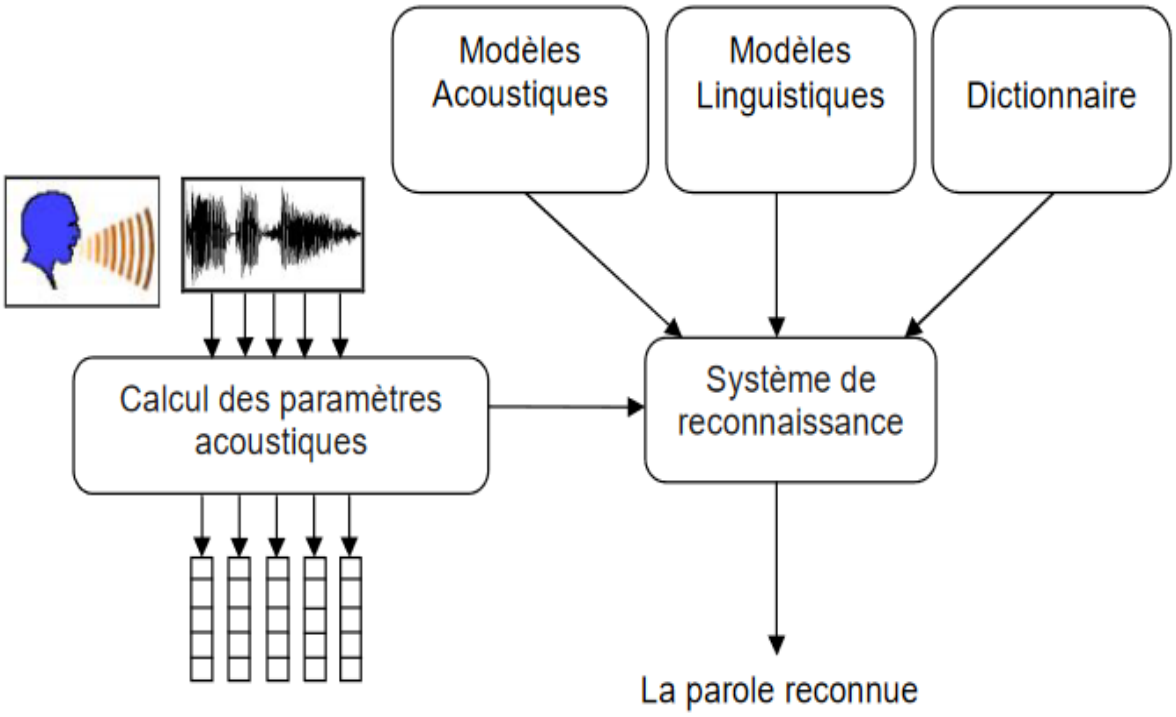

Figure 1.4 : Schéma principal d'un système de reconnaissance automatique de la parole [3]

3.1. Performances d'un SRAP

Le taux de performance ou taux de reconnaissance permet de mesurer l'efficacité du système de reconnaissance testé. Ce taux varie fortement selon le type de canal de transmission utilisé (microphone, téléphone), la taille du vocabulaire, et le type d'élocution.

Il existe différentes valeurs mesurant les performances d'un système de reconnaissance [4] [5] :

1. Le taux de reconnaissance (taux d'erreur) : pourcentage de mots ou de phrases reconnus correctement.
2. Le taux de substitution : pourcentage de mots pour lesquels le système a commis une erreur.
3. Le taux suppression : pourcentage de mots non détectés.
4. Le taux d'insertion : pourcentage de mots reconnus alors qu'aucun mot n'a été prononcé.

4. Approches de reconnaissance de la parole

Le problème de la reconnaissance automatique de la parole est d'extraire les informations échantillonnées par le signal de parole (le signal électrique obtenu en sortie du microphone est généralement échantillonné à 8 kHz dans le cas d'une ligne téléphonique, ou de 10 kHz à 16 kHz dans le cas du microphone, et peut y arriver jusqu'à 44,1 kHz dans d'autres cas). Bien que cela pose également le problème de la compréhension de la parole. [6]

Il existe deux approches de reconnaissance vocale, selon le type de reconnaissance, qu'il s'agisse d'un mot isolé ou d'une parole continue :

- Approche globale (Reconnaissance par comparaison à des exemples).
- Approche analytique (Reconnaissance par modélisation d'unités de parole).

Les principes sont à peu près le même que ce soit pour l'approche analytique ou l'approche global, La différence entre les deux méthodes est l'entité à reconnaître : pour la première il s'agit du phonème, pour la deuxième est mot.

4.1. Approche globale :

L'idée de l'approche globale est très simple dans son principe, consiste à faire prononcer un ou plusieurs exemples de chacun des mots susceptibles d'être reconnus, et à les enregistrer sous forme de vecteurs acoustiques (typiquement : un vecteur de coefficients LPC ou assimilés toutes les 10 ms). Puisque cette suite de vecteurs acoustiques caractérise complètement l'évolution de l'enveloppe spectrale du signal enregistré, on peut dire qu'elle correspond à un l'enregistrement d'un spectrogramme. L'étape de reconnaissance consiste alors à analyser le signal inconnu sous la forme d'une suite de vecteurs acoustiques similaires, et à comparer la suite inconnue à chacune des suites des exemples préalablement enregistrés. Le mot « reconnu » sera alors celui dont la suite de vecteurs acoustique (le « spectrogramme ») colle le mieux à celle du mot

inconnu. Il s'agit en quelque sorte de voir dans quelle mesure les spectrogrammes se superposent.[6]

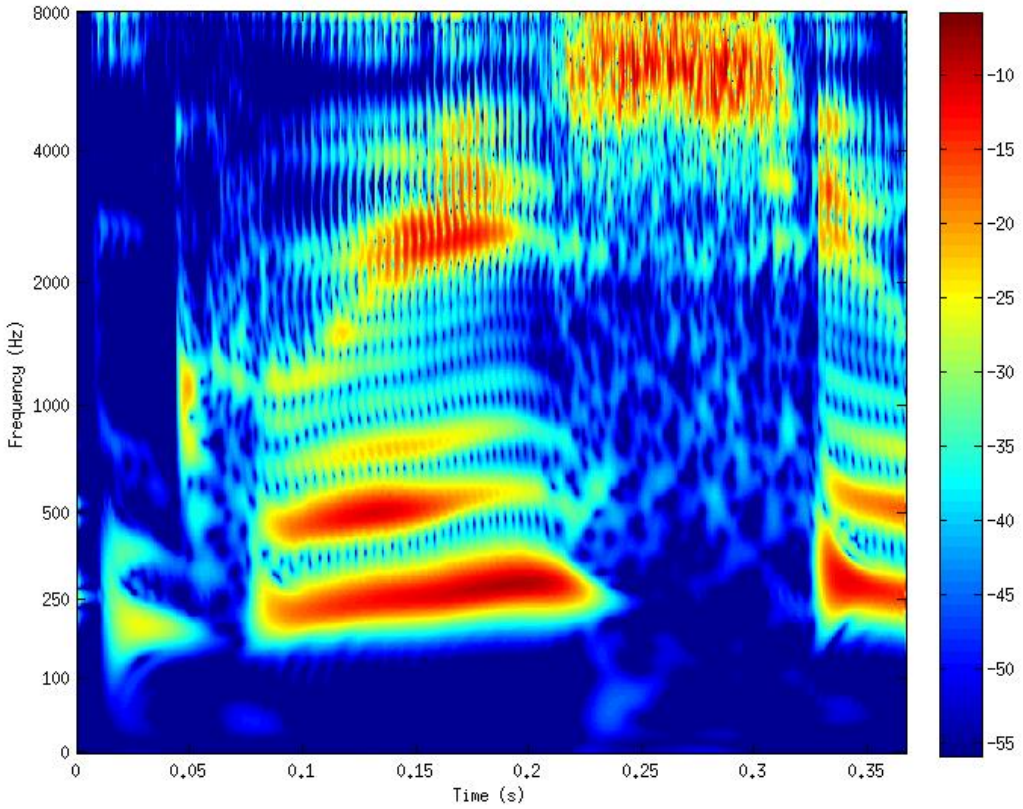

Figure 1.5 : Exemple d'un spectrogramme

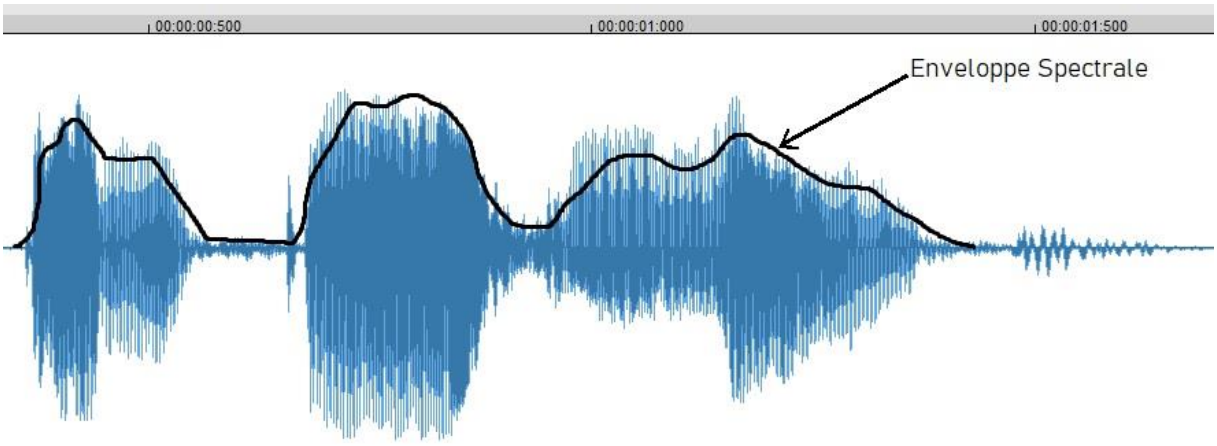

Figure 1.6 : Enveloppe spectrale

Ce principe de base n'est pas implémentable directement : un même mot peut en effet être prononcé d'une infinité de façons différentes, en changeant le rythme de l'élocution. Il en résulte des spectrogrammes plus ou moins distordus dans le temps. La

superposition du spectrogramme inconnu aux spectrogramme de base doit dès lors se faire en acceptant une certaine « élasticité » sur les spectrogrammes candidats. Cette notion d'élasticité est formalisée mathématiquement par un algorithme désormais bien connu : l'algorithme DTW (Dynamic Time Warping, en anglais).

On comprend aisément qu'une telle technique soit intrinsèquement limitée par la taille du vocabulaire à reconnaître (une centaine de mots tout au plus) et qu'elle soit plus propice à la reconnaissance monolocuteur (une reconnaissance multilocuteur imposerait d'enregistrer, de stocker, et surtout d'utiliser pour la comparaison, de nombreux exemples pour chaque mot). Les résultats obtenus, dans le contexte monolocuteur/petit vocabulaire, sont aujourd'hui excellents (proches de 100%).

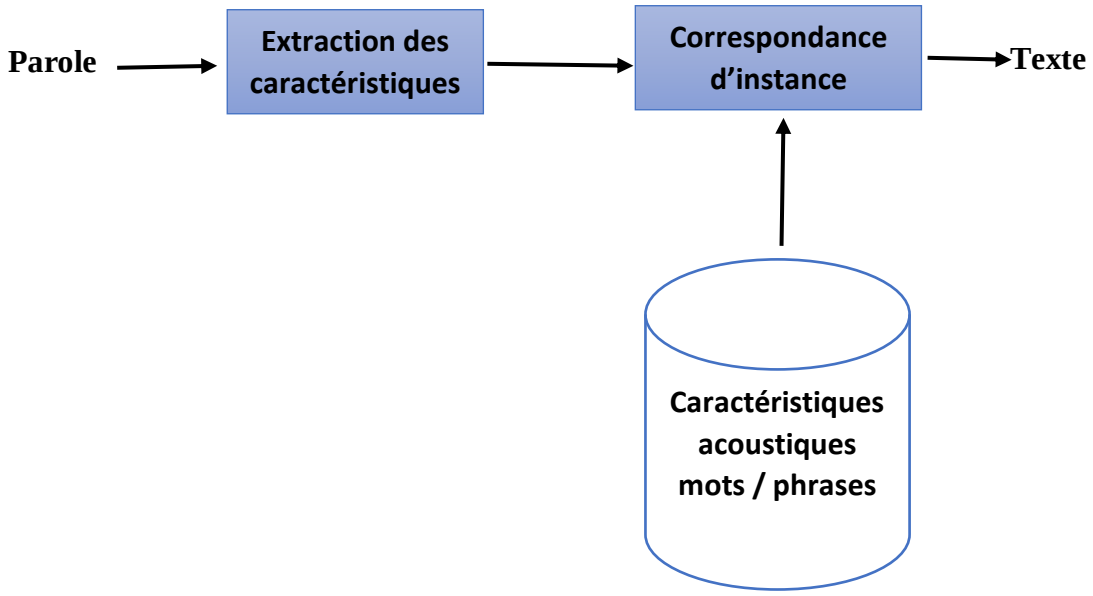

Figure 1.7 : Exemple de la méthode globale

4.2. Approche analytique :

Dès que l'on cherche à concevoir un système réellement multilocuteur, à plus grand vocabulaire, et s'adaptant facilement à une application, il devient nécessaire de mener la reconnaissance sur base d'unités de parole de plus petite taille (phonème). On ne se contente plus alors d'exemples de ces unités, mais on cherche plutôt à en déduire un modèle (un modèle par unité), qui sera applicable pour n'importe quelle voix.

Le formalisme de reconnaissance de la parole est alors souvent décomposé en plusieurs modules, généralement au nombre de quatre :

- Un module de traitement du signal et d'analyse acoustique qui transforme le signal de parole en une séquence de vecteurs acoustiques.
- Un module acoustique qui peut produire une ou plusieurs hypothèses phonétiques pour chaque segment de parole de 10ms (c'est-à-dire, pour chaque vecteur acoustique), associées en général à une probabilité. Ce générateur d'hypothèse locale est généralement basé sur des modèles statistiques d'unités élémentaires de parole (typiquement des phonèmes) qui sont entraînés sur une grande quantité de données de parole (par exemple, enregistrement de nombreuses phrases) contenant plusieurs fois les différentes unités de parole dans plusieurs contextes différents. Ces modèles statistiques sont le plus souvent constitués de lois statistiques paramétriques dont on ajuste les paramètres pour « coller » au mieux aux données, ou de réseaux de neurones artificiels (ANN : Artificial Neural Networks). Un tel générateur d'étiquettes phonétiques intègre toujours un module d'alignement temporel qui transforme les hypothèses locales (prises sur chaque vecteur acoustique indépendamment) en une décision plus globale (prise en considérant les vecteurs environnants). Ceci se fait le plus souvent via des modèles de Markov cachés (HMM pour ''Hidden Markov Model'', en anglais). L'ensemble (lois statistiques paramétriques ou réseau de neurones +HMM) constitue le modèle acoustique sous-jacent à un reconnaisseur de parole.

- Un module lexical qui interagit avec le module d'alignement temporel pour forcer le reconnaisseur à ne reconnaître que des mots existants effectivement dans la langue considérée. Un tel module lexical embarque en général des modèles des mots de la langue (les modèles de base étant de simples dictionnaires phonétiques ; les plus complexes sont de véritables automates probabilistes, capables d'associer une probabilité à chaque prononciation possible d'un mot).

- Un module syntaxique qui interagit avec le module d'alignement temporel pour forcer le reconnaisseur à intégrer des contraintes syntaxiques, voire sémantiques. Les connaissances syntaxiques sont généralement formalisées dans un modèle de la langue, qui associe une probabilité à toute suite de mots présents dans le lexique.[6]

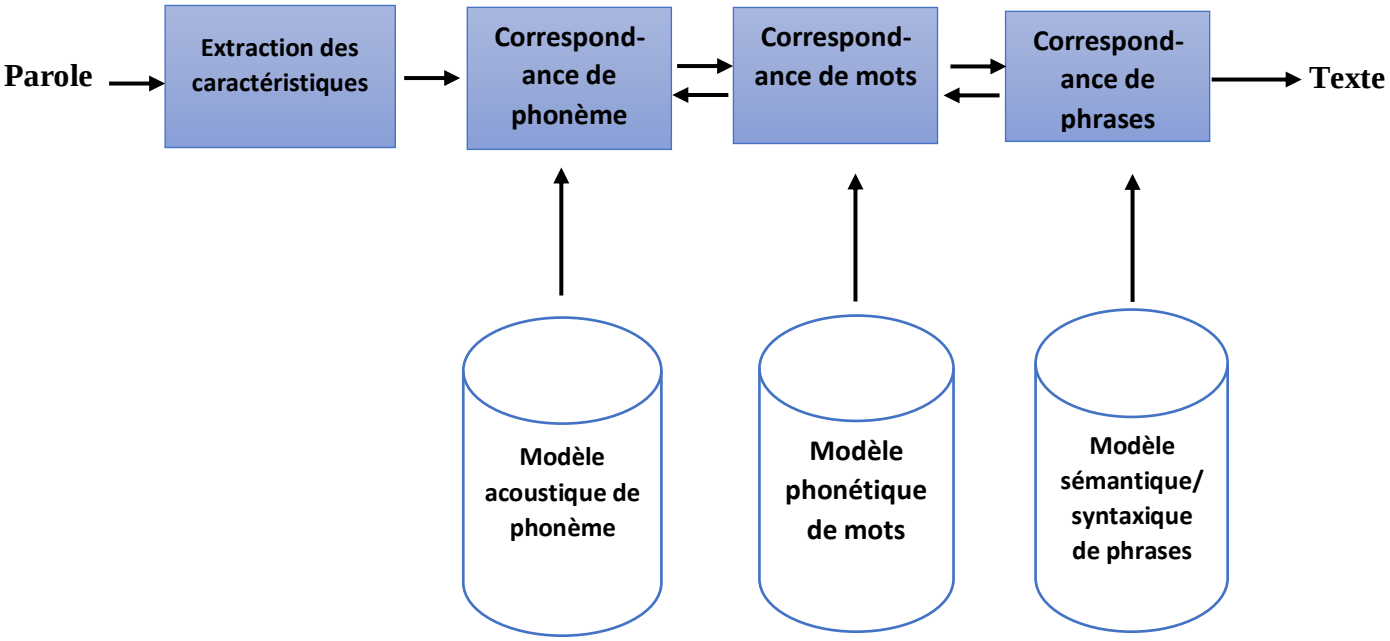

Figure 1.8 : Exemple de l'approche analytique [6]

5. Applications des SRAP

Les systèmes de reconnaissance automatique de la parole sont de plus en plus répandus et utilisés dans des conditions acoustiques très variées, on peut citer : saisie d'informations, machine à dicter et bureautique, commande de machines « mains libres », avions, hélicoptères, systèmes de guidage d'automobiles (GPS), traduction automatique, jouets, aide aux handicapés.

Dans ces applications de reconnaissance vocale, nous distinguerons les systèmes de dictée et les systèmes de commandes vocales, vu notre problématique on ne s'intéresse qu'à cette catégorie des systèmes de commandes vocales.

5.1. Les systèmes de dictée

Les systèmes de dictée constituent le problème le plus difficile à résoudre dans le domaine de la reconnaissance vocale. Comme pour les systèmes de commande vocale, les applications de dictée peuvent être divisées en plusieurs catégories en fonction de leur complexité [7] :

- Les systèmes de reconnaissance discrète : On classe ici les systèmes où l'utilisateur doit parler avec de courtes pauses entre chaque mot ; ce sont les premiers systèmes de dictée qui ont été développés dans les années 80, nous mentionnons parmi ces systèmes le « **Dictaphone** » est une marque déposée par Columbia Graphophone Company en 1907 pour un appareil enregistreur de dictée, « **NaturallySpeaking** » premier logiciel de dictée vocale par la société Dragon.

- Les systèmes de reconnaissance continue : permettre à un utilisateur de dicter son texte à l'ordinateur, de façon continue, avec un vocabulaire riche, à une vitesse de locution normale et avec une reconnaissance proche de 100%.

5.2. Les systèmes de commandes vocales

Les applications de ce type permettent notamment à l'utilisateur de contrôler verbalement des équipements. Par complexité croissante, on peut classer ces systèmes en trois groupes [7] :

- Les systèmes à reconnaissance discrète : Ce sont les applications où soit un nombre limité de mots soit de courtes phrases peuvent être utilisés pour commander le système. On y retrouve par exemple les applications téléphoniques où l'on peut choisir vocalement un point de menu (navigation interactive), le contrôle vocal des commandes de menus dans les logiciels ("fermer fichier, sortir"), certains logiciels de saisie automatique de données où les valeurs sont à choisir dans une liste limitée connue.

- Les systèmes à reconnaissance « à la volée » : Ces systèmes permettent à l'utilisateur de s'exprimer par des phrases mais sont entraînés à repérer certains mots de la phrase, mots qui se trouvent dans son dictionnaire interne et sur lesquels ils basent leur action. La consultation d'un horaire de chemin de fer est un exemple de ce type de système : sur base de la phrase "je voudrais me rendre de Bruxelles à Paris lundi prochain", le système repérera "Bruxelles", "Paris" et "lundi" pour proposer l'horaire correspondant.

- Les systèmes à reconnaissance continue : on trouve ici les applications les plus avancées de commande vocale où l'on peut s'adresser au système en langage naturel. On trouve des exemples dans les systèmes évolués de dictée où l'on eut inclus des commandes vocales évoluées comme "souligner et mettre en gras le troisième mot de ce paragraphe".

Nous citons quelques systèmes de commande vocale : Cortona, Alexa, Siri …

6. Les difficultés de la reconnaissance automatique de parole

De toute évidence, le signal vocal est l'un des signaux les plus complexes. Outre la complexité physiologie inhérente au système de parole et les problèmes de prononciation conjointe qui en découlent, la bande sonore entre les personnes est également très différente, et la mesure du signal de parole est également fortement affectée par la fonction de transfert (y compris les équipements d'acquisition et de transmission ainsi que l'influence du milieu ambiant) [19].

Pour bien comprendre le problème de la reconnaissance automatique de la parole, il est préférable de comprendre les différences de complexité et les différents facteurs qui la rendent difficile.

- **Premièrement, il y a le problème de la variabilité intra et inter locuteurs. Le système est-il lié au locuteur (optimisé pour un locuteur spécifique) ou indépendant du locuteur (peut identifier n'importe quel utilisateur) ?**

De toute évidence, un système indépendant du locuteur est plus facile à développer et a un meilleur taux de reconnaissance qu'un système dépendant du locuteur, car la variabilité du signal de la parole est plus limitée. Cette dépendance au locuteur est cependant acquise au prix d'un entraînement spécifique à chaque utilisateur. Ceci n'est cependant pas toujours possible. Par exemple, dans le cas d'applications téléphoniques, il est évident que les systèmes doivent pouvoir être utilisés par n'importe qui et doivent donc être indépendants du locuteur. Bien que la méthodologie de base reste la même, cette indépendance au locuteur est cependant obtenue par l'acquisition de nombreux locuteurs (couvrant si possible les différents dialectes) qui sont utilisés simultanément pour l'entraînement de modèles susceptibles d'en extraire toutes les caractéristiques majeures. Une solution intermédiaire parfois utilisée est de développer des systèmes capables de s'adapter (de façon supervisée ou non supervisée) rapidement au nouveau locuteur.

- **Le système reconnaît-il des mots isolés ou de la parole continue ?**

De toute évidence, il est beaucoup plus facile d'identifier des mots isolés bien espacés en silence par rapport au reconnaître la séquence de mots constituant une phrase. En effet, dans ce dernier cas, non seulement la frontière entre mots n'est plus connue mais, de plus, les mots deviennent fortement articulés (c'est-à-dire que la prononciation de chaque mot est affectée par le mot qui précède ainsi que par celui qui suit - un exemple simple et bien connu étant les liaisons du français).

Dans le cas de la parole continue, le niveau de complexité varie également selon qu'il s'agisse de texte lu, de texte parlé ou, beaucoup plus difficile, de langage naturel avec ses hésitations, phrases grammaticalement incorrectes, faux départs, etc.

Un autre problème, qui commence à être bien maîtrisé, concerne la reconnaissance de mots clés en parole libre. Dans ce dernier cas, le vocabulaire à reconnaître est relativement petit et bien défini mais le locuteur n'est pas contraint de parler en mots isolés. Par exemple, si un utilisateur est invité à répondre par « oui » ou « non », il peut répondre « oui, s'il vous plait ». Dans ce contexte, un problème qui reste particulièrement difficile est le rejet de phrases ne contenant aucun mots clés.

- **La taille du vocabulaire et son degré de confusion :**

La taille du vocabulaire et son degré de confusion sont également des facteurs importants. Les petits vocabulaires sont évidemment plus faciles à reconnaître que les grands vocabulaires, étant donné que dans ce dernier cas, les possibilités de confusion augmentent. Certains petits vocabulaires peuvent cependant s'avérer particulièrement difficiles à traiter ; ceci est le cas, par exemple, pour l'ensemble des lettres de l'alphabet, contenant surtout des mots très courts et acoustiquement proches.

- **Le système est-il robuste ? : c'est à dire capable de fonctionner proprement dans des conditions difficiles ?**

En effet, de nombreuses variables pouvant affecter significativement les performances des systèmes de reconnaissance ont été identifiées :

- ✓ Les bruits d'environnement tels que bruits additifs stationnaires ou non-stationnaires (par exemple, dans une voiture ou dans une usine).
- ✓ Acoustique déformée et bruits (additifs) corrélés avec le signal de parole utile (par exemple, distorsions non linéaires et réverbérations).
- ✓ Utilisation de différents microphones et différentes caractéristiques (fonctions de transfert) du système d'acquisition du signal (filtres), conduisant généralement à du bruit de convolution.
- ✓ Bande passante fréquentielle limitée (par exemple dans le cas des lignes téléphoniques pour lesquelles les fréquences transmises sont naturellement limitées entre environ 350Hz et 3200Hz).
- ✓ Elocution inhabituelle ou altérée, comprenant entre autres : l'effet Lombard, (qui désigne toutes les modifications, souvent inaudibles, du signal acoustique lors de l'élocution en milieu bruité), le stress physique ou émotionnel, une vitesse d'élocution inhabituelle, ainsi que les bruits de lèvres ou de respiration.

Certains systèmes peuvent être plus robustes que d'autres à l'une ou l'autre de ces perturbations, mais en règle générale, les reconnaisseurs de parole actuelle restent encore trop sensibles à ces paramètres.

7. Travaux connexes

Sandeep Rathor et al. [11] proposent un système de reconnaissance de la parole hindi, son domaine allant de la reconnaissance de la parole hindi à la reconnaissance de la parole d'origine indienne comme Bangla et Marathi. Ils ont utilisé le CMU sphinx 4 comme modèle de base de la reconnaissance vocale et l'ont construit sur un modèle de langage acoustique et un dictionnaire phonème spécifique à la langue.

Tan Baohua et al. [12] ont proposé un système de commande à distance de la parole basé sur Computer Telecommunication Integration (CTI), Ce système a adopté la technologie CTI pour réaliser une commande vocale à distance sur un robot de l'industrie à travers un contrôleur logique de mouvements basé sur les composants de service Web. Le système a quatre principales fonctions, qui sont « TTS module (Text To Speech, attaché à la technologie de synthèse vocale), IVR module (Interactive Voice Response, fournit un service de réponse vocale interactive), ASR module (Automatic Speech Recognition, Son traitement de configuration comprend la carte vocale initiale, l'invocation de la carte vocale, le réglage de l'environnement de reconnaissance vocale et l'initialisation de l'application principale, etc.) et contrôleur logique mouvements (ils ont conçu un modèle de mouvement simple, qui ne contenait que deux modes de mouvement translation et rotation et quatre degrés de mouvement libre)». Les résultats des tests montrent que le système fonctionnait normalement dans diverses conditions d'essai.

Hae-Duck J. Jeong et al. [14] ont construit un système informatique de contrôle à distance utilisant les technologies de reconnaissance vocale des appareils mobiles pour les personnes aveugles et handicapées. La configuration du système se compose d'un smartphone, un serveur PC et un serveur Google qui sont connectés les uns aux autres

George E Dahi et al. [15] ont proposé d'établir un système basé sur un modèle subordonné pour la reconnaissance vocale de vocabulaire énorme qui ont des progrès tardifs en utilisant le modèle HMM de réseau neuronal profond. C'est la substitution du Gaussian mixture model. En utilisant cette méthodologie, la prédiction peut être améliorée et réduite et la précision des mots peut être élargie. Les résultats sont apparus dans ce document et indique

que c'est le modèle acoustique le plus dominant pour le LVSR (large-vocabulary speech recognition) par rapport à tout ce qui a été dit au sujet du modèle.

Chee Yang Loh et al. [16] ont implémenté un système interactif de reconnaissance de la parole pour le véhicule qui est utilisé pour effectuer certaines actions utilisant des commandes vocales par le conducteur ou les passagers. Plusieurs algorithmes et fonctions sont utilisés pour effectuer le processus de correspondance du système qui sont Mel Frequency Cepstral Coefficients (MFCC) et Vector Quantization using Linde-Buzo-Gray (VQLBG). Ils ont signalé un taux de reconnaissance de 78,57 %.

Paul Jasmin Rani et al. [17] ont construit un système de commande vocale pour la maison, basé sur l'internet des objets (IoT), intelligence artificielle et Natural Language Processing (NLP). L'utilisateur envoie une commande par la parole à l'appareil mobile, qui interprète le message et envoie la commande appropriée à l'appareil spécifique. La commande vocale donnée par l'utilisateur est interprétée par l'appareil mobile à l'aide du traitement Natural Language. L'appareil mobile sert de console centrale, elle détermine quelle opération doit être effectuée par quel appareil pour répondre à la demande de l'utilisateur.

Mohammad A. M. Abu Shariah et al. [18] ont développé un système de reconnaissance automatique de la parole à mots isolés (IWASR) fondé sur Vector Quantization (VQ). Ce système reçoit, analyse, recherche et fait correspondre un signal vocal d'entrée avec l'ensemble de signaux vocaux formés qui sont stockés dans le codebook, et retourne les résultats de correspondance aux utilisateurs. Pour extraire les caractéristiques des signaux vocaux, l'algorithme Mel-Frequency Cepstral Coefficients (MFCC) a été appliqué. Les résultats expérimentaux ont montré que le taux de reconnaissance a été amélioré avec l'augmentation de la taille du codebook et ont montré que la taille du codebook de 81 vecteurs caractéristiques avait un taux de reconnaissance dépassé 85 %.

8. Conclusion

La reconnaissance automatique de la parole est l'une des tâches pionnières de l'intelligence artificielle, son objectif est de reconnaître la séquence de phonèmes dans un signal de la parole à l'aide d'un dispositif informatique. Malgré les efforts considérables et les progrès impressionnants réalisés, la capacité des machines à reconnaître la parole est encore loin d'être comparable à celle des humains. En fait, l'analyse des signaux vocaux est très compliquée, car elle ne transmet pas seulement les informations de message linguistique d'un locuteur, mais également un ensemble d'informations sur ce locuteur.

Plusieurs raisons sont à l'origine de cette complexité, en particulier la redondance, la continuité et les effets de coarticulation, ainsi que l'ample variabilité intra et inter locuteurs. Toutes ces caractéristiques rendent très difficile la tâche d'un système RAP.

Nous avons brièvement parcouru dans ce chapitre une petite introduction à la reconnaissance automatique de la parole, suivi par l'historique de la RAP, en passant par les différents modules d'un système de la RAP, puis la performance d'un SRAP, les approches du traitement de signal de RAP globale et analytique, ensuite les applications, en citant : les systèmes de commande vocale, les systèmes de dictées.

On a montré aussi quelques difficultés dans ce domaine de recherche et pour finir nous avons cité quelques travaux de la RAP dans différent domaines parmi eux le domaine de robotique, des systèmes de commande vocale pour la maison, la voiture …

Pour notre part on va créer une base de données pour un système de télé-commandement vocale destiné aux sites web, on va parler en détails du protocole de création dans le chapitre suivant.

Chapitre 02 :
Approche proposée

1. Introduction

Les personnes aveugles et handicapées physiques éprouvent des difficultés et des inconvénients à utiliser l'ordinateur au moyen d'un clavier et/ou d'une souris. Notre objectif est de fournir un système de télé-commandement basé sur l'apprentissage approfondi pour les sites web, facile à utiliser pour cette catégorie de personnes, en utilisant seulement la parole en langue Arabe. Et pour cela nous avons opté à créer notre propre base de données contenant des mots clés contextuels.

La préparation des données permet une analyse efficace, Le terme « préparation des données » désigne les opérations de nettoyage et transformation qui doivent être appliqués aux données brutes avant leur traitement et analyse. Il s'agit d'une étape importante avant le traitement proprement dit. L'un des principaux objectifs de la préparation des données est de s'assurer que les informations préparées pour l'analyse sont exactes et cohérentes, afin que les résultats soient valides.

Dans ce chapitre on va présenter l'architecture générale de l'approche, puis on va entamé les étapes précédemment mentionnés dans l'architecture, le protocole suivi lors de l'enregistrement de nos propres audios, ensuite la préparation de données entre segmentation des audios et régularisation des échantillons, après on va montrer le processus d'extraction des caractéristiques et la constructions des vecteurs de caractéristiques destinées pour la phase de classification, et finalement nous résumons par une petite conclusion.

2. Architecture générale de l'approche proposée

Nous avons suivi une architecture pour la création de notre base de données

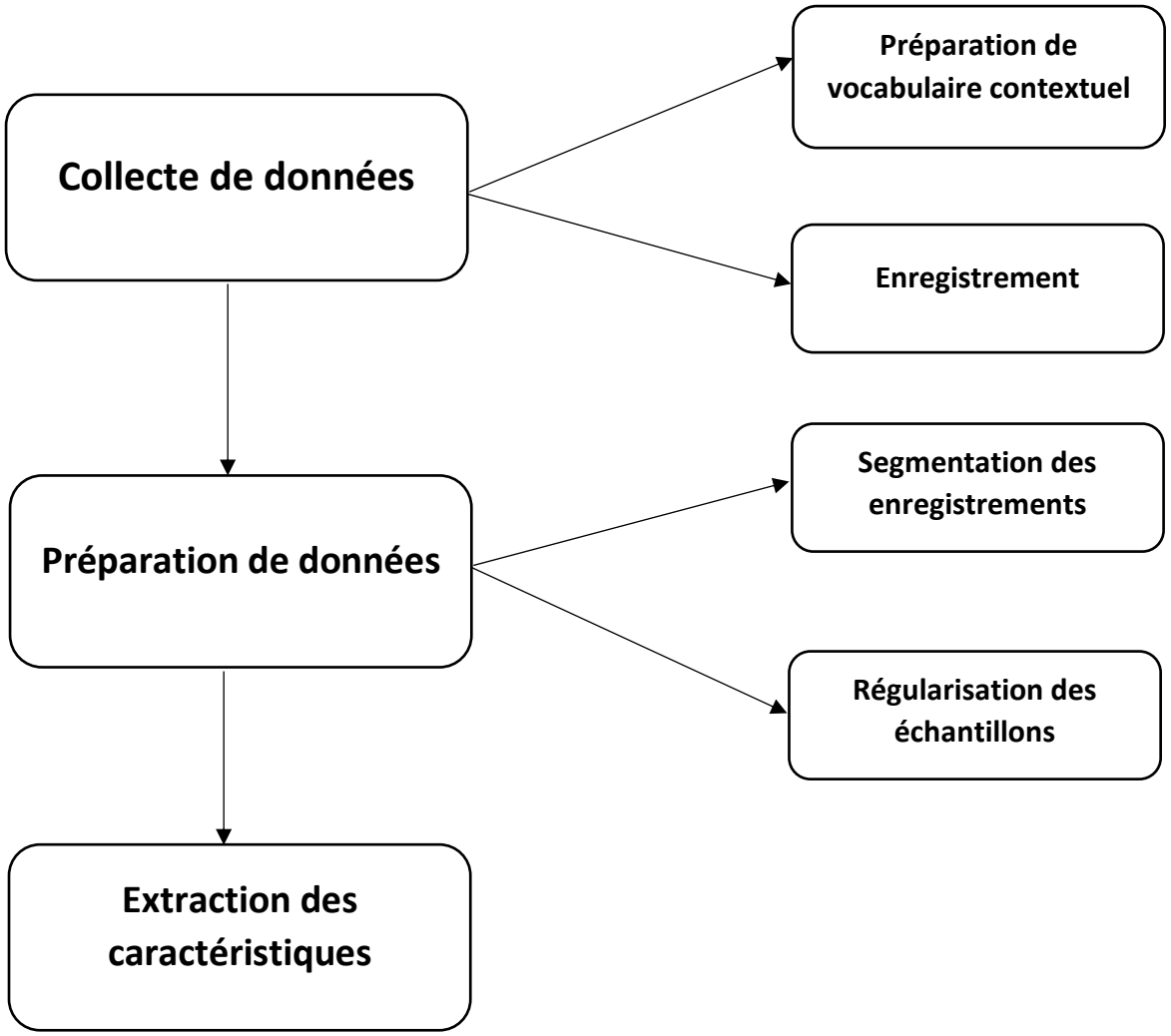

Figure2.1 : Architecture générale pour la création de base de données

3. Collecte des données

Avant d'entamé la préparation des données il fallait d'abord préparer le protocole d'enregistrement

3.1. Protocole de l'enregistrement

La préparation est l'étape la plus importante, et une bonne planification évitera beaucoup de problèmes à la fin.

- Ecrire les mots clés contextuels : le premier pas est de choisir et écrire les mots clés. L'écriture de la liste des mots à enregistrer aide à créer une présentation plus soignée et aide également à respecter les limites de temps,

on devait travailler sur une étude statistique plus élaborée pour choisir les mots, toutefois les contraintes du temps n'ont pas permis ceci.

- Préparer la liste des mots pour l'enregistrement : rendre la liste plus facile à lire, la police doit être suffisamment grande pour ne pas fatiguer les yeux, créer de larges marges.

- L'environnement d'enregistrement : Trouver un espace calme pour enregistrer, Prendre un moment pour écouter les sons ambiants dans la salle, on ne peut pas éliminer complètement le son ambiant, mais on peut le minimiser. Eteindre tous les téléphones et autres appareils pourraient perturber l'enregistrement. Fermer les portes et les fenêtres pour bloquer les sons de l'extérieur. Enregistrer à une heure calme de la journée.

3.2. Vocabulaire contextuel sélectionné

Dans cette étape, nous avons codé chaque mot du vocabulaire à une séquence d'unités sonores représentant la prononciation, il contient tous les mots avec plusieurs variantes possibles de leur prononciation, tenir compte de la variabilité de la prononciation, causée par diverses manières de parler et la spécificité de l'arabe.

Le choix des mots c'était à partir d'une recherche dans les sites web arabes, on a choisi quelques mots qui sont les plus répandus dans les menus.

Figure2.2 : Choix des mots

Le tableau 2.1 montre la liste des mots utilisés lors de la phase de l'apprentissage.

	Mots clés
01	القائمة
02	الرئيسية
03	الموقع
04	بحث
05	مساعدة
06	تنفيذ
07	إلغاء
08	رجوع
09	سابق
10	لاحق
11	يمين
12	يسار
13	أعلى
14	أسفل
15	نزول
16	صعود
17	دخول
18	خروج
19	فوق
20	تحت
21	الأول
22	الثاني
23	الثالث
24	الرابع
25	الخامس
26	السادس

27	السابع
28	الثامن
29	التاسع
30	الأخير

Tableau 2.1 : les mots clés utilisés pour l'apprentissage

3.3. L'enregistrement

Lors de l'enregistrement, il fallait suivre les étapes suivantes mentionnées :

- ✓ Assurer que le microphone capte le son.
- ✓ Enregistrer 1 second de silence entre chaque mot.
- ✓ Se tenir droit, aide à enregistrer beaucoup mieux.
- ✓ La distance entre la personne qui va enregistre et le microphone est à l'environ de 10 cm. [29]
- ✓ Toutes les personnes enregistrent 5 fois.
- ✓ L'enregistrement via l'application mobile « Enregistreur vocal » sur Google Play.

➢ Comment utiliser l'application :

- Nom de l'application :

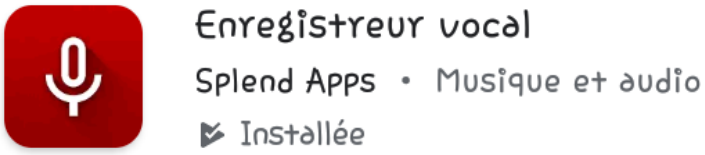

Figure 2.3 : Nom de l'application de l'enregistrement

- Paramétrage de l'application :

Le format d'enregistrement est : Waveform Audio File, ou simplement WAV Pourquoi ? Parce qu'il n'autorise aucune compression. Le rendu propose un son aussi clair et audible que la maquette originale, enregistrée au studio.

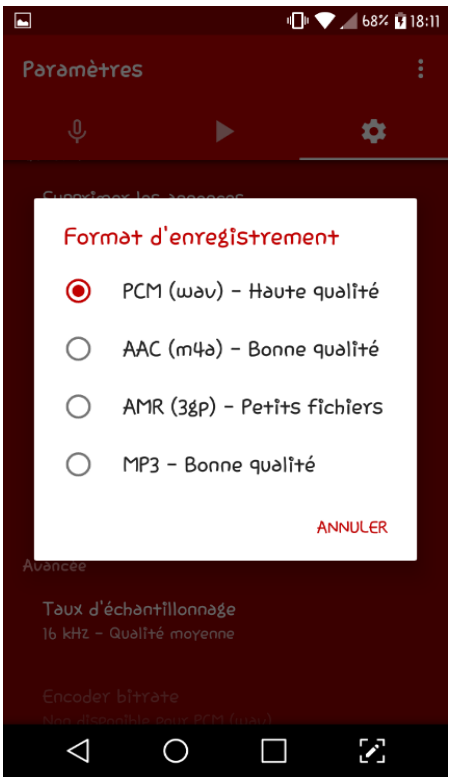

Figure2.4 : Format d'enregistrement

Le taux d'échantillonnage : Le taux d'échantillonnage peut varier entre 11kHz, 22kHz et 44kHz, avec un échantillonnage sur 8 ou 16 bit, et nous avons opté pour 44 kHz, pour se permettre une expérimentation sur les autres niveau d'échantillonnage, ce qui serait impossible de le faire si on a enregistrer depuis le début avec un taux de 8khz par exemple.

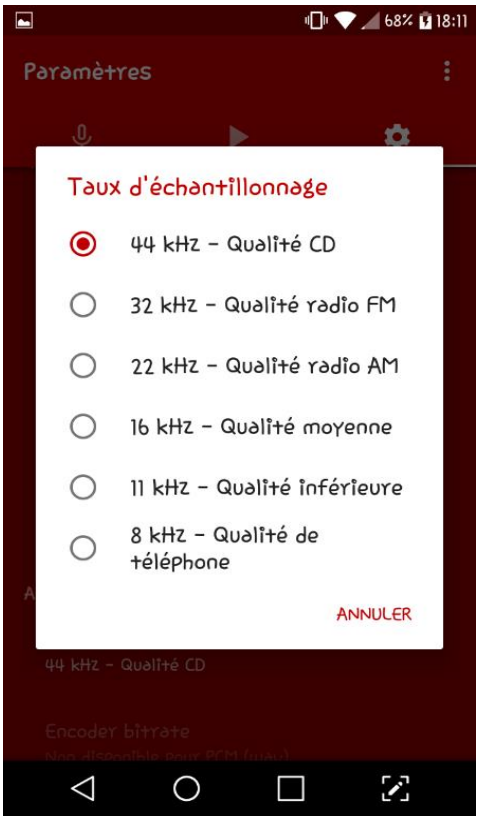

Figure2.5 : Taux d'échantillonnage

4. Préparation de données

Dans cette étape on va montrer la segmentation des audios et la régularisation des échantillons

4.1. Les outils de préparation des données

Python

Python est un langage de programmation qui nous a permis de travailler plus rapidement et d'intégrer nos systèmes plus efficacement. L'index de package Python (PyPI) héberge des milliers de modules tiers pour Python. La bibliothèque standard de Python et les modules apportés par la communauté offrent des possibilités infinies (développement Web et Internet, base de données, accès, interfaces graphiques de bureau, scientifique et numérique, éducation, réseau, programmation, développement de logiciels et de jeux) [20]

Spyder

Spyder est un environnement scientifique libre et open source écrit en Python, pour Python, et conçu par et pour les scientifiques, ingénieurs et analystes de données. Il offre une combinaison unique de la fonctionnalité avancée d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution interactive, l'inspection approfondie et de belles capacités de visualisation d'un progiciel scientifique. [21]

- ✓ **Pydub :** Pydub est une bibliothèque Python qui permet de manipuler des sons. Avec Pydub on va pouvoir manipuler des fichiers de son et pour pouvoir faire ça il faut d'abord les ouvrir avec python. Pour les ouvrir on utilise la fonction **AudioSegment.from_file("nom_de_ton_fichier_son",format="format")** le format est souvent mp3 mais il peut aussi être wav, mp4, aac, ogg,..

4.2. Segmentation des audios

Nous avons développé une fonction qui renvoie une liste des segments par la division d'audio sur les sections silencieuses en utilisant la méthode « split_on_silence » de la bibliothèque pydub, le paramétrage de la fonction split_on_silence, nécessite une recherche empirique, les deux paramètres min_silence_len et silence_thresh ils se diffèrent d'un audio à l'autre.

4.3. Protocole de nommage des échantillons

Après la segmentation on a passé à l'étape de nommage, nous avons suivi un protocole précis pour le nommage des échantillons, on a 30 mots nous avons intégré les labels de chaque mot sur le nom d'échantillon selon le tableau référence du vocabulaire, et cette étape nous a aidé à l'extraction des labels à partir du nom.

4.4. Régularisation des échantillons

Après la segmentation les échantillons ont été de taille différentes, et vu qu'on va aller vers l'extraction des caractéristiques par MFCC qui impose que toutes les échantillons ont la même taille, on a développé un code Python qui a pour but de régulariser les échantillons en utilisons une technique dont on cherche le plus grand échantillon, puis on

calcule la différence entre cet échantillon et les autres échantillons, et finalement on ajoute cette différence sous forme d'un entête et queue.

5. Extraction des caractéristiques

Le principal défi du système de reconnaissance automatique de la parole est la variabilité à grande échelle due aux différents types de locuteurs, à leur contenu vocal et à différentes conditions acoustiques. Les principaux composants de la reconnaissance du locuteur et de la reconnaissance vocale automatique sont l'extraction et l'analyse de caractéristiques. Dans un système RAP, Les composants d'analyse des caractéristiques jouent un rôle important dans la performance globale du système. Par conséquent, pour l'analyse des caractéristiques et la reconnaissance vocale ultérieure, la première étape est l'extraction des caractéristiques.

5.1. Caractéristiques utilisées pour la RAP

- LPC « Linear prediction coefficients »
- LPCC « Linear prediciton cepstral coefficients»
- LSF « Line spectral frequencies »
- DWT « Discrecte wavelet transform »
- PLP « Perceptual linear prediction »
- MFCC « Mel frequency cepstral coefficients »

On a choisi le MFCC pour l'extraction des caractéristiques

5.2. Mel-Frequency Cepstral Coefficient (MFCC)

MFCC sont les caractéristiques les plus couramment utilisées pour ASR. Leur succès dépend de leur capacité à effectuer des types similaires de filtrage qui sont corrélés au système auditif humain et à leur faible dimensionnalité. Le calcul des caractéristiques MFCC comporte sept étapes, le processus global est illustré dans la figure 2.5 [23]

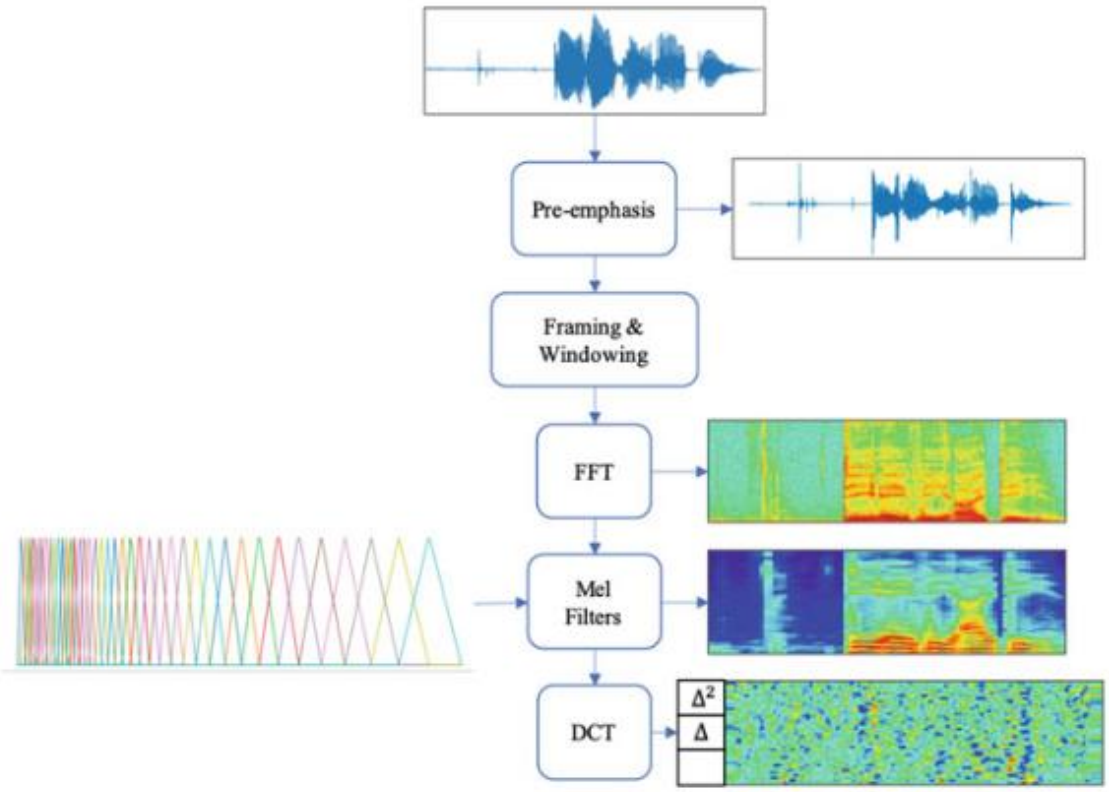

Figure 2.6 : Processus d'extraction des caractéristiques MFCC

MFCC est le coefficient ceptral calculé par transformée en cosinus discrète appliquée au spectre de puissance du signal. Les bandes de fréquences de ce spectre sont espacées de manière logarithmique sur l'échelle de Mel.

5.2.1. Calcul de MFCC

- ✓ Calculer la transformée de Fourier de la trame à analyser
- ✓ Pondérer le spectre d'amplitude (ou de puissance, selon la situation) à partir d'un ensemble de filtres triangulaires avec des intervalles d'échelle Mel.
- ✓ Calculer la transformée en cosinus discrète log-mel-spectre.

Le coefficient résultant de ce DCT est MFCC.

5.2.2. Les paramètres de MFCC

- ✓ Signal : l'audio dont on va travailler sur.
- ✓ N_mfcc : le nombre de MFCC à retourner, le nombre le plus utilisé est 13.
- ✓ Sr : taux d'échantillonnage.
- ✓ Hop_length : Le nombre d'échantillons entre les trames successives.

5.3. Extraction des caractéristiques et Labels

En utilisant la bibliothèque Librosa (une bibliothèque Python pour le traitement de la musique et l'analyse des audios), nous avons développé un code qui a pour objectif d'extraire la matrice des MFCCs résultantes des échantillons, après redimensionnement et les enregistre dans un fichier .csv, et une partie de ce code c'était pour extraire les labels à partir des noms des audios puis on les enregistre dans un fichier .csv.

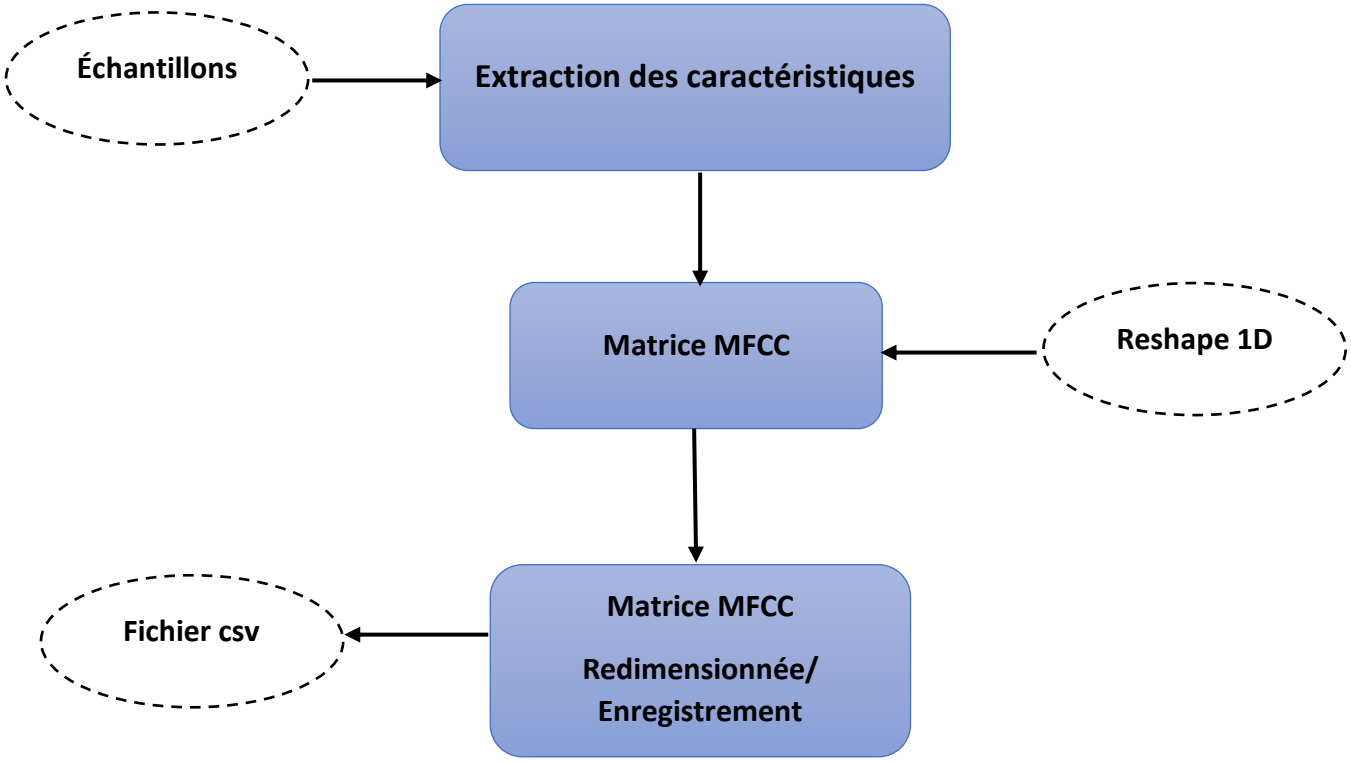

Figure 2.7 : Processus d'extraction des caractéristiques

6. Conclusion

La préparation des données est le processus de collecte, de structuration et d'organisation des données afin qu'elles puissent être analysées, Nous avons brièvement parcouru dans ce chapitre une petite présentation de notre approche , suivi par son architecture, passant par le collecte de données où on a mentionnée le protocole d'enregistrement, suivi par la préparation de données entre la segmentation des audios et la régularisation des échantillons, et pour finir nous avons montré l'étape de l'extraction des caractéristiques en utilisant MFCC .

Dans le chapitre suivant on va parler en détails de l'étape suivant qui est le travail expérimental et les futures perspectives.

Chapitre 03 :

Implémentation et expérimentation

1. Introduction

Dans ce chapitre on mettra en avant notre architecture et on va montrer comment réaliser notre système de télé-commandement vocal pour un site web en utilisant seulement la parole en langue Arabe. Ainsi on va présenter notre modèle de réseau de neurone convolutif, puis on l'appliquera sur notre base de données. Pour cela on va travailler avec le langage de programmation python et des bibliothèques comme Tensorflow et Keras pour l'apprentissage et pour améliorer les performances de modèle.

2. Architecture du système de télé-commandement vocal

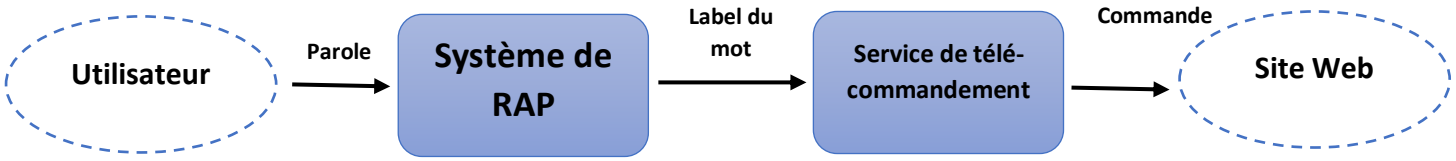

Figure 3.1 : Architecture de système de télé-commandement vocal

L'étape suivante pour l'approche proposé est la construction du modèle de donnée à base de Deep Learning.

3. Deep Learning

L'apprentissage profond est un ensemble d'algorithmes d'apprentissage automatique qui tentent d'apprendre à plusieurs niveaux, correspondant à différents niveaux d'abstraction. Il a la capacité d'extraire des caractéristiques des données d'origine grâce à un traitement multicouche consistant en plusieurs transformations linéaires et non linéaires, et de comprendre ces caractéristiques couche par couche avec une intervention manuelle minimale. [27]

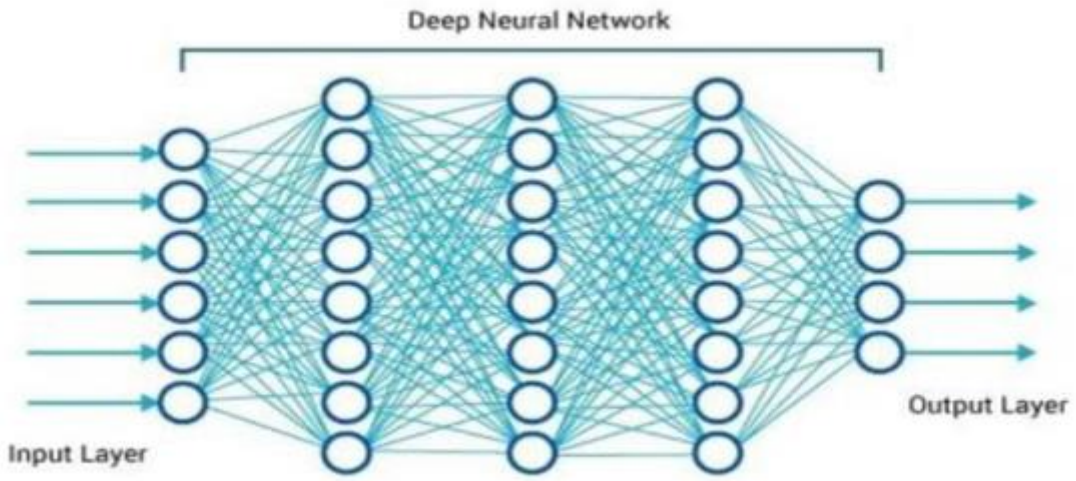

Figure 3.2 : Schéma illustratif de DL avec plusieurs couches [27]

4. Les réseaux neurones convolutionnels

Les réseaux de neurones convolutifs, sont le choix populaire des réseaux de neurones pour différentes tâches de vision par ordinateur telles que la reconnaissance d'image et la reconnaissance de la parole. Le nom « convolution » est dérivé d'une opération mathématique impliquant la convolution de différentes fonctions. La conception d'un CNN comporte 4 étapes principales :

- **Convolution :** le signal d'entrée est reçu à ce stade.
- **Sous-échantillonnage :** les entrées reçues de la couche de convolution sont lissées pour réduire la sensibilité des filtres au bruit ou à toute autre variation.
- **Activation :** cette couche contrôle la façon dont le signal circule d'une couche à l'autre, semblable aux neurones de notre cerveau.
- **Entièrement connecté :** dans cette étape, toutes les couches du réseau sont connectées avec chaque neurone d'une couche précédente aux neurones de la couche suivante Voici un aperçu approfondi de l'architecture CNN et de son fonctionnement, comme l'explique le célèbre chercheur en IA Giancarlo Zaccone. [28]

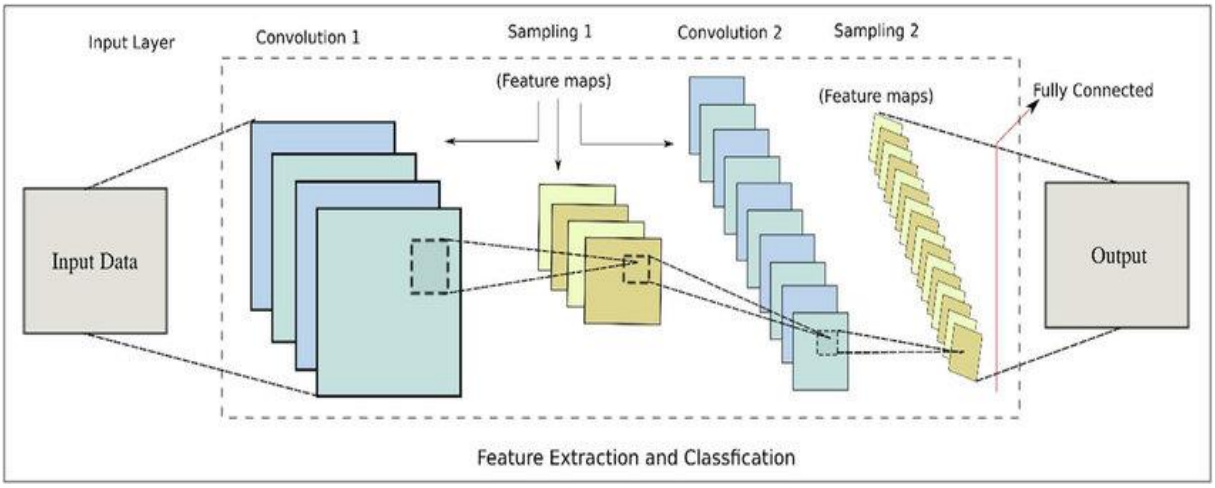

Figure 3.3 : Architecture de réseau neuronal convolutif

5. Logiciels et librairies utilisés

5.1. Python

Python est un excellent langage de programmation orienté objet, interprété et interactif. Il est souvent comparé (favorablement bien sûr) à Lisp, Tcl, Perl, Ruby, C #, Visual Basic, Visual Fox Pro, Scheme ou Java ... etc.

Python combine un pouvoir remarquable avec une syntaxe très claire. Il comporte des modules, des classes, des exceptions, des types de données dynamiques de très haut niveau et le typage dynamique. Il existe des interfaces vers de nombreux appels systèmes et bibliothèques, ainsi que vers différents systèmes de fenêtrage.

Les nouveaux modules intégrés sont faciles à écrire en C ou C ++ (ou dans d'autres langages, selon l'implémentation choisie). Python est également utilisable comme langage d'extension pour les applications écrites dans d'autres langages nécessitant des interfaces de script ou d'automatisation facile à utiliser. [20]

5.2. Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning et Deep Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. [24]

5.3. Tensorflow

TensorFlow est une bibliothèque logicielle open source pour le calcul numérique de haute performance. Son architecture flexible permet un déploiement facile du calcul sur diverses plates-formes (CPUs, GPUs, TPUs), et des ordinateurs de bureau aux clusters de serveurs, aux périphériques mobiles. Initialement développé par des chercheurs et des ingénieurs de l'équipe de Google Brain au sein de l'organisation de l'IA de Google, il s'appuie sur l'apprentissage automatique et l'apprentissage en profondeur. [25]

5.4. Keras

Keras est une API de réseaux neuronaux de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow, CNTK ou Theano. Il a été développé dans le but de permettre une expérimentation rapide. Être en mesure de passer de l'idée au résultat le plus rapidement possible, est la clé pour faire de la recherche :

- Permet un prototypage facile et rapide (grâce à la convivialité, à la modularité et à l'extensibilité).
- Prend en charge les réseaux convolutionnels et les réseaux récurrents ainsi que les combinaisons des deux.
- Fonctionne de manière transparente sur le processeur et le processeur graphique. [26]

6. Présentation de notre data-set

Nous avons créé une base de données pour notre système de télé-commandement vocal pour un site web en utilisant seulement la parole en langue Arabe. Notre base de données contient 4080 audios, partitionner en 30 classes (136 audios par classe) enregistré par 34 locuteurs femmes et hommes de différents âge, dans le futur on va travailler à enrichir et agrandir notre base de données.

7. Structure de réseau neuronal convolutif

Après avoir accédé à la base de données sur google drive, on expliquera l'apprentissage avec la méthode de réseau neuronal Convolution après avoir importé Tensorflow. Dans ce CNN nous avons deux couches de convolution, deux couches de maxpooling et deux couches fully connected. Cette structure était le résultat d'un ensemble d'essais empiriques, et qui peut être améliorée avec plus d'expérimentation.

La couche en entré à une taille de (13X173X1), chaque couche de convolution composée de plusieurs filtres 64 pour la première couche, la taille de chaque filtre est de 3*3, dans la deuxième couche on change quelques paramètres comme le nombre de filtres qui devient 32 filtres de taille 3*3 a la place de 64, la fonction d'activation ''Relu'' est utilisée à chaque fois qu'on passe par une couche de convolution, cette fonction d'activation force les neurones à retourner des valeurs positives.

Après chaque couche de convolution on a appliqué une couche de maxpooling, la taille de ses filtres suit celle de la couche de convolution qui lui précède.

Avant chaque couche de maxpooling, nous avons aussi une couche de normalisation.

Après les couches de convolution et de maxpooling, notre partie classification se compose de deux couches entièrement connectées. Cependant, ces couches ne peuvent accepter que des données à une dimension. Pour convertir nos données 3D en 1D, nous utiliserons la fonction Flatten, cela permettra essentiellement d'arranger notre volume 3D en un vecteur 1D.

Les dernières couches d'un réseau neurones convolutionnel sont des couches entièrement connectées. Ces dernières ont des connexions complètes avec toutes les activations de la couche précédente.

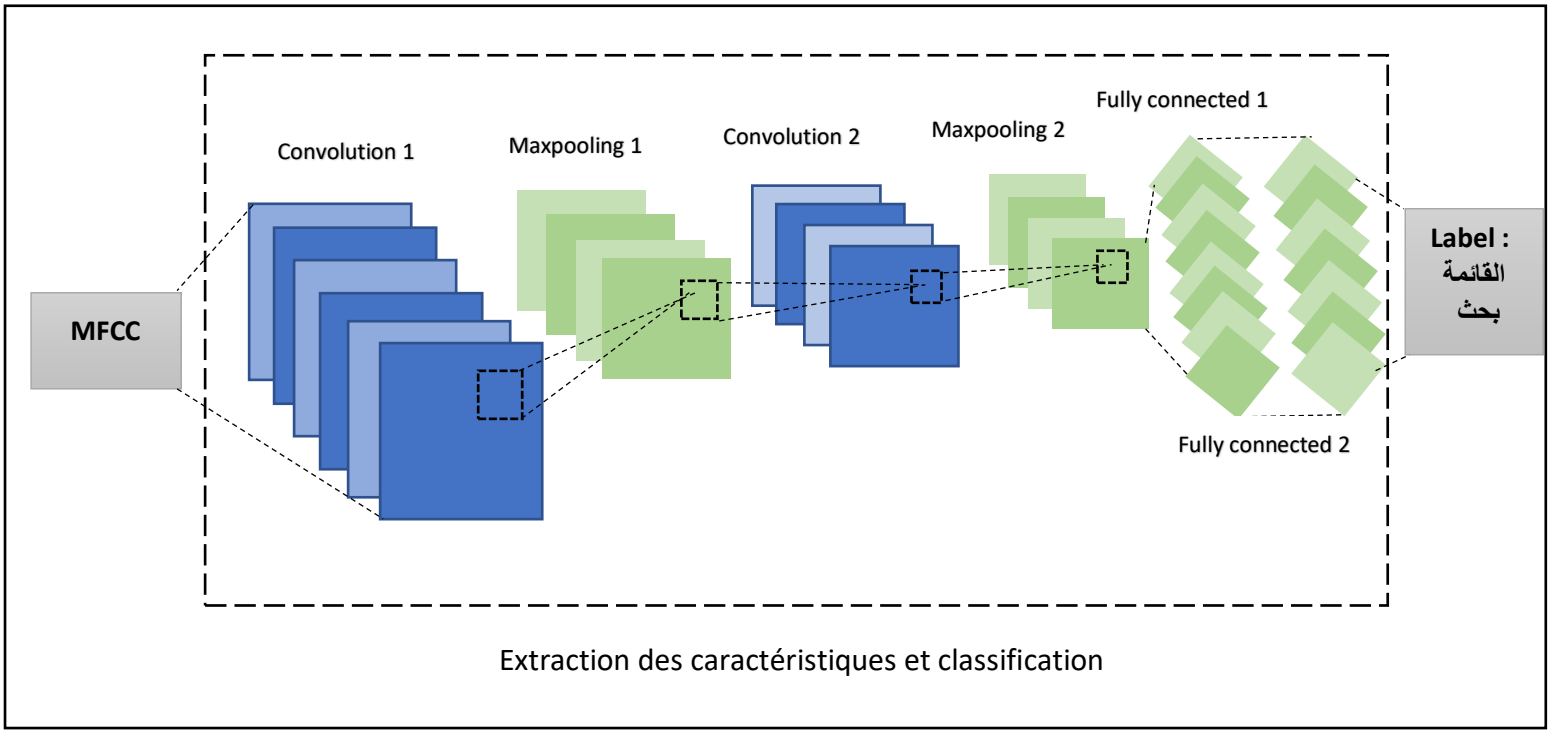

Figure 3.4 : Structure de réseau neuronal convolutif

8. Résultats Expérimentaux et discussions

8.1. Protocole d'expérimentation

Le protocole vise plusieurs dimensions, mais dans notre cas on va travailler seulement sur :

1. Le nombre de mfcc (il y'a beaucoup de configurations mfcc qu'on peut mettre en œuvre tels que : le nombre de mfcc, le hop_length, etc.)
2. Le taux d'échantillonnage.

8.2. Expérimentations et discussions

8.2.1. Première expérimentation (nombre de mfcc)

Dans la première expérimentation on va travailler sur le nombre de caractéristiques MFCC.

8.2.1.1. Fonction Train_test_split

Train_test_split est une fonction utilisée pour estimer la performance des algorithmes d'apprentissage approfondi lorsqu'ils sont utilisés pour faire des prédictions sur des données non utilisées pour former le modèle.

- **MFCC 13**

Dans un premier temps, nous avons mis en œuvre le réseau neuronal convolutif avec une base de données de taux d'échantillonnage 44100Hz et le nombre de MFCC est 13.

Paramètres fixe :

- Batch_Size : 32
- Epoques : 40
- Learning_rate : 0,001
- Taux d'échantillonnage : 44100 Hz

On a obtenu un taux de précision de 91,17%, avec un taux d'erreur de 0,38%.

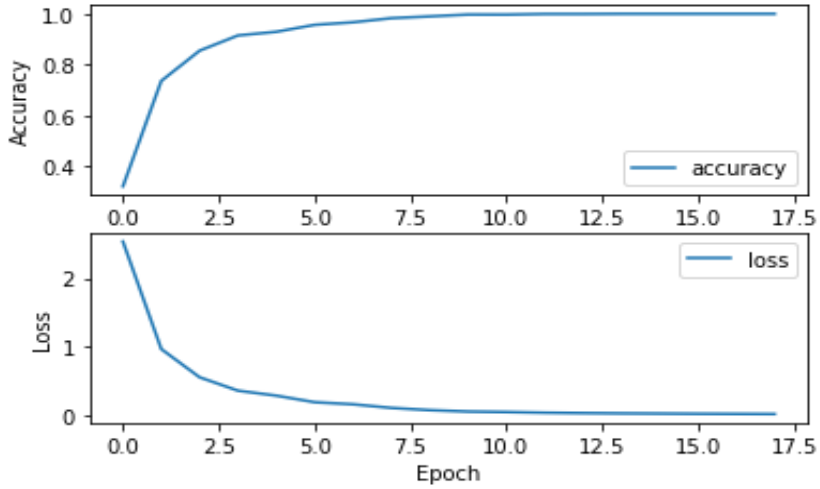

Figure 3.5 : Taux de précision train-test-split 13/44100

- **MFCC 16**

Ensuite on a testé sur le nombre de caractéristiques MFCC 16, et on a atteint taux de précision de 91,05%, et 0,41% taux d'erreur.

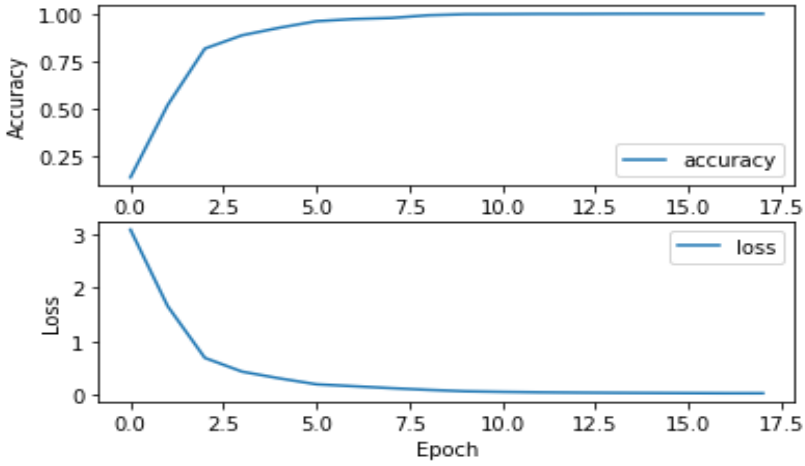

Figure 3.6 : Taux de précision train-test-split 16/44100

- **MFCC 20**

Pour le nombre de MFCC 20 on a obtenu un taux de précision de 91,05 %, et un taux d'erreur 0,42%.

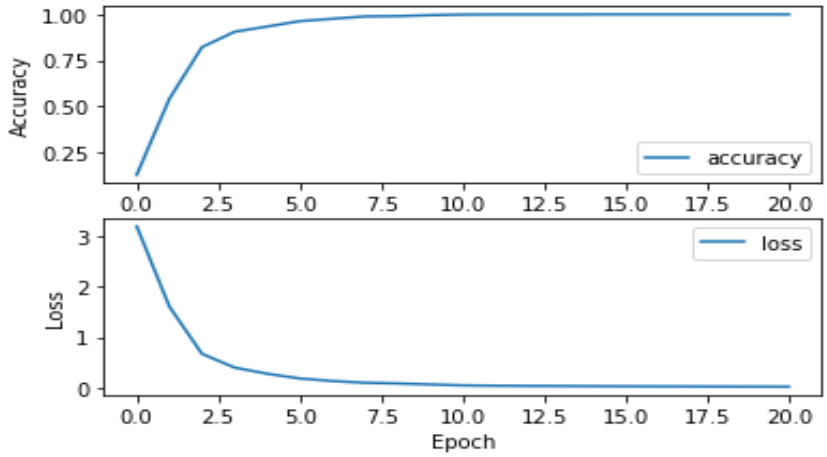

Figure 3.7 : Taux de précision train-test-split 20/44100

- **MFCC 22**

Et pour finir avec la fonction Train_test_split on a testé le modèle avec un nombre de caractéristique mfcc 22, et nous avons atteint 91,05 comme un taux de précision et 0,49% taux d'erreur.

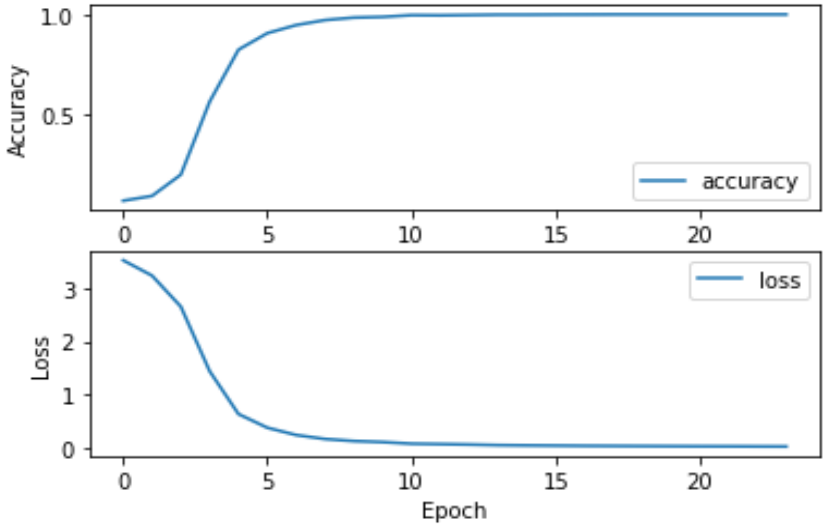

Figure 3.8 : Taux de précision train-test-split 22/44100

D'après les résultats obtenus on trouve que le nombre de mfcc n'influence pas sur le taux de précision du modèle.

8.2.1.2. Validation croisée

On va appliquer la validation croisée qui est une méthode en Deep Learning pour l'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage. On parle en générale de validation croisée à K blocs (ou K-fold cross validation) pour désigner une technique d'évaluation d'un algorithme de Deep Learning. Cela consiste à découper le dataset en K sous-ensemble (ou K folds) puis prendre un des K sous-ensemble comme dataset de validation (validation set) et les K-1 restants comme dataset d'entrainement (training set). On répète l'opération sur toutes les combinaisons possibles. On obtient K mesures de performance dont la moyenne représente la performance de l'algorithme et ceci va permettre à tous les échantillons de participer à l'apprentissage.

- **MFCC 13**

Dans un premier temps, nous avons mis en œuvre le réseau neuronal convolutif avec une base de données de taux d'échantillonnage 44100Hz et le nombre de MFCC est 13, avec K-fold (5/10/15/20).

K-Fold	Précision	Erreur
5	92,13%	0,51
10	88,22%	0,40
15	90,91%	0,27
20	91,27%	0,53

Tableau 3.1 : Résultats obtenus mfcc13/44100

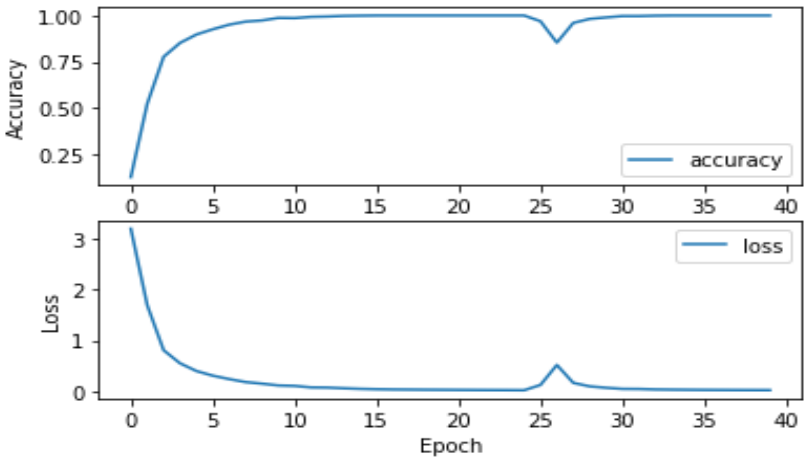

Figure 3.9 : précision et erreur mfcc13/44100

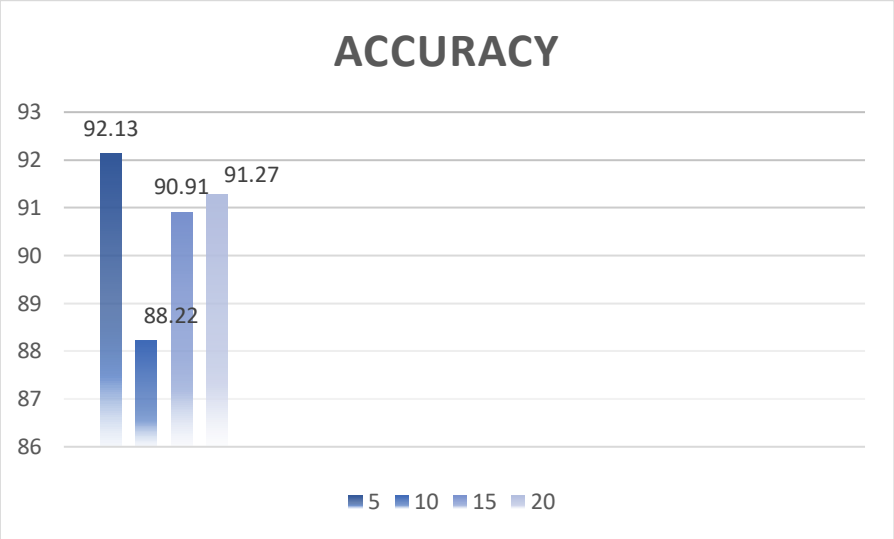

Figure 3.10 : Histogramme de précision mfcc13/44100

L'histogramme ci-dessus exprime les résultats de précision présenté dans le tableau.

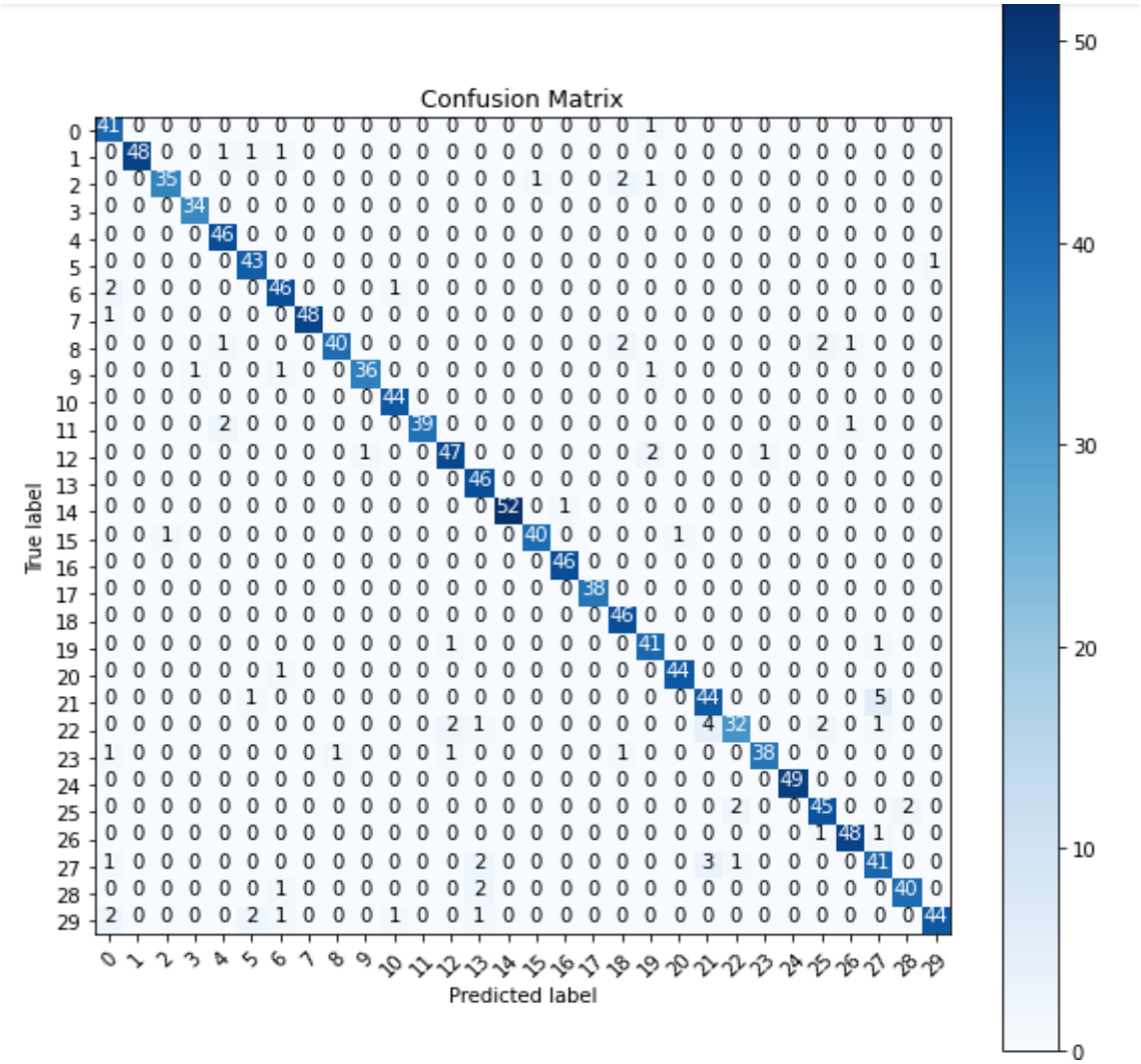

Figure 3.11 : Matrice de confusion mfcc13/44100

La matrice de confusion permet d'évaluer la performance de notre modèle, puisqu'elle reflète les métriques du Vrai positif, Vrai négatif, Faux positif et Faux négatif. La figure 3.11 illustre de près la position de ces métriques pour chaque classe. A titre d'exemple le modèle a bien classé la majorité des échantillons.

a. MFCC 16 :

Ensuite nous continuerons le même travail avec les mêmes paramètres fixe mais cette fois on va augmenter le nombre de MFCC a 16.

K-Fold	Précision	Erreur
5	92,23%	0,36
10	92,20%	0,35
15	92,35%	0,38
20	90,15%	0,36

Tableau 3.2 : Résultats obtenus mfcc16/44100

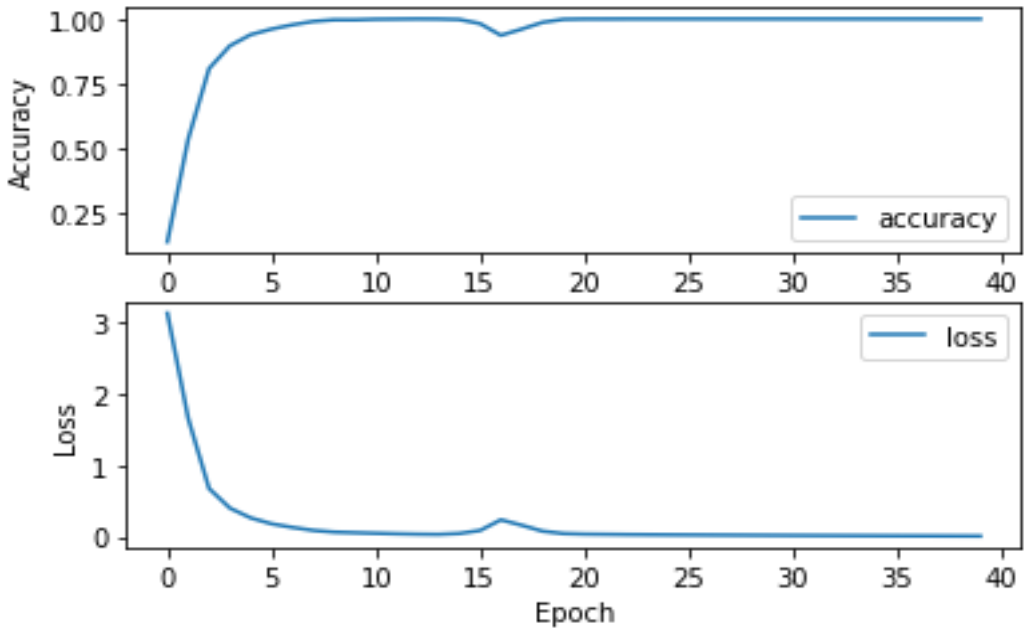

Figure 3.12 : précision et erreur mfcc16/44100

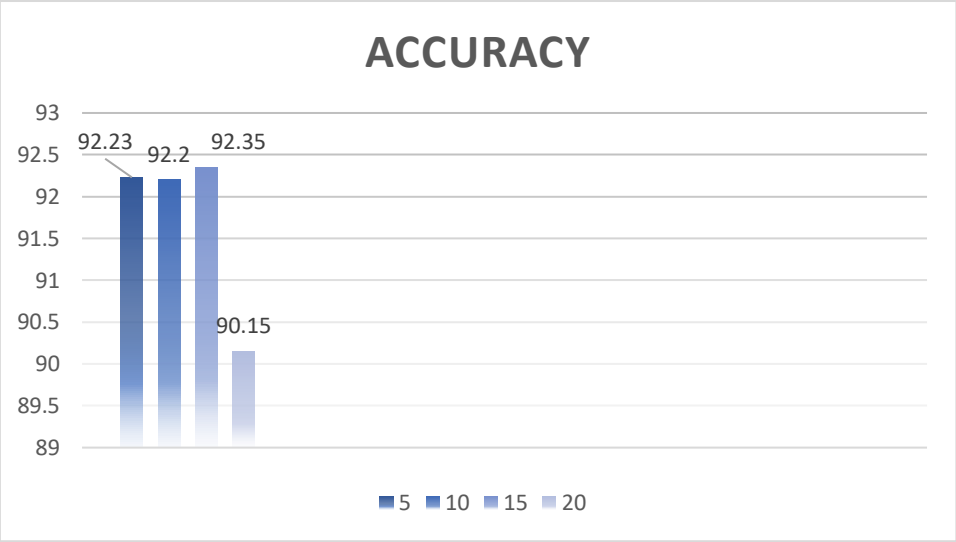

Figure 3.13 : Histogramme de précision mfcc16/44100

b. MFCC 20 :

Maintenant on va tester avec le nombre 20 et on verra ce que ça donne.

K-Fold	Précision	Erreur
5	90,41%	0,47
10	88,92%	0,48
15	90,83%	0,61
20	89,92%	0,54

Tableau 3.3 : Résultats obtenus mfcc20/44100

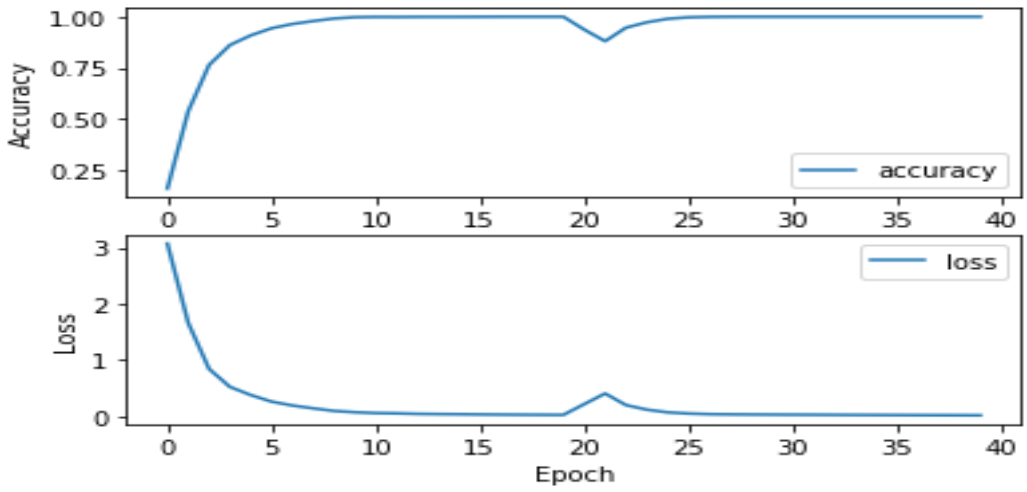

Figure 3.14 : précision et erreur mfcc20/44100

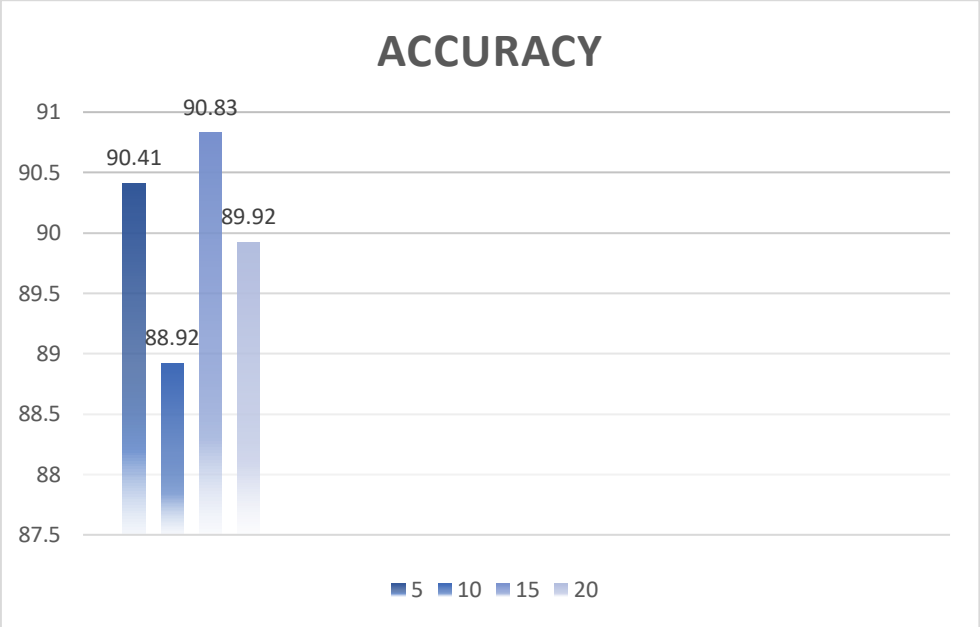

Figure 3.15 : Histogramme de précision mfcc20/44100

L'histogramme ci-dessus exprime les résultats de précision présenté dans le tableau.

c. MFCC 22 :

Et pour finir avec le taux d'échantillonnage 44100 Hz on va essayer les caractéristiques MFCC avec le nombre 22.

K-Fold	Précision	Erreur
5	87,96%	0,55
10	90,44%	0,58
15	88,11%	0,40
20	89,97%	0,57

Tableau 3.4 : Résultats obtenus mfcc22/44100

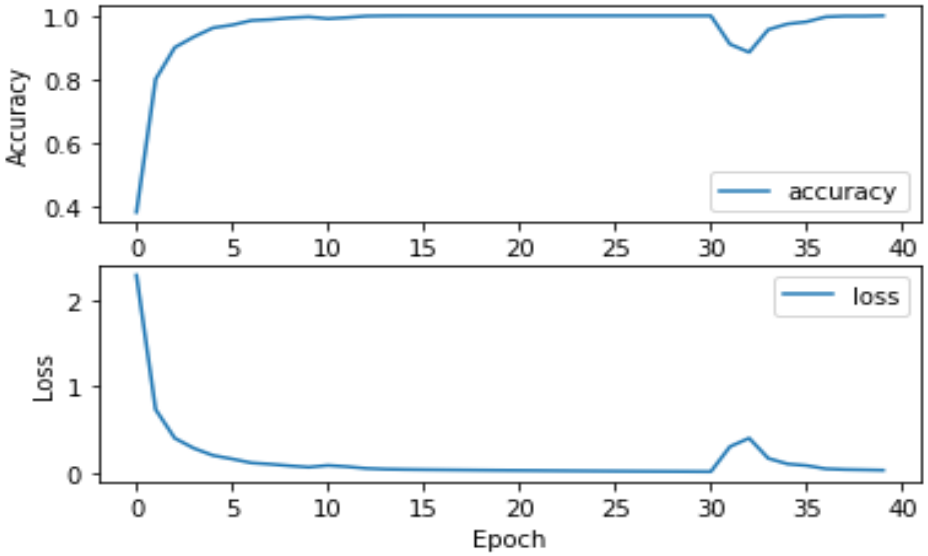

Figure 3.16 : précision et erreur mfcc22/44100

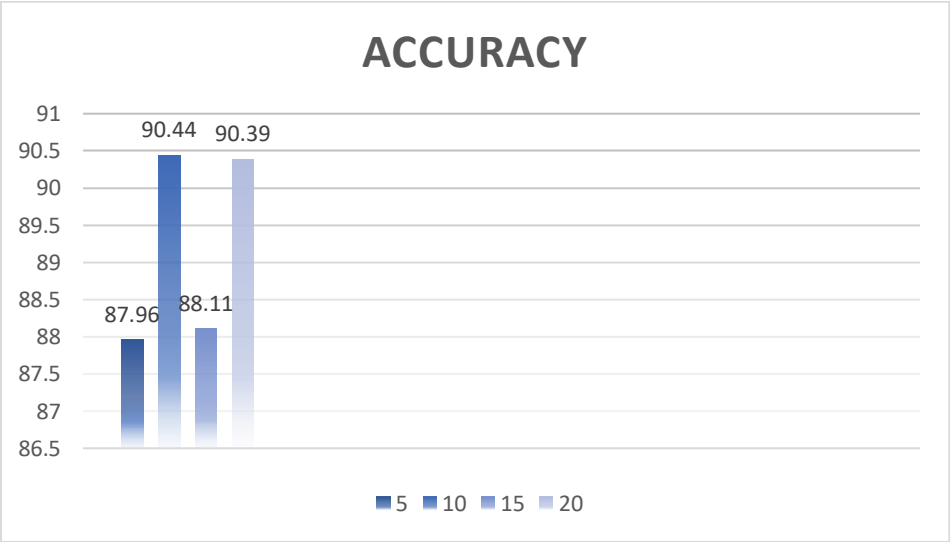

Figure 3.17 : Histogramme de précision 22/44100

L'histogramme ci-dessus exprime les résultats de précision présenté dans le tableau.

D'après les tableaux 1, 2, 3 et 4 on trouve que les résultats obtenus varient dans le même intervalle, ce qui prouve que l'augmentation de nombre des caractéristiques MFCC n'influence pas au taux de précision de notre système, alors nous n'avons pas besoin d'augmenté le nombre de MFCC puisque cela rendre la taille des fichiers plus grande.

8.2.1.3. Dataset équilibré

Tous les classes ont la même chance d'être entrainé et testé par le même nombre des échantillons.

- **MFCC 13**

On avait mis en œuvre le réseau neuronal convolutif avec une base de données équilibré de taux d'échantillonnage 44100Hz et le nombre de MFCC est 13. Et pour cela on a atteint un taux de précision de 93,21% et un taux d'erreur de 0,39%.

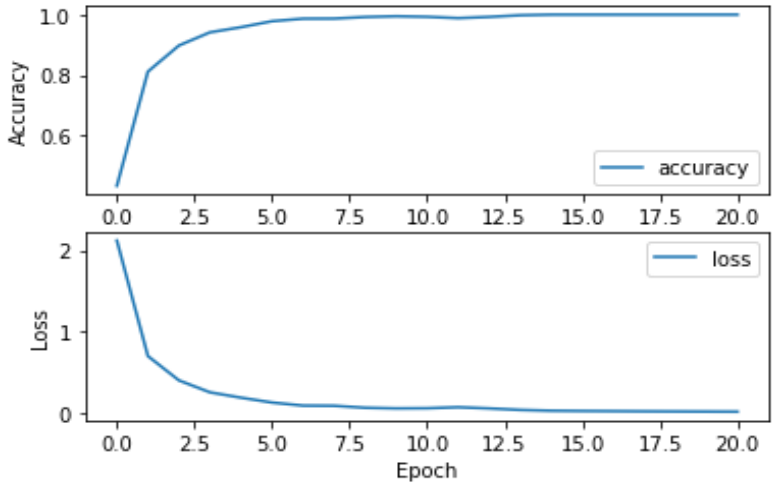

Figure 3.18 : précision et erreur dataset équilibré 44100

Dans la prochaine expérimentation on va fixer le nombre de MFCC a 13 et on joue sur les taux d'échantillonnage et on va voir ce que ça donne.

8.2.2. Deuxième expérimentation (taux d'échantillonnage)

Après les résultats obtenus par les différents nombres de caractéristiques MFCC, on a décidé de fixer le nombre a 13 et réduire les taux d'échantillonnage.

8.2.2.1. Fonction Train_test_split

- **Taux d'échantillonnage 22050**

On va commencer à appliquer le réseau neuronal convolutif sur une base de données de taux d'échantillonnage 22050 Hz.

Paramètres fixe :

- Batch_Size : 32
- Epoques : 40
- Learning_rate : 0,001
- MFCC : 13 / Hop_length = 512

On a obtenu un taux de précision 92,27%, et un taux d'erreur 0,40%.

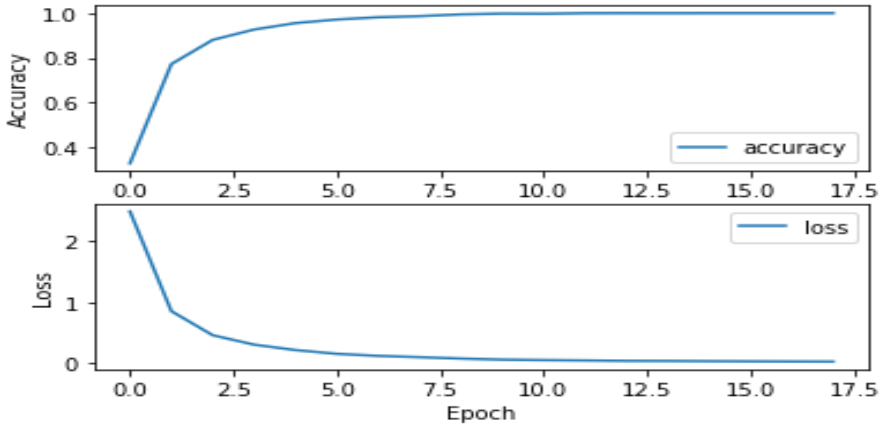

Figure 3.19 : train_test_split22050/13

- **Taux d'échantillonnage 11025**

On avait mis en œuvre le réseau neuronal convolutif avec une base de données de taux d'échantillonnage 11025Hz et le nombre de MFCC est 13. Et pour cela on a atteint un taux de précision de 91,91% et un taux d'erreur de 0,37%.

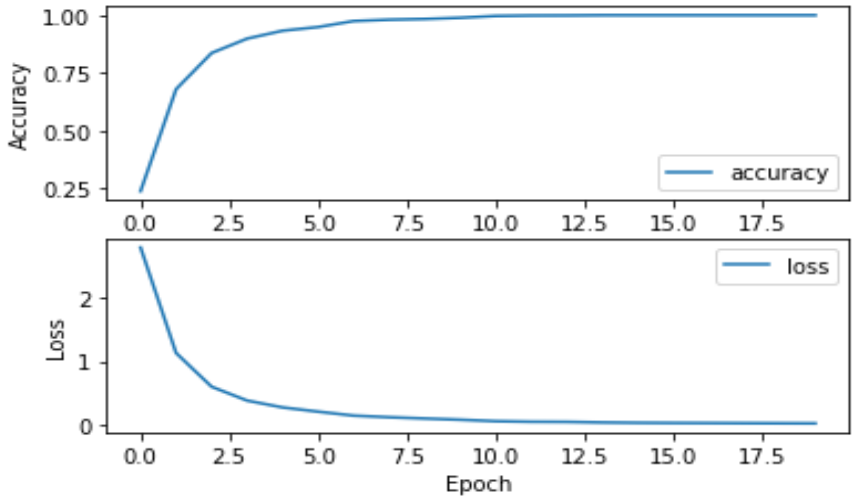

Figure 3.20 : train_test_split11025/13

- **Taux d'échantillonnage 8000**

Nous avons mis en œuvre le réseau neuronal convolutif avec une base de données de taux d'échantillonnage 8000Hz et le nombre de MFCC 13. On a atteint un taux de précision de 90,56% et un taux d'erreur de 0,47%.

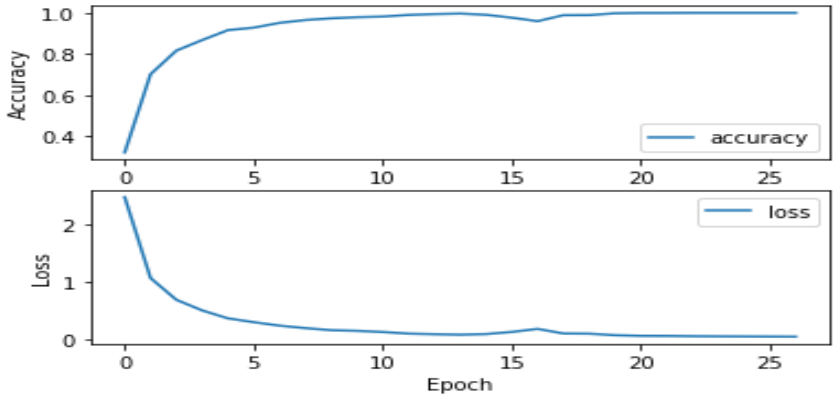

Figure 3.21 : train_test_split8000/13

8.2.2.2. Validation croisée

- **Taux d'échantillonnage 22050**

Dans un premier temps, nous avons mis en œuvre le réseau neuronal convolutif avec une base de données de taux d'échantillonnage 22050Hz et le nombre de MFCC est 13, avec K-fold (5/10/15/20).

K-Fold	Précision	Erreur
5	93,63%	0,38
10	93,90%	0,33
15	93,43%	0,32
20	94,26%	0,24

Tableau 3.5 : Résultats obtenus mfcc13/22050

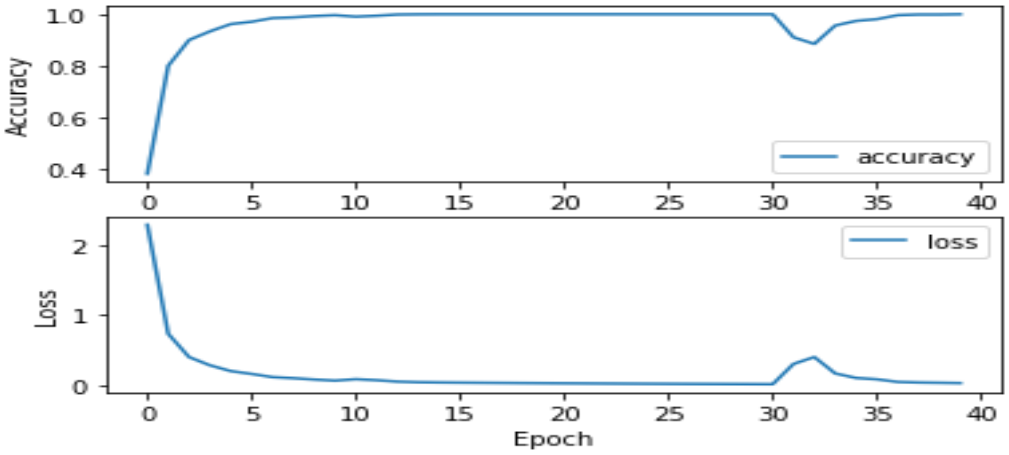

Figure 3.22 : précision et erreur mfcc13/22050

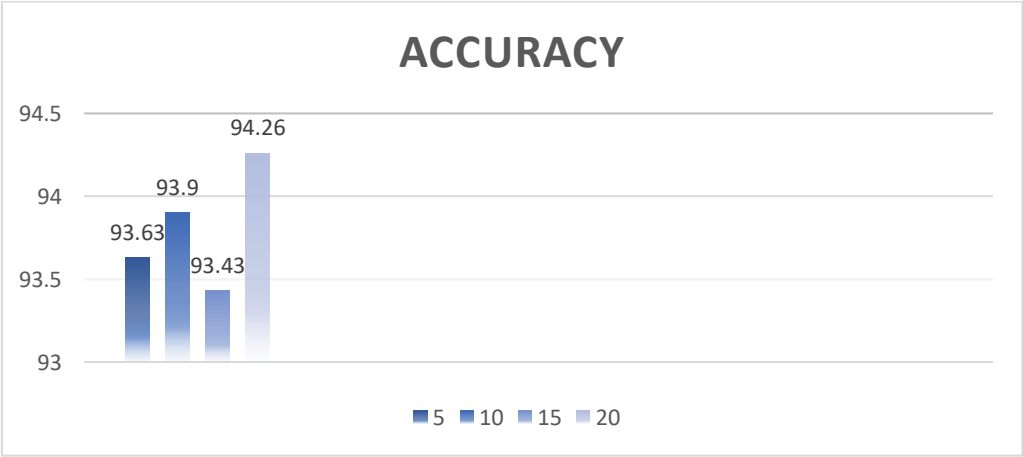

Figure 3.23 : Histogramme de précision mfcc13/22050

L'histogramme ci-dessus exprime les résultats de précision présenté dans le tableau.

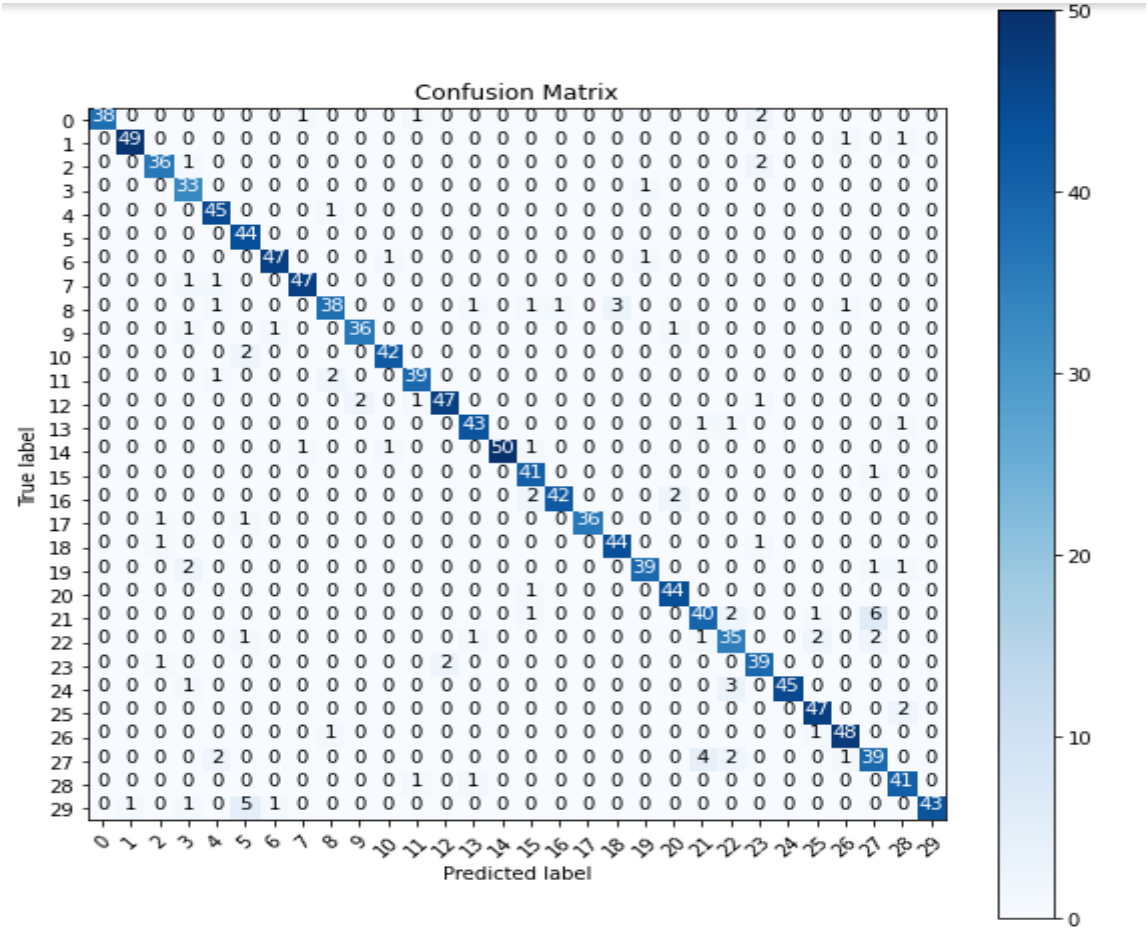

Figure 3.24 : Matrice de confusion mfcc13/22050

D'après la figure 3.24 on trouve que pareil pour le taux d'échantillonnage 22050 Hz le modèle a bien classé la majorité des échantillons.

- **Taux d'échantillonnage 11025 :**

Ensuite nous continuerons le même travail avec les mêmes paramètres fixe mais cette fois par un taux d'échantillonnage 11025 Hz.

K-Fold	Précision	Erreur
5	91,00%	0,35
10	92,18%	0,25
15	91,79%	0,20
20	92,96%	0,24

Tableau 3.6 : Résultats obtenus mfcc13/11025

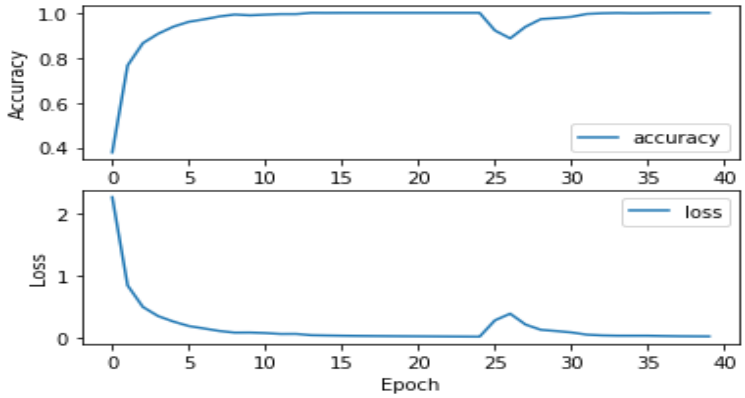

Figure 3.25 : précision et erreur mfcc13/11025

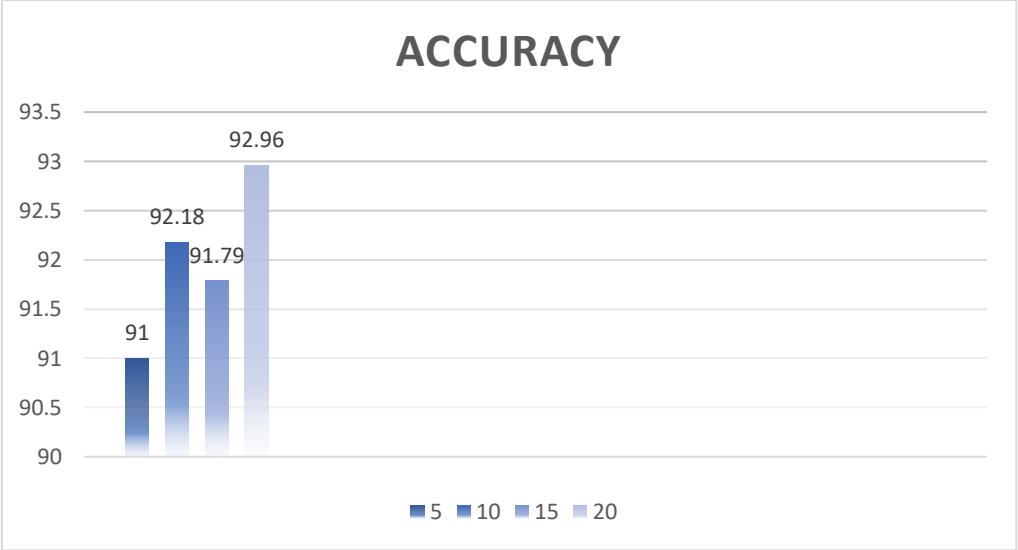

Figure 3.26 : Histogramme de précision mfcc13/11025

L'histogramme ci-dessus exprime les résultats de précision présenté dans le tableau.

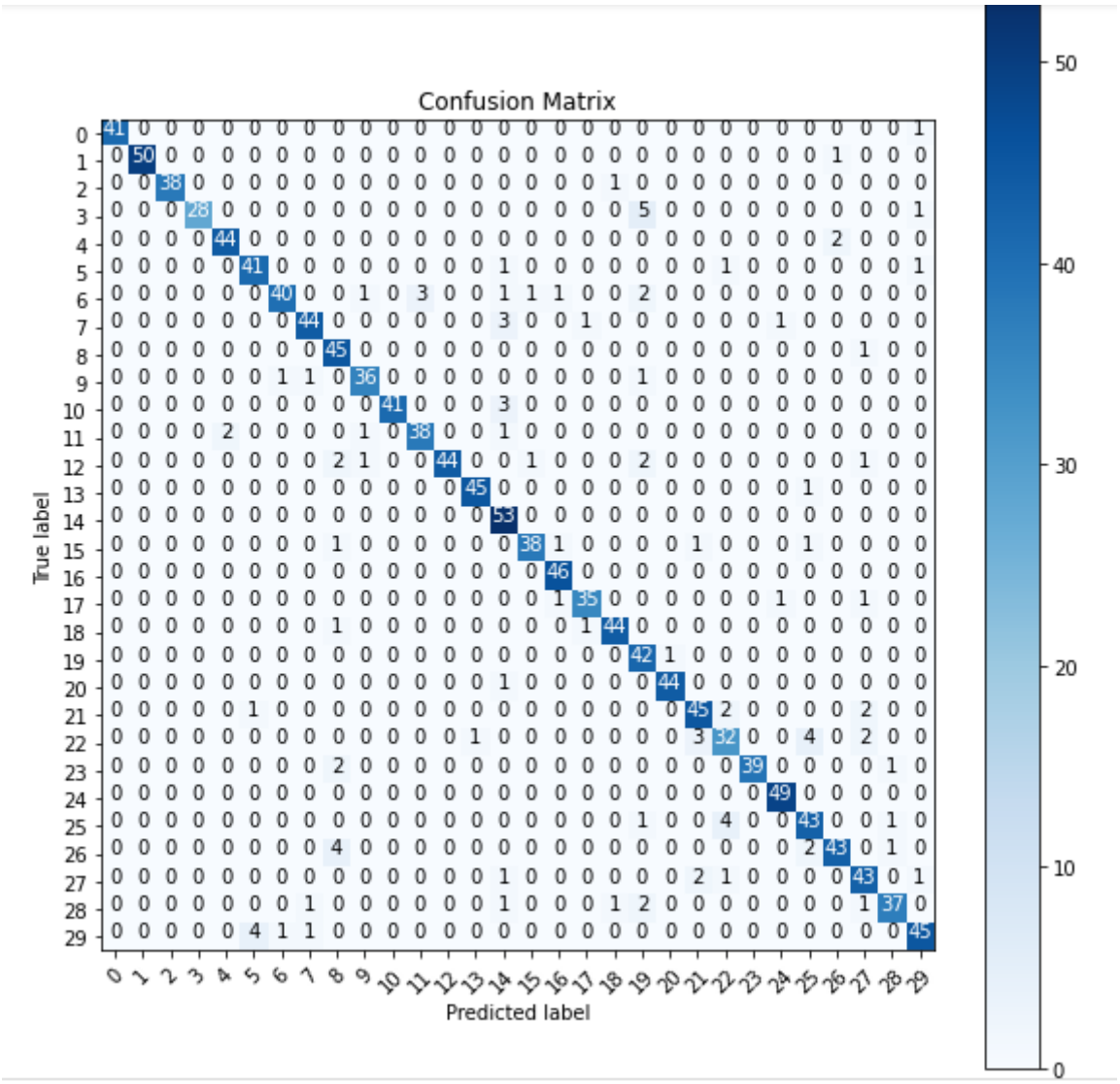

Figure 3.27 : Matrice de confusion mfcc13/11025

La figure 3.27 nous montre que le modèle a bien classé la plupart des échantillons.

- **Taux d'échantillonnage 8000 :**

Finalement on va tester par une base de données de taux d'échantillonnage 8000 Hz.

K-Fold	Précision	Erreur
5	90,00%	0,40
10	91,79%	0,39
15	87,38%	0,27
20	89,07%	0,29

Tableau 3.7 : Résultats obtenus mfcc13/8000

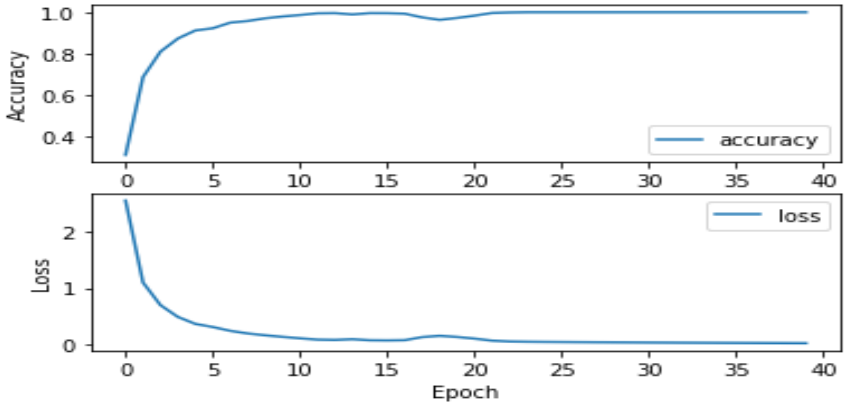

Figure 3.28 : précision et erreur mfcc13/8000

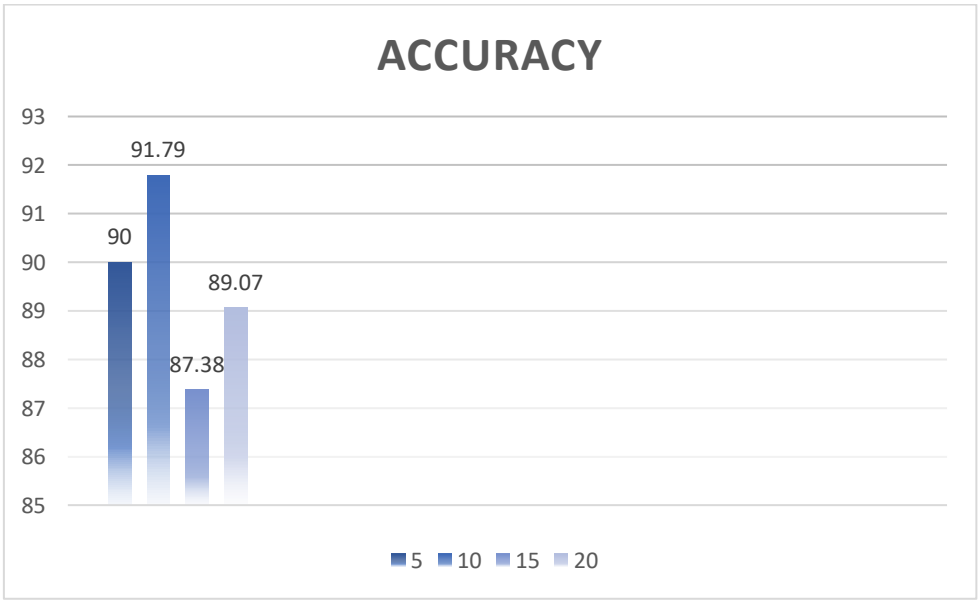

Figure 3.29 : Histogramme de précision mfcc13/8000

L'histogramme ci-dessus exprime les résultats de précision présenté dans le tableau.

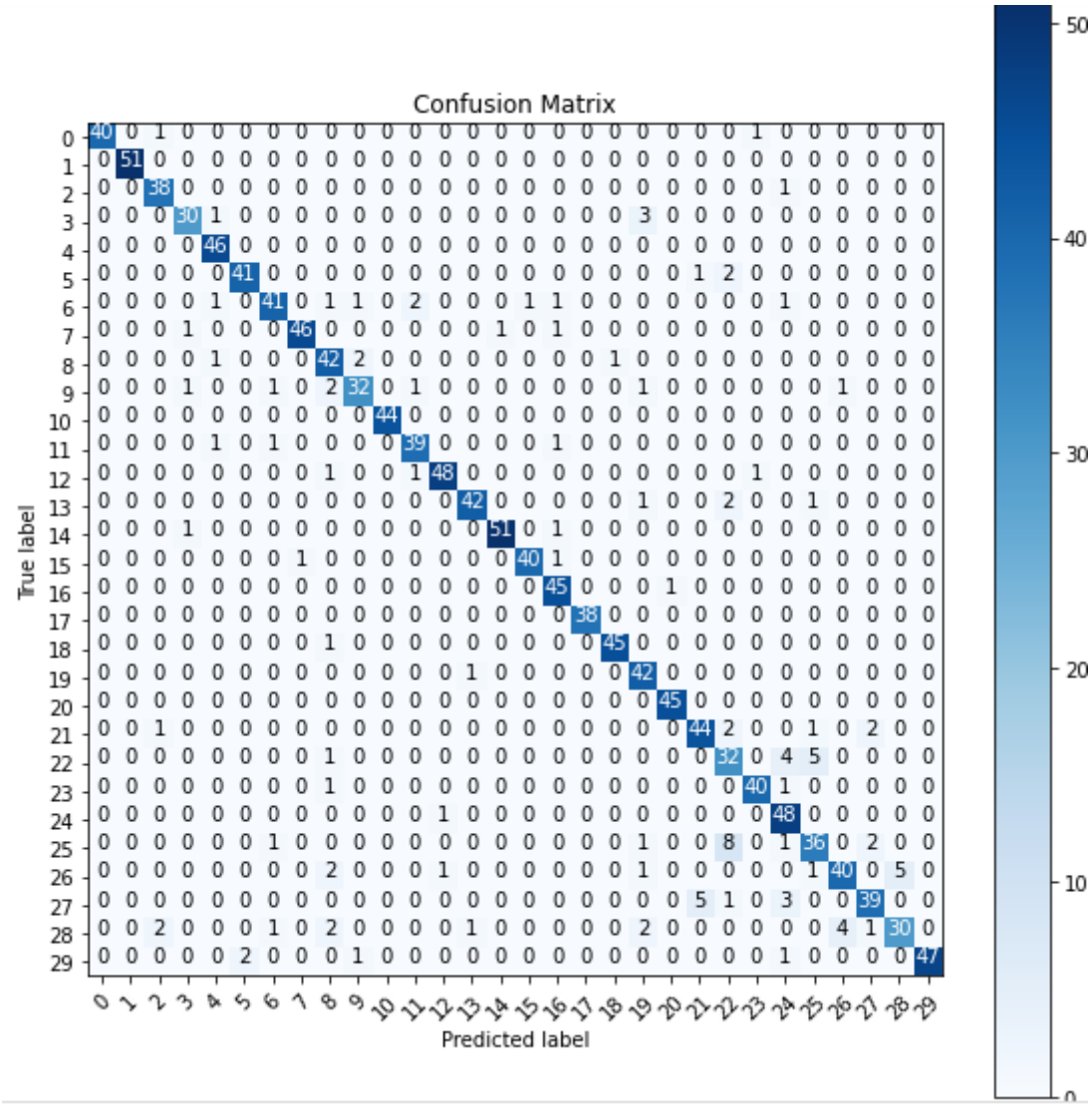

Figure 3.30 : Matrice de confusion mfcc13/8000

D'après la figure 3.30 on trouve que même pour le taux d'échantillonnage 8000 Hz le modèle a bien classé la majorité des échantillons.

D'après les tableaux 5, 6 et 7 on trouve que les résultats obtenus sont presque les mêmes, ce qui montre que si on réduire le taux d'échantillonnage ça n'influence pas au taux de précision de notre système, alors nous n'avons pas besoin d'utiliser des grands taux d'échantillonnage, puisque cela rendre la taille des fichiers plus grande.

8.2.2.3. Dataset équilibré

- **Taux d'échantillonnage 22050**

On avait mis en œuvre le réseau neuronal convolutif avec une base de données équilibré de taux d'échantillonnage 22050Hz et le nombre de MFCC est 13. Et pour cela on a atteint un taux de précision de 92,02% et un taux d'erreur de 0,38%.

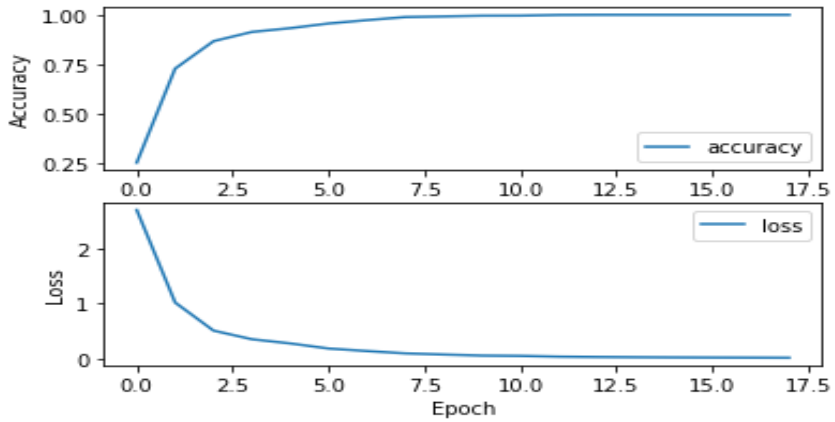

Figure 3.31 : précision et erreur 22050/13

- **Taux d'échantillonnage 11025**

Et pour finir on avait testé avec une dataset équilibré de taux d'échantillonnage 11025 Hz, on a obtenu 91,19 % taux de précision, et un taux d'erreur 0,36%.

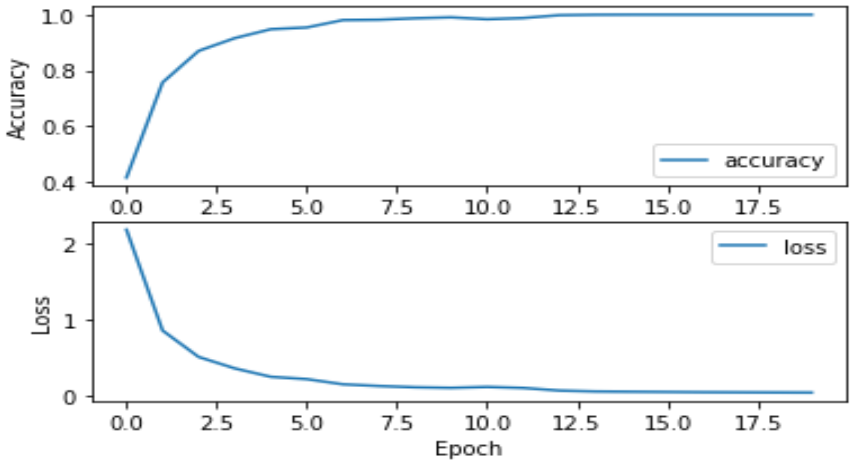

Figure 3.32 : précision et erreur 11025/13

Nous avons conclu que si on augmente le nombre de caractéristiques MFCC ou non ça ne change rien dans la précision du système, aussi pour le taux d'échantillonnage grand ou petit, il n'influence pas sur le taux de précision de notre système RAP. Alors il n'est pas obligé d'augmenter le nombre de MFCC ni avoir des bases de données de taux d'échantillonnage élevé, parce qu'a fur et à mesure que le nombre de MFCC augmenté ou le taux est élevé ça rend la taille des fichiers plus grandes.

9. Futures perspectives

Ce travail pour nous est une initiation à la recherche scientifique dans le domaine de la reconnaissance automatique de la parole, il reste encore beaucoup à faire en termes d'amélioration d'efficacité et de précision. C'était prévu une implémentation du système RAP mais qui reste toujours dans les perspectives envisagées. Pour la poursuite de ce modeste travail nous espérons généraliser le système RAP en enrichir notre base de données et élargir le vocabulaire contextuel et varier les conditions d'enregistrement pour qu'il sera valide dans d'autre domaine tels que le contrôle de voiture, la robotique, la domotique, etc.

D'autre part, pour un vocabulaire étendu, il est intéressant d'utiliser des modèles de phonèmes au lieu de mots, du fait que le nombre de phonèmes qui permet la construction de n'importe quel mot est faible, ce qui facilite l'entraînement avec des bases relativement petites.

Comparer les performances du système de reconnaissance proposé pour les différentes techniques d'analyse acoustique, les mêmes conditions d'expériences doivent être réutilisées pour les différents tests, en utilisant les coefficients LPC, PLP, DWT, ..., ainsi que des combinaisons éventuelles de ces derniers.

10. Conclusion

Nous avons parcouru dans ce chapitre l'architecture de notre système de reconnaissance automatique de la parole destiné aux sites web, suivi par une brève définition de Deep Learning, les logiciels et les bibliothèques utilisés lors de l'implémentation, ensuite nous avons présenté la structure du réseau neuronal convolutif, puis on a montré les résultats expérimentaux et les discussions, et pour finir on a parlé des futures perspectives.

Conclusion générale

Conclusion générale

La simplicité et la souplesse de la parole comme un moyen de communication justifient les recherches effectuées dans ce domaine.

La reconnaissance automatique de la parole recouvre tous les aspects liés à l'interprétation, par la machine, du langage humain. Les applications de cette technologie sont nombreuses : Navigation sur un serveur vocal au téléphone, Apprentissage d'une langue étrangère, commandes vocales dans les voitures, les téléphones ou bien encore dans les salles d'opérations chirurgicales, dictée vocale, identification vocale dans les zones sécurisées ou bien dans le cadre d'une enquête judiciaire, etc.

Malgré le grand nombre des arabophones dans le monde, et malgré l'arrivée de produits commerciaux des systèmes RAP au grand public avec une bonne qualité et financièrement accessibles la reconnaissance automatique de la parole arabe est à ces débuts par rapport à d'autres langues. Pour enrichir cette langue et rendre accessible à l'ensemble de la communauté scientifique et au grand public dans le monde nous prendrons la responsabilité de réaliser un système de télé-commandement vocal pour un site web en utilisant seulement la parole en langue Arabe.

Pour réaliser notre projet on a utilisé le deep Learning et nous avons choisi les réseaux de neurones convolutionnels (CNN) comme une technique d'apprentissage qui met en évidence un grand succès dans le domaine du traitement et de la reconnaissance automatique de la parole.

L'obtention des résultats de classification nécessite une base des données équilibré et bien organisée, et pour cela nous avons opté à créer notre propre base de données contenant des mots clés contextuels.

Les résultats obtenus sont très satisfaisants ou on a atteint un taux de précision de 92%, après l'étude effectuée, on peut conclure que le Deep Learning a donné une performance remarquable.

Bibliographie

[1] Julien RACHEDI, Reconnaissance et classification de phonèmes, Mémoire pour le Master Sciences et Technologie de l'UPMC, Paris, Mars / Aout 2005.

[2] Hervé Haut, Les systèmes de dictée continue, Publication technique de la SmalS-MvM, Bruxelles, 04/2005.

[3] Christophe LEVY, Modèles acoustiques compacts pour les systèmes embarqués, thèse de Doctorat en Sciences de l'Université d'Avignon et des Pays de Vaucluse, 30 novembre 2006.

[4] Andrzej Drygajlo, TRAITEMENT DE LA PAROLE, Groupe de traitement de la Parole et de Biométrie (GTPB), Martigny Lausanne, 2003.

[5] Olivier Deroo, Modèles dépendants du contexte et méthodes de fusion de données appliqués à la reconnaissance de la parole par modèles hybrides HMM/MLP, thèse de doctorat en sciences appliqués,22 décembre 1998.

[6] Thierry Dutoit, Introduction au Traitement Automatique de la Parole, Notes de cours / DEC2, Faculté Polytechnique de Mons –T. Dutoit, 2000

[7] H. Cerf-Danon, M. El-Bèze, B. Merialdo, Reconnaissance automatique de la parole, Centre Scientifique IBM -France, Paris, 1994

[8] Yala Nawel, Reconnaissance Automatique du Locuteur. Thèse de magistère, Université de Haouari Boumediene Algérie 2010.

[9] « Apple Launches iPhone 4S iOS5 iCloud » [archive], sur apple.com, 4 octobre 2011.

[10] « Microsoft annonce une avancée considérable en reconnaissance vocale » [archive], sur actuia.com.

[11] Sandeep Rathor, R. S. Jadon "Speech Recognition and System Controlling using Hindi Language", IEEE 2019.

[12] Tan Baohua, LI Jiaxiong "A Speech Remote Control System Realization Based on Computer Telecommunication Integration", IEEE 2011.

[13] El Amrani, M. Y., Rahman, M. H., Wahiddin, M. R., & Shah, A., "Building CMU Sphinx language model for the Holy Quran using simplified Arabic phonemes" Egyptian informatics journal, vol. 17, issue 3, pp. 305-314, 2016.

[14] Hae-Duck J. Jeong, Sang-Kug Ye, Jiyoung Lim, Ilsun You, and WooSeok Hyun, "A Remote Computer Control System Using Speech Recognition Technologies ofMobile Devices", IEEE 2013.

[15] George E. Dahl, Dong Yu, Li Deng and Alex Acero, “Context- Dependent Pre-Trained Deep Neural Network for Large- Vocabulary Speech Recognition”, IEEE transactions on Audio, Speech and Language Processing, Vol. 20, No. 1, 2012

[16] Chee Yang Loh, Kai Lung Boey and Kai Sze Hong, “Speech Recognition Interactive System for Vehicle”, IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA 2017), 10 - 12 March 2017, Penang, Malaysia, 2017

[17] Paul Jasmin Rani, Jason Bakthakumar, Praveen Kumaar.B, Praveen Kumaar.U and Santhosh Kumar, “VOICE CONTROLLED HOME AUTOMATION SYSTEM USING NATURAL LANGUAGE PROCESSING (NLP) AND INTERNET OF THINGS (IoT)”, Third International Conference on Science Technology Engineering & Management (ICONSTEM), IEEE 2017

[18] Mohammad A. M. Abu Shariah, Raja N. Ainon, Roziati Zainuddin, Othman O. Khalifa, “Human Computer Interaction Using Isolated-Words Speech Recognition Technology”, International Conference on Intelligent and Advanced Systems, IEEE 2017.

[19] Thierry Dutoit, Introduction au Traitement Automatique de la Parole, Notes de cours / DEC2, Faculté Polytechnique de Mons –T. Dutoit, 2000.

[20] https ://www.python.org/. Consulté avril 2021.

[21] https://www.spyder-ide.org/. Consulté avril 2021.

[23] Uday Kamath, John Liu, James Whitaker, “Deep Learning for NLP and Speech Recognition”, Springer 2019.

[24] https://ledatascientist.com/google-colab-le-guide-ultime/ Consulté Mai 2021

[25] https://www.tensorflow.org/ Consulté Mai 2021

[26] https://keras.io/ Consulté Mai 2021

[27] Moualek Djaloul. Y, « Deep Learning pour la classification des images », Thèse de Master en informatique, Université de Tlemcen, Algérie, 2017.

[28] Hub.packtpub, https.www.hub.packtpub.com/top-5-deep-learning-architectures/

[29] http://magnussvanfeldt.se/how-to-record-great-singing-vocals-on-your-smartphone/#:~:text=Settings%20to%20improve%20the%20recording%20quality,-Some%20sound%20recording&text=Microphone%20gain%20level.,this%20will%20result%20in%20distortion.