

République du Sénégal



Un Peuple - Un But - Une fois

UNIVERSITÉ ALIOUNE DIOP DE BAMBEY



**UFR : Sciences Appliquées et Technologies de l'Information et de la Communication
(SATIC)**

DEPARTEMENT : Technologies de l'Information et de la Communication (TIC)

SPECIALITE : Système d'Information (SI)

MEMOIRE

Présentée par

Fatou Bintou LOUCOUBAR

Pour l'obtention du diplôme de

Master en Système d'Information

SUJET :

Reconnaissance de mots wolof à l'aide de CNN : Le
cas d'une plateforme d'autodiagnostic COVID-19

Directeur de mémoire :

Dr. Christelle SCHARFF

Année Académique 2020-2021

DÉDICACE

Je dédie ce travail :

À ma très chère mère Awa SARR,

Honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de faire depuis ma naissance.

Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.

Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le TOUT PUISSANT, te préserver et t'accorder santé, longue vie et bonheur.

À mon très cher père Mor LOUCOUBAR,

Aucune dédicace ne saurait exprimer l'amour que j'ai toujours eu pour toi. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de vos sacrifices que vous avez consentis pour mon éducation et ma formation.

À mon très cher mari Mame Cheikh SARR,

Aucune dédicace ne saurait exprimer l'amour que j'ai pour toi. Merci pour ton grand soutien inébranlable durant ces mois.

REMERCIEMENTS

Je ne saurais commencer ce mémoire sans remercier ALLAH le tout puissant, qui m'a éclairé sur le chemin de la connaissance, par sa protection et sa bénédiction et qui m'a donné la force et la patience d'accomplir ce modeste travail.

Je remercie également Serigne Hamsatou MBACKE.

Arrivant à l'aboutissement de mon projet, je me trouve dans l'obligation respectueuse de devoir présenter mes chaleureux remerciements et témoignage de ma gratitude à tous ceux qui ont contribué aimablement et avec patience à l'élaboration de ce mémoire.

Tout d'abord, je tiens à remercier profondément mon encadreur Docteur Christelle SCHARFF, pour sa générosité en matière d'encadrement, pour ses judicieux conseils, son soutien moral et son aide durant l'élaboration de ce projet. Votre contribution est d'un intérêt capital. Je ne saurais vous remercier pour toute l'aide que vous m'avez fournie et l'intérêt que vous avez accordé à ce travail.

Merci à Krishna Mohan BATHULA et Kaleema UNNISA, qui m'ont toujours guidé et soutenu. Je vous remercie aussi de m'avoir accordé autant de temps et d'avoir contribué activement au travail de ce mémoire.

Un grand merci à tous les enseignants-chercheurs du département des technologies de l'information et de la communication de l'UFR SATIC de l'UADB qui ont participé pleinement à ma formation.

Un grand merci à toutes les personnes qui ont participé à la conception de mon ensemble de données.

Enfin, je ne saurais terminer sans remercier ma famille pour leurs encouragements, leur amour inconditionnel et leurs précieux conseils.

RÉSUMÉ / ABSTRACT

Résumé

Méthode d'apprentissage en Intelligence Artificielle, le deep learning (ou apprentissage profond) est bien connu pour son applicabilité à la reconnaissance d'images, mais une autre utilisation clé de la technologie est la reconnaissance vocale.

Les technologies vocales telles que les systèmes de reconnaissance automatique de la parole et de synthèse vocale avec les langues africaines peuvent jouer un rôle important dans la réduction de la fracture numérique en Afrique. En effet, ces technologies permettent de réduire le niveau de sophistication requis pour accéder aux services d'information et contribuent ainsi à l'établissement d'une société de l'information pleinement inclusive.

Dans ce contexte, notre mémoire propose un système de reconnaissance vocale pour les mots de la langue wolof. Nous proposons un modèle de deep learning (apprentissage profond) qui reconnaît les mots en wolof, "oui" et "non" en particulier. Ce modèle a été déployé sur une plateforme web utilisée pour aider au diagnostic des patients suspectés d'être atteints de la COVID-19. L'objectif de la plateforme est de contribuer à la lutte contre la pandémie de COVID-19 par un autodiagnostic rapide et la mise à disposition d'informations relatives à la COVID-19.

Pour atteindre notre objectif, nous avons enregistré des personnes prononçant "oui" et "non" en wolof afin de créer un ensemble de données de 310 enregistrements audios.

Nous avons ensuite utilisé le deep learning pour la reconnaissance vocale et créé un modèle de reconnaissance de mots en Wolof. Les réseaux de neurones convolutifs (CNN) se sont avérés très efficaces dans la classification d'images et sont prometteurs pour le traitement d'audio quand les sons sont transformés en spectres de fréquences du son. Pour faire de la classification sonore, nous avons donc appliqué une méthodologie de classification d'images à l'aide de CNN en transformant d'abord les audio en spectres.

Nous avons fait le prétraitement des caractéristiques à l'aide du MFCC (Mel-Frequency Cepstral Coefficients) et entraîné notre modèle à l'aide de CNN. Enfin nous avons terminé par créer une plateforme web qui permet à une personne de pouvoir utiliser le système de reconnaissance de mots Wolof que nous avons conçu.

Mots clés : Deep Learning, Reconnaissance Vocale, Langue Wolof, COVID-19, CNN, MFCC.

Abstract

A learning method in Artificial Intelligence, Deep Learning is well known for its applicability across various fields to solve problems related to image recognition and another significant use of the technology is speech recognition.

Speech recognition algorithms such as automatic speech recognition and text-to-speech systems with African languages can play an important role in bridging the digital divide in Africa. Indeed, these technologies reduce the level of sophistication required to access information services and thus contribute to the establishment of a fully inclusive information society.

In this context, our study proposes a speech recognition system for words in the Wolof language. We propose a deep learning model that recognizes wolof words, « yes » and « no » in particular. The model is deployed to a web platform used for screening the the patients suspected of suffering from COVID-19, and helping in the fight against the COVID-19 pandemic which has become a priority worldwide. The goal of the platform is to contribute to the fight against he COVID-19 pandemic with a rapid self-diagnosis and the provision of COVID-19 related information.

To achieve our goal, we recorded people pronouncing "yes" and "no" in Wolof to create a dataset of 310 audio recordings.

We used a deep learning algorithm for speech recognition and created a Wolof word recognition model. Convolutional neural networks (CNNs) have proven to be very effective in image classification and are promising for audio processing when sounds are transformed into sound frequency spectra. To perform voice classification, we first transformed the recordings into spectra, then applied an image classification methodology using CNNs.

We applied the feature preprocessing using MFCC (Mel-Frequency Cepstral Coefficients) and trained our model using CNN. We created a web platform built on the model that to allow any person to use the Wolof word recognition system we designed.

Keywords : Deep Learning, Voice Recognition, COVID-19, CNN, MFCC, Wolof language.

SOMMAIRE

DÉDICACE.....	i
REMERCIEMENTS	i
RÉSUMÉ / ABSTRACT.....	ii
SOMMAIRE	iii
LISTE DES FIGURES	vi
SIGLE ET ABRÉVIATION	vii
INTRODUCTION GÉNÉRALE.....	1
CHAPITRE 1 : PRÉSENTATION GÉNÉRALE	3
1.1. Présentation du Sujet	3
1.1.1. Contexte	3
1.1.2. Problématique.....	3
1.1.3. Objectifs	4
1.2. Cadre Théorique	4
1.2.1. COVID-19.....	4
1.2.1.1. Introduction.....	4
1.2.1.2. Solutions Adaptées dans le Monde	5
1.2.1.3. Solutions Numériques Adaptées au Sénégal.....	5
1.2.2. Reconnaissance Vocale	6
1.2.2.1. Techniques et Algorithmes Utilisés en Reconnaissance Vocale	6
1.2.2.2. Application de la Reconnaissance Vocale	8
1.2.2.3. Robustesse des Systèmes de Reconnaissance vocale	9
1.2.2.4. Problématiques Liés à la Reconnaissance Vocale	9
1.2.2.5. La Reconnaissance Vocale des Langues Locales	9
1.2.2.6. La Reconnaissance Vocale du Wolof	10
1.2.2.7. Notre contribution	10
1.2.3. Deep Learning	11
1.2.3.1. Fonctionnement du Deep Learning.....	11
1.2.3.2. Applications du Deep Learning	11
1.2.3.3. Réseaux de Neurones Convolutifs (CNN).....	12
1.2.3.4. Architecture d'un Réseau de Neurones Convolutif	12
1.2.3.5. Deep Learning Appliqué à la Reconnaissance Vocale	16
1.2.4. Son et Signal Sonore	17
1.2.4.1. Représentation Numérique du Son	18

1.2.4.2. Parole Humaine	18
1.2.4.3. Préparation des Données Audio pour un Modèle d'Apprentissage en Profondeur 19	
CHAPITRE 2 : MISE EN OEUVRE	22
2.1. Outils Utilisés	22
2.2. Implémentation du Système	24
2.2.1. Vue d'Ensemble du Système.....	24
2.2.2. Description des Données	25
2.2.3. Prétraitement des Données	25
2.2.3.1. Explication de la Méthodologie Utilisée.....	25
2.2.3.2. Étapes de Prétraitement des Données	27
2.2.4. Création et Entraînement du modèle	28
2.2.5. Réaliser une Prédiction.....	32
2.2.6. Validation du Modèle.....	32
CHAPITRE 3 : DÉPLOIEMENT DU MODÈLE.....	33
3.1. Architecture Client-Serveur du Système	33
3.2. Interfaces de la Plateforme Web.....	34
CONCLUSION GÉNÉRALE	38
RÉFÉRENCES.....	39

LISTE DES FIGURES

Figure 1: Classement des concepts IA, Machine Learning, Deep Learning. Source : [15]	11
Figure 2: Une séquence CNN pour classer les chiffres manuscrits. Source : [16]	12
Figure 3: Schéma d'une opération de pooling avec un noyau MaxPool de taille 2*2 et d'un pas de 2. Source : [17]	14
Figure 4: Schéma d'une couche entièrement connectée avec 6 classes. Source : [17]	15
Figure 5: Réseau de Neurones Convolutif avec ses différentes couches. Source: [16]	15
Figure 6: Étapes de la construction d'un modèle d'apprentissage audio en profondeur	17
Figure 7: Signal répétitif simple montrant l'amplitude en fonction du temps. Source : [18] ...	17
Figure 8: Forme d'onde musicale avec un signal répétitif complexe. Source : [19]	18
Figure 9: Mesures d'échantillons à intervalles de temps réguliers. Source : [20]	18
Figure 10: Spectre montrant les fréquences qui composent un signal sonore. Source : [21]...	19
Figure 11: Domaine temporel et domaine fréquentiel. Source : [22]	20
Figure 12: Spectrogrammes des mots Wolof « waaw » et « deideid ».	21
Figure 13: Signal sonore du mot Wolof “waaw” et son spectrogramme.	21
Figure 14: Architecture des différentes étapes de l'implémentation	25
Figure 15: Waveform du mot Wolof « waaw »	26
Figure 16: Spectrogramme du mot Wolof « deideid ».	26
Figure 17: MFCC du mot Wolof « deideid ».	27
Figure 18: Du fichier audio à la prédiction du mot Wolof	27
Figure 19: Architecture du Modèle.	30
Figure 20: Courbes d'entraînement et de validation	31
Figure 21: Architecture client-serveur du Projet	33
Figure 22 : Interface web qui permet la reconnaissance vocale	35
Figure 23: Cas où la réponse est "deideid"	35
Figure 24: Page d'accueil pour démarrer l'autodiagnostic	36
Figure 25: La première question de l'autodiagnostic	37
Figure 26: Type d'orientation	37
Figure 27: Questions et réponses automatiques sur la Covid-19 en Wolof	Erreur ! Signet non défini.

SIGLE ET ABRÉVIATION

COVID-19	Coronavirus Disease 2019
TIC	Technologies de l'Information et de la Communication
IA	Intelligence Artificielle
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
MFCC	Mel-Frequency Cepstral Coefficients
ReLU	Rectified Linear Unit
FC	Fully Connected
FFT	Fast Fourier Transform
API	Application Programming Interface
IDE	Integrated Development Environment
CPU	Central Processing Unit
GPU	Graphics Processing Unit
OMS	Organisation Mondiale de la Santé

INTRODUCTION GÉNÉRALE

Notre monde est en train de vivre une pandémie jamais vécue dont la transmission virale a surpris tout le monde. La COVID-19 n'a épargné quasiment aucun pays, même si c'est à des degrés divers. Face à cette situation, les gouvernements à travers le monde ont adopté des mesures, des stratégies et des politiques de lutte contre la pandémie du coronavirus. Certaines de ces stratégies et politiques s'appuient sur les outils et les services innovants qu'offrent les Technologies de l'Information et de la Communication (TIC).

En réalité, le numérique constitue pour l'État du Sénégal un secteur économique à part entière. Il ouvre de nouvelles perspectives, d'où l'introduction du concept moderne de « société de l'information » (Senegal Numerique 2025, SN2025).

Ainsi, pour éradiquer la pandémie, l'État, par le biais des technologies numériques, a mis en place des stratégies sanitaires, économiques, et juridiques. La meilleure façon de prévenir la maladie et de ralentir sa transmission est d'informer et d'être informé. Entre autres, l'État a mis en place une plateforme de sensibilisation et d'information (covid19.gouv.sn) et un chatbot Docteur Covid via WhatsApp qui permet de répondre aux questions des populations en lien avec le coronavirus.

Cependant, la plupart des plateformes disponibles pour s'informer sur la COVID-19 utilisent le français. Ceci est une vraie barrière pour beaucoup de personnes qui ne peuvent pas profiter des bienfaits de ces outils car elles ne parlent ou/et ne lisent pas le français. Dans un pays comme le Sénégal, où le taux d'alphabétisation est inférieur à 55%, rendre disponible ces plateformes d'orientation et d'information sous un format utilisant les langues locales pourrait aider à améliorer le taux de pénétration de ces nouvelles technologies et contribuer ainsi à la lutte contre la COVID-19. Par le biais de l'intelligence artificielle, il est également possible de développer des systèmes de reconnaissance vocale, de synthèse vocale et de traduction avec les langues locales.

C'est dans ce contexte que nous avons décidé de créer un système de reconnaissance vocale de mots wolof pour l'utiliser dans le cadre de la lutte contre la COVID-19 et permettre aux populations sénégalaises d'être mieux informées et orientées pour tout ce qui concerne cette pandémie.

Pour mener à bien notre travail, nous avons adopté le plan ci-dessous.

Dans le premier chapitre, nous allons présenter le contexte, la problématique, et les objectifs visés dans ce projet. Nous allons également aborder le cadre théorique en parlant de la COVID-19, de la Reconnaissance Vocale et du Deep learning.

Dans le deuxième chapitre, nous allons aborder tout ce qui concerne la mise en œuvre du projet en parlant des outils utilisés, des données collectées, du modèle construit avec les données et présenter les résultats obtenus.

Dans le troisième chapitre, nous allons couvrir le déploiement de notre modèle sur une plateforme web.

CHAPITRE 1 : PRÉSENTATION GÉNÉRALE

1.1. Présentation du Sujet

Dans cette partie, nous allons expliquer les grandes lignes de notre projet, son contexte, sa problématique et les objectifs visés.

1.1.1. Contexte

L'année 2020 a placé nos sociétés devant d'immenses défis. Les gouvernements ont dû réagir rapidement et efficacement à la situation exceptionnelle et évolutive causée par la pandémie de COVID-19. La pandémie a contraint les gouvernements à adopter des mesures rapides et efficaces et à recourir toujours davantage à des technologies numériques, notamment des applications mobiles et des chatbots. L'utilisation de technologies émergentes qui remplacent le contact humain par la communication à distance et d'algorithmes remplaçant l'intervention humaine a tout simplement explosé. L'intelligence artificielle (IA) s'est montrée efficace pour répondre à la crise sanitaire de la COVID-19. Des solutions diverses et variées ont été proposées, qu'il s'agisse de comprendre ce nouveau coronavirus, de s'autodiagnostiquer, d'en prévoir l'évolution, de ralentir sa propagation ou d'accélérer d'autres aspects de la recherche médicale.

En réalité, la meilleure manière de prévenir cette maladie est d'informer et d'être informé. Les populations l'ont compris. Par exemple, 6,8 millions de personnes dans le monde ont appelé le service IVR COVID-19 de Viamo en 2020 pour en savoir plus sur la façon dont le coronavirus se propage et sur les moyens de se protéger. 70 millions d'informations clés ont été écoutées dans 49 langues. Au Sénégal, le chatbot Docteur Covid via WhatsApp a été mis en place pour aider les populations à s'informer sur cette maladie.

De plus, la pandémie de la COVID-19 a mis en évidence les avantages de la commande sans contact des appareils. L'interface vocale connaît un essor important et devient accessible. Les enceintes connectées comme Alexa et Google Home sont déjà utilisées pour identifier des problèmes de santé en posant une suite de questions sur des symptômes et en reliant les utilisateurs aux sites de santé les plus proches.

Ce mémoire s'inscrit dans le contexte de lutte contre la COVID-19. Il s'agit d'aider les citoyens sénégalais, mêmes ceux qui ne comprennent pas le Français, à être mieux informés et participer également à la lutte contre la pandémie.

Pour résumer, ce mémoire vient à l'heure où les gouvernements utilisent les technologies numériques pour développer des solutions contre la propagation de la COVID-19.

1.1.2. Problématique

Il y a des décalages entre les solutions numériques proposées pour la lutte contre la COVID-19 et les besoins sur le terrain.

Beaucoup de sénégalais ne peuvent pas profiter des bienfaits de ces plateformes à cause de la barrière linguistique. Les plateformes d'information et d'orientation sur la COVID-19 sont, en effet, accessibles en français et plutôt dans des formats textes. Le wolof est la langue la plus parlée au Sénégal [1]. Le taux d'alphabétisation en français est de 51,9% au Sénégal en 2017 (hommes: 64,8% et femmes: 39,8%) [CIA World Factbook, 2019]. Des applications basées sur la voix sont cruciales pour une approche plus inclusive. Il y a un manque d'application de

l'intelligence artificielle en rapport avec les langues locales (reconnaissance vocale, synthèse vocale, traduction automatique). Les recherches sur les langues se focalisent surtout sur l'anglais et le français, très peu sur les langues locales.

1.1.3. Objectifs

Nos objectifs sont de mettre en place un système qui va permettre :

- de mettre la langue wolof sur la carte de l'intelligence artificielle
- d'inclure les populations ne parlant pas le français dans l'ère du numérique
- de proposer un système de reconnaissance vocale pour le Wolof
- de faire un autodiagnostic de la COVID-19 en langue Wolof
- d'informer les populations sur la COVID-19 en utilisant la langue Wolof

1.2. Cadre Théorique

1.2.1. COVID-19

1.2.1.1. Introduction

Qu'est-ce que la COVID-19 ?

La COVID-19 est une maladie provoquée par une nouvelle souche de coronavirus. D'abord appelée « nouveau coronavirus 2019 » ou « nCoV-2019 », la maladie a été rebaptisée « maladie à coronavirus 2019 » (COVID-19) – « CO » pour corona, « VI » pour virus et « D » pour maladie (en anglais) par l'Organisation Mondiale de la Santé (OMS). Le virus de la COVID-19 est un nouveau virus de la même famille que d'autres virus tels que le syndrome respiratoire aigu sévère (SRAS) et certains types de rhumes courants. Il est à l'origine d'infections pulmonaires et a été détecté en Chine en 2019.

Symptômes de la COVID-19

Les symptômes peuvent inclure de la fièvre, de la toux et un essoufflement. Dans les cas les plus graves, l'infection peut provoquer une pneumonie ou des difficultés respiratoires. La maladie peut également être mortelle. Dans beaucoup de cas, les symptômes sont comparables à ceux de la grippe (*influenza*) ou d'un rhume banal, des maladies beaucoup plus courantes que la COVID-19, d'où la nécessité de procéder à des examens afin de confirmer qu'une personne est bien atteinte de la COVID-19. Certains patients sont toutefois asymptomatiques.

Comment la COVID-19 se propage-t-elle ?

Le virus se transmet par contact direct avec les gouttelettes respiratoires produites par une personne infectée (lorsqu'elle tousse, éternue ou parle). Il est aussi possible d'être infecté en touchant des surfaces contaminées par le virus ou en se touchant le visage (par exemple, les yeux, le nez ou la bouche). Le virus de la COVID-19 peut survivre sur les surfaces pendant plusieurs heures, mais de simples désinfectants peuvent le tuer.

Personnes les plus à risque

Les personnes âgées et les personnes souffrant de maladies chroniques, telles que le diabète ou ayant une maladie cardiaque, semblent courir davantage de risques de développer des symptômes graves. La COVID-19 étant provoquée par un nouveau virus, il n'y a pas encore assez de données pour évaluer ses effets sur les enfants.

Comment traite-t-on la COVID-19 ?

Il existe maintenant des vaccins contre la COVID-19 (Pfizer, Moderna etc.). En plus, la plupart des symptômes de la maladie sont traitables et une prise en charge médicale rapide peut atténuer les risques.

Comment peut-t-on ralentir ou prévenir la propagation de la COVID-19 ?

Tout comme pour d'autres infections respiratoires telles que la grippe ou un rhume banal, les mesures de santé publique sont déterminantes pour ralentir la propagation de la maladie. De telles mesures sont des actions préventives appliquées au quotidien, qui incluent:

- de rester chez soi quand on est malade ;
- de se couvrir la bouche et le nez avec le pli du coude ou un mouchoir en cas de toux ou d'éternuement, puis de jeter immédiatement le mouchoir usagé ;
- de se laver fréquemment les mains avec de l'eau et du savon ;
- de nettoyer fréquemment les surfaces et les objets que l'on touche.

1.2.1.2. Solutions Adaptées dans le Monde

Depuis l'irruption et la propagation de la pandémie du COVID-19, notre mode de vie et système de fonctionnement ont soudainement changé. Le monde entier essaye de s'adapter pour faire face à la propagation de ce virus.

Dans une réaction, désormais, globalisée, des mesures sans précédent, souvent drastiques, sont prises par tous les pays pour protéger les populations et éradiquer cette pandémie qui ne connaît pas de frontière. Tous les États ont appliqué à des moments différents des restrictions de circulation, voire des fermetures de frontières.

L'Organisation Mondiale de la Santé (OMS) et ses partenaires ont mené des actions pour riposter à cette pandémie. Parmi ces actions on peut citer le suivi de la pandémie, les conseils donnés sur les interventions essentielles, la distribution de fournitures médicales vitales à ceux qui en ont besoin.

En Europe, la situation sanitaire avait contraint les autorités à prendre des mesures comme le confinement, la fermeture d'écoles et d'université, l'interdiction des rassemblements, et le port obligatoire du masque pour lutter contre la propagation de la COVID-19.

La plupart des pays africains ont rapidement pris des mesures pour limiter les déplacements et les rassemblements et mis en œuvre des mesures clés de santé publique.

Parallèlement aux actions menées par les pays pour lutter contre cette pandémie, une course aux vaccins a été engagée. L'OMS a accordé une autorisation d'utilisation d'urgence du vaccin contre la COVID-19 de Pfizer (BNT162b2) le 31 décembre 2020. Des vaccins sont apparus rapidement : AstraZeneca/Oxford le 15 février 2021, Ad26.COV2.S le 12 mars 2021 etc. La vaccination a visé, dans tous les pays, en priorité les populations vulnérables.

1.2.1.3. Solutions Numériques Adaptées au Sénégal

Les TIC sont également l'un des moyens qui ont permis à l'État du Sénégal de lutter contre la propagation du coronavirus.

Sur le plan sanitaire, le numérique est d'une grande nécessité aussi bien au niveau de la sensibilisation, de l'information, de la prévention et du traitement de la COVID-19 que du mode organisationnel des différents acteurs. C'est dans ce cadre que l'Agence De l'Informatique de l'État (ADIE), actrice majeure dans le secteur du numérique au Sénégal a tenu à jouer un rôle crucial. À cet effet, elle a eu recours à l'Intelligence Artificielle et a mis à la disposition du Ministère de la Santé et de l'Action Sociale (MSAS) une plateforme, un chatbot et des outils pour une communication à distance efficace pour la gestion de la crise.

De plus, l'ADIE a doté les forces de défense et de sécurité et le SAMU de téléphones offrant des options de communication par voix ou vidéo avec partage de documents et intégration de la géolocalisation. Ces outils de communication et de coordination permettent aux autorités impliquées dans cette lutte d'être opérationnels.

L'ADIE a aussi appuyé le MSAS dans la lutte contre le COVID-19 à travers la mise à disposition de téléphones eLTE pour la coordination des opérations sur le terrain, la mise en place de la plateforme de sensibilisation et d'information (covid19.gouv.sn), et d'un agent conversationnel, à travers le numéro de téléphone 76 600 05 26 pour répondre aux questions des populations en lien avec le coronavirus.

La crise sanitaire mondiale COVID-19 a suscité des élans de solidarité et de créativité exceptionnels. Au Sénégal, des jeunes talentueux ont mis leur créativité au service de leur pays pour faire face aux conséquences de la pandémie.

L'invention de Doctor Car dans le secteur de la santé en est un exemple [2]. Ce robot développé par des étudiants de l'ESP Dakar permet de limiter les contacts avec les patients en isolation atteints de la COVID-19 en livrant médicaments et nourriture ou encore en prenant leur température.

La plateforme Xel-Xeeli Academy lancée par l'association Jamarek favorise l'autonomie éducative sénégalaise en cette période de crise [3]. Enfin, de nombreuses innovations citoyennes ont vu le jour avec le développement d'applications d'entraide.

Le vélo Mobigel distribue du gel hydroalcoolique pour aider les personnes dans l'impossibilité de rester chez elles [4]. Il a été conçu par des étudiants d'ENACTUS ESP.

1.2.2. Reconnaissance Vocale

La reconnaissance vocale est un domaine couvrant tous les aspects liés à l'interprétation, par la machine, du langage humain.

En effet, la parole est un moyen de communication primordial chez l'homme. Elle est tellement riche en informations que les scientifiques essaient sans cesse de l'analyser afin d'en comprendre les différents aspects. Depuis les années 1950 [5], de nombreuses équipes de chercheurs (informaticiens, phonéticiens, mathématiciens, linguistes...) se sont penchées sur un objectif commun : automatiser les processus d'interprétation de la parole, mais aussi de sa production. La reconnaissance automatique de la parole (RAP), la reconnaissance du locuteur et la synthèse de la parole ont particulièrement intéressé les académiques ainsi que les entreprises. Les résultats de ces problématiques sont visibles actuellement dans les interfaces d'échange de nos appareils intelligents, notamment nos téléphones et nos assistants vocaux.

1.2.2.1. Techniques et Algorithmes Utilisés en Reconnaissance Vocale

Traditionnellement, deux méthodologies sont proposées en reconnaissance de la parole : l'approche analytique et l'approche globale. Les réseaux de neurones (récurrent et convolutif) sont aussi utilisés pour faire de la reconnaissance de la parole.

Approche Globale

L'approche globale évite toute segmentation a priori et ne fait pas d'hypothèses sur le type des éléments à traiter. Elle effectue des comparaisons sur un ensemble de références en traitant les données et les connaissances dans leurs globalités. Le principe de base de cette méthode consiste à donner au système de reconnaissance au moins une image de chacune des unités qu'il est censé devoir identifier par la suite. Cette opération est faite lors de la phase d'apprentissage qui permet de constituer la base de données de référence du système. Le processus de décodage consiste alors à comparer l'image de l'unité à identifier avec celles de la base de référence. L'unité dont la référence est la plus proche est déclarée reconnue. Par ailleurs, dans la mesure où les données à traiter sont constituées d'une suite d'unités (comme, par exemple, la reconnaissance d'une phrase composée de mots), l'unité de base sera le plus souvent le mot considéré comme une entité globale, c'est à dire non décomposée.

Approche Analytique

Les caractéristiques principales de l'approche analytique sont une segmentation a priori du signal acoustique, une organisation modulaire hiérarchique et l'utilisation de bases de connaissances formelles. Elle tire parti de la structure linguistique des mots et tente de détecter et d'identifier les composantes élémentaires (phonèmes, syllabes, etc.). Celles-ci sont les unités de base à reconnaître. Cette approche a un caractère plus général que l'approche globale : pour reconnaître de grands vocabulaires, il suffit d'enregistrer dans la mémoire de la machine les principales caractéristiques des unités de base. La méthode analytique est la plus utilisée car les mots ne sont pas mémorisés dans leur intégralité, mais traités en tant que suite de phonèmes, la capacité de mémoire nécessaire est donc moins importante.

Les Réseaux de Neurones Récurrent (RNN) pour la Reconnaissance de la parole

Les RNN sont souvent utilisés en Natural Language Processing et leurs applications en reconnaissance vocale sont très prometteuses. En effet, ils constituent un modèle puissant et expressif pour les données séquentielles. La caractéristique principale des RNN est que les neurones internes à une couche peuvent être reliés entre eux. Ils peuvent également être reliés à eux même, et à des neurones de couches précédentes en sens inverse. Cette caractéristique correspond à la prise en compte du temps dans le réseau de neurone : les données introduites à un instant T dans le réseau sont réintégrées dans le calcul des nouvelles données. La forme de RNN la plus utilisée est celle des Long Short Term Memory (LSTM) qui permettent de ne conserver que les informations les plus pertinentes du passé.

Les Réseaux de Neurone Convolutif (CNN) pour la Reconnaissance de la Parole

Le CNN est une architecture de réseau neuronal profond qui relève du domaine des réseaux neuronaux artificiels (ANN). Il s'agit d'un type de réseau qui donne de bons résultats sur des données de type image, en raison de la manière dont il gère le grand nombre de paramètres qui découlent de l'utilisation de données de type image comme entrées. En empilant les couches CNN, le réseau est capable d'utiliser la non-linéarité pour apprendre les caractéristiques spatiales locales d'un volume d'entrée. Ce format convient à la tâche de prédiction des caractéristiques spectrales puisqu'elles représentent une cartographie dimensionnelle comprimée de l'énergie des signaux. Un spectrogramme d'un signal trace son spectre au fil du temps et ressemble à une « photographie » du signal. Il trace le temps sur l'axe des x et la

fréquence sur l'axe des y. Les spectrogrammes sont généralement visualisés sous la forme d'images en niveaux de gris. C'est cette technique que nous utilisons dans ce projet de mémoire.

Comparaison des Techniques

Approche Globale	Approche Analytique	RNN	CNN
L'unité de base est le mot (donc non décomposable). Elle se limite aux petits vocabulaires prononcés par un nombre restreint de locuteurs (les mots peuvent être prononcés de manière différente suivant le locuteur).	Elle utilise la structure des mots en identifiant les composantes élémentaires (phonèmes, syllabes). Elle est plus adaptée à reconnaître de grands vocabulaires.	Ils ont des capacités d'apprentissage de données séquentielles. Ils ont amélioré la reconnaissance de grands vocabulaires et la synthèse texte-parole.	Ils offrent des avantages dans quatre domaines : conditions d'entraînement et de test non concordantes, robustesse au bruit, reconnaissance de la parole à distance et modèles à faible encombrement.

1.2.2.2. Application de la Reconnaissance Vocale

Ces dernières années, on peut la place de plus en plus grande de la communication vocale dans nos appareils intelligents. Aujourd'hui, on peut activer son smartphone via une simple commande vocale, on peut dicter à sa voiture la destination souhaitée, et on peut même demander à un assistant vocal de commander à manger.

Assistants Virtuels

Les assistants virtuels peuvent être intégrés dans plusieurs types de plateformes comme Amazon Alexa, Siri ou dans des appareils comme des enceintes intelligentes telles qu'Amazon Echo ou Google Home.

Amazon Echo est une enceinte connectée conçue par Amazon, ayant la capacité d'obéir à la voix humaine, de parler et, dans une certaine mesure, d'interagir avec un humain. L'appareil peut être connecté à des objets domotiques qui peuvent ainsi être contrôlés par la voix humaine.

Alexa est le nom de l'assistant personnel virtuel développé par Amazon, rendu populaire par les appareils Echo. Il est capable d'interaction vocale, de lire de la musique, faire des listes de tâches, régler des alarmes, lire des podcasts et des livres audio, et donner la météo, le trafic et d'autres informations en temps réel. Alexa peut également contrôler plusieurs appareils intelligents en faisant office de hub domotique.

Siri est une application informatique de commande vocale développée par Apple qui comprend les instructions verbales données par les utilisateurs et répond à leurs requêtes. Les résultats renvoyés sont individualisés. Siri permet de réaliser plusieurs interactions entre la voix de l'utilisateur et les applications du système iOS comme le navigateur Safari, les applications de SMS, l'application téléphone, l'application Mail ou encore l'application de cartographie.

Google Home est une famille d'enceintes connectées associées à un assistant personnel intelligent fabriqué par Google. Elles sont munies d'un haut-parleur et de 1 à 6 microphones selon le modèle, qui permettent aux appareils de réagir aux commandes vocales des utilisateurs.

Systèmes Vocaux dans le Domaine de la Santé

Dans le contexte de la COVID-19, Allocovid [6] est une plateforme téléphonique intelligente. Elle a été créée et développée par des chercheurs et experts de l'INSERM (Institut national de la santé et de la recherche médicale) et de l'Université de Paris, en collaboration avec Voyageurs SNCF, filiale digitale de la SNCF, et Allo-Media, une startup spécialisée dans l'intelligence artificielle. Cet agent virtuel est capable de synthétiser les informations médicales transmises par ses interlocuteurs et de les informer sur leur susceptibilité d'être atteintes ou non par la COVID-19. Il est aussi capable de détecter les signes de gravité de la maladie, ainsi que les patients vulnérables nécessitant une attention particulière.

1.2.2.3. Robustesse des Systèmes de Reconnaissance vocale

Un système de reconnaissance de la parole est robuste quand il est capable de fonctionner dans des conditions difficiles. Nous pouvons citer quelques exemples de ces conditions :

- Bruits d'environnement (dans une rue, etc.)
- Déformation de la voix des interlocuteurs par l'environnement (réverbérations, échos, etc...).
- Qualité du matériel utilisé (micro, carte son etc...)
- Elocution inhabituelle ou altérée des interlocuteurs (stress, émotions, fatigue, etc...).

Certains systèmes peuvent être plus robustes que d'autres face à l'une ou l'autre de ces perturbations, mais, en règle générale, les systèmes de reconnaissance de la parole sont encore sensibles à ces perturbations.

1.2.2.4. Problématiques Liés à la Reconnaissance Vocale

Plusieurs problèmes font que la reconnaissance de la parole est un domaine difficile. Il y a une grande variabilité de la parole en plus de la continuité et de la coarticulation. Nous pouvons citer quelques exemples :

- Variabilité intralocuteur : voix chantée, crie, murmurée, sous stress, bégaiement etc.
- Variabilité interlocuteur : timbres différents, voix masculines, féminines, voix d'enfants etc.
- La production d'un son est fortement influencée par le son qui le précède et qui le suit en raison de l'anticipation du geste articulatoire.

1.2.2.5. La Reconnaissance Vocale des Langues Locales

Les technologies de la parole peuvent jouer un rôle important dans l'adoption d'autres technologies. L'apprentissage automatique permettrait de comprendre des centaines de langues et faciliterait les opportunités et l'accès à l'information entre les pays. Cependant, le traitement des langues africaines a accumulé un retard considérable. Ce retard constitue pour l'Afrique un

handicap pour le développement social et économique. Pour combler ce retard, plusieurs projets sont récemment nés pour revaloriser les langues africaines en les dotant de ressources linguistiques. Parmi ces projets nous pouvons citer:

- Le projet DiLaf vise à informatiser des dictionnaires langues africaines-français (bambara, haoussa, kanouri, tamajaq, songhai-zarma) afin de pouvoir les diffuser plus largement et étendre leur couverture [7].
- Le projet ALFFA (African Languages in the Field: speech Fundamentals and Automation) vise à proposer, à terme, des micros services vocaux pour les téléphones mobiles en Afrique [8].
- Grâce à l'intelligence artificielle, Google espère améliorer, entre autres, ses moyens de traduction. Les langues africaines sont pour l'instant très mal représentées dans le corpus de Google Translate, son outil de traduction.

1.2.2.6. La Reconnaissance Vocale du Wolof

Le Sénégal étant un pays riche de par sa diversité ethnique avec une multitude de langues locales (25 environs) dont le Wolof ou Ouolof qui est la langue maternelle de l'ethnie Wolof. Il est aussi parlé en Gambie et en Mauritanie. Le Wolof est une langue orale africaine typique. Cela signifie qu'aucune grammaire formelle n'a été définie. Très peu de dictionnaires [9] ont été produits et la prononciation du même graphème peut être très différente. Le Wolof est compris et parlé par environ 80% des sénégalais, tandis que, la majorité des sénégalais ne peut lire ni écrire le français langue officielle du pays [10].

Ainsi, l'utilisation de l'intelligence artificielle peut relever le défi des langues locales et permettre à tout un chacun de communiquer dans sa langue natale. On peut citer quelques travaux sur l'utilisation des langues locales dans les TICs.

- Le projet iBaatukaay est un projet dont l'objectif est la conception d'une base lexicale multilingue contributive sur le Web pour les langues africaines notamment sénégalaises. C'est un projet collaboratif [11].
- L'entreprise sénégalaise Baamtu développe des systèmes de reconnaissance vocale, de synthèse vocale et de traduction avec les langues locales comme le Wolof [12]. Dans le domaine du transport, Weego s'est allié à Baamtu pour proposer leur moteur de suggestion d'itinéraire en langue locale à travers la reconnaissance vocale et le traitement du langage naturel [13].
- Le projet SYSNET3LOc a permis de construire un corpus qui contient environ 70000 phrases parallèles Français-Wolof de différents domaines [14].

1.2.2.7. Notre contribution

Nous avons construit un modèle de Deep learning qui reconnaît oui et non en Wolof. Pour ce faire, nous avons collecté un ensemble de données contenant 310 enregistrements audios. Notre modèle a atteint 97% de prévision après entraînement sur ces données. Le modèle a été déployé dans une application web permettant à une personne de faire un autodiagnostic de la COVID-19.

1.2.3. Deep Learning

Le deep Learning ou apprentissage profond est l'une des technologies principales du machine learning. Avec le deep Learning, nous parlons d'algorithmes capables de mimer les actions du cerveau humain grâce à des réseaux de neurones artificiels. Les réseaux sont composés de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente.

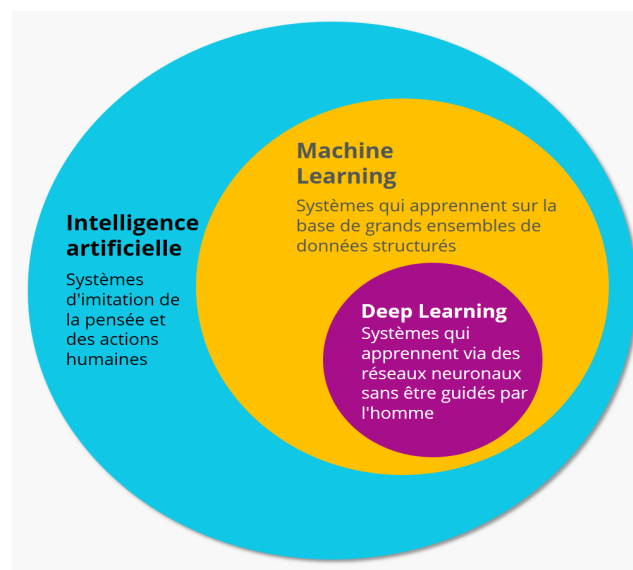


Figure 1: Classement des concepts IA, Machine Learning, Deep Learning. Source : [15]

1.2.3.1. Fonctionnement du Deep Learning

Comme à l'intérieur du cerveau humain, les signaux voyagent entre les neurones artificiels. Dans le cas de la reconnaissance visuelle, pour être performant, l'algorithme du deep Learning doit être capable d'identifier toutes les formes existantes et dans tous les angles. Ainsi, il sera capable de détecter une voiture sur la route au milieu du paysage. Ceci n'est possible que si la machine a suivi un entraînement poussé. Et ceci passe par la visualisation de milliers de photographies sur lesquelles apparaissent une voiture, de toutes les formes et dans tous les angles possibles. Lorsque l'image nouvelle apparaît, elle est envoyée au réseau de neurones qui se charge de l'analyser et de déterminer si l'objet est bel et bien une voiture.

1.2.3.2. Applications du Deep Learning

Le deep learning est utilisé dans de nombreux domaines :

- reconnaissance d'image,
- reconnaissance vocale,

- traduction automatique,
- recommandations personnalisées,
- chatbots (agents conversationnels)

1.2.3.3. Réseaux de Neurones Convolutifs (CNN)

Dans le cadre du deep Learning, un réseau de neurone convolutif ou CNN est un type de réseau neurone artificiel, largement utilisé pour la reconnaissance et la classification des images/objets. Le deep Learning reconnaît donc des objets dans une image en utilisant un CNN.

Un réseau de neurones convolutifs (ConvNet/CNN) est un algorithme du deep Learning (apprentissage en profondeur) qui peut prendre une image d'entrée, attribuer une importance (poids et biais apprenables) à divers aspects de l'image et être capable de les différencier les unes des autres. Les réseaux neuronaux convolutifs sont généralement formés comme des méthodes supervisées, ce qui signifie que les entrées (c'est-à-dire les caractéristiques des images dans une tâche de reconnaissance d'images) et leurs étiquettes (c'est-à-dire les labels des images) sont disponibles dans les données d'entraînement du modèle.

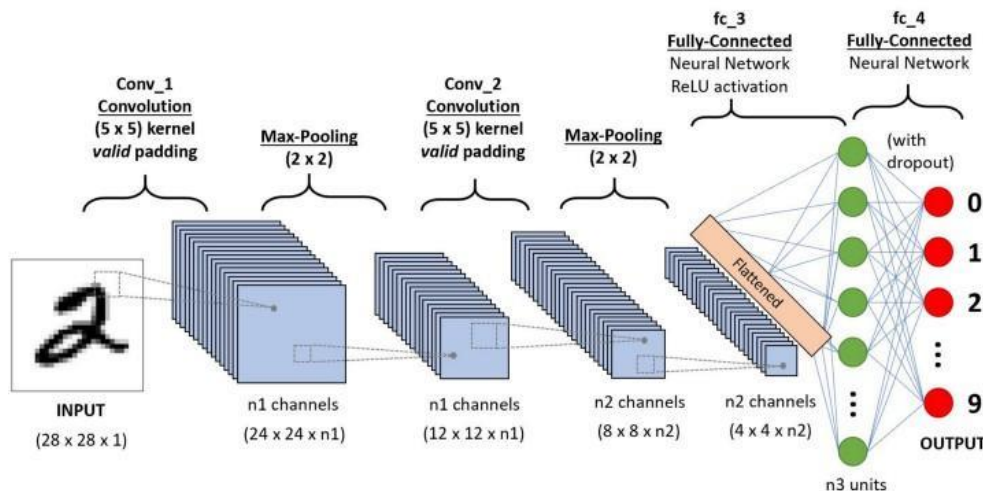


Figure 2: Une séquence CNN pour classer les chiffres manuscrits. Source : [16]

1.2.3.4. Architecture d'un Réseau de Neurones Convolutif

L'architecture d'un réseau de neurones convolutifs est formée par une succession de blocs de traitement pour extraire les caractéristiques discriminant la classe d'appartenance de l'image des autres. Un bloc de traitement se compose d'une à plusieurs :

- couches de convolution (CONV) qui traitent les données d'un champ récepteur ;
- couches de correction (ReLU), souvent appelée par abus « ReLU » en référence à la fonction d'activation (Unité de rectification linéaire) qui permet de filtrer les données;
- couches de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).

Les blocs de traitement s'enchaînent jusqu'aux couches finales du réseau qui réalisent la classification de l'image et le calcul de l'erreur entre la prédiction et la valeur cible :

- couche « entièrement connectée » (FC), qui est une couche de type perceptron ;
- couche de perte (LOSS) qui calcule l'erreur entre la valeur prédite et la valeur cible.

La façon dont s'enchaînent les couches de convolution, de correction et de pooling dans les blocs de traitement, ainsi que les blocs de traitement entre eux, font la particularité de l'architecture du réseau.

La Couche de Convolution (CONV)

L'image est découpée en sous-régions, appelées tuiles et analysée par un noyau de convolution. Ce noyau de convolution a la taille d'une tuile, souvent 3*3 ou 5*5. La zone analysée (champ réceptif) est légèrement plus grande que le noyau de façon à ce que les champs réceptifs se chevauchent. Cela permet d'obtenir une meilleure représentation de l'image et d'améliorer la cohérence du traitement de celle-ci.

L'analyse des caractéristiques de l'image par le noyau de convolution est une opération de filtrage avec une association de poids à chaque pixel. L'application du filtre à l'image est appelée une convolution.

Après une convolution, une carte de caractéristiques (en anglais features map) est obtenue, c'est une représentation abstraite de l'image. Ses valeurs dépendent des paramètres du noyau de convolution appliqué et des valeurs de pixels de l'image d'entrée.

Une couche de convolution est un empilement de convolutions. En effet, l'image est parcourue par plusieurs noyaux de convolution qui donnent lieu à plusieurs cartes de caractéristiques de sorties. Chaque noyau de convolution possède des paramètres spécifiques à l'information qui est recherchée dans l'image.

Le choix des paramètres du noyau de convolution dépend de la tâche à résoudre. Avec les méthodes Deep learning, ces paramètres sont automatiquement appris par l'algorithme à partir des données d'entraînement.

La Couche de Correction (ReLU)

La couche de correction ou d'activation est l'application d'une fonction non-linéaire aux cartes de caractéristiques en sortie de la couche de convolution. En rendant les données non-linéaires, elle facilite l'extraction des caractéristiques complexes qui ne peuvent pas être modélisées par une combinaison linéaire d'un algorithme de régression.

Les fonctions non-linéaires les plus utilisées sont :

- sigmoïde ou logistique,

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

- tangente hyperbolique,

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

- Unité de rectification linéaire (ReLU).

$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$

Bien souvent, la fonction ReLU est choisie, car elle maximise la décision de la fonction affine appliquée par convolution.

La Couche de Pooling (POOL)

L'étape de pooling est une technique de sous-échantillonnage. Généralement, une couche de pooling est insérée régulièrement entre les couches de correction et de convolution. En réduisant la taille des cartes de caractéristiques, donc le nombre de paramètres du réseau, cela accélère le temps de calcul et diminue le risque de surapprentissage.

L'opération de pooling la plus courante est celle du maximum : MaxPool (2*2, 2). Elle est plus efficace que la moyenne, car elle maximise le poids des activations fortes. Elle est appliquée à la sortie de la couche précédente comme un filtre de convolution de taille (2*2) et se déplace avec un pas de 2. En sortie de la couche de pooling est obtenue une carte de caractéristique compressée par un facteur de 4.

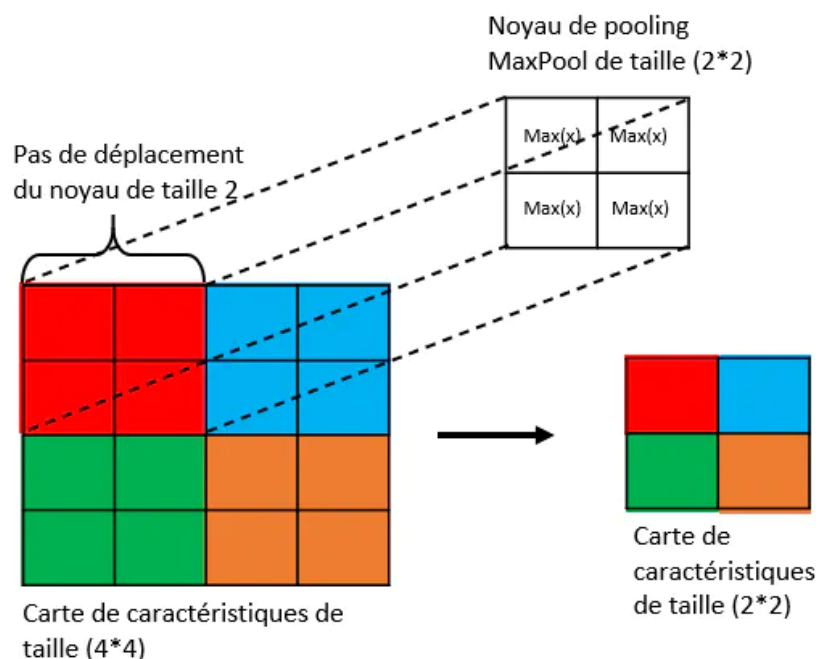


Figure 3: Schéma d'une opération de pooling avec un noyau MaxPool de taille 2*2 et d'un pas de 2. Source : [17]

La Couche "entièrement connectée" (FC)

Cette couche est à la fin du réseau. Elle permet la classification de l'image à partir des caractéristiques extraites par la succession de bloc de traitement. Elle est entièrement connectée, car toutes les entrées de la couche sont connectées aux neurones de sorties de celle-ci contrairement à la phase d'extraction des caractéristiques où les neurones de traitement sont indépendants entre eux et ont uniquement accès à l'information du champ réceptif qu'ils traitent. Dans les FC, les neurones ont accès à la totalité des informations d'entrée. Chaque

neurone attribue à l'image une valeur de probabilité d'appartenance à la classe i parmi les C classes possibles.

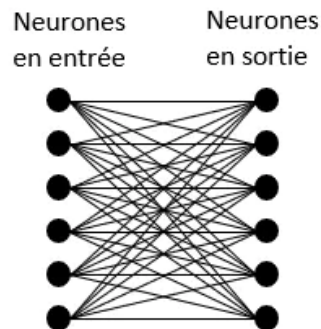


Figure 4: Schéma d'une couche entièrement connectée avec 6 classes. Source : [17]

La Couche de Perte (LOSS)

La couche de perte est la dernière couche du réseau. Elle calcule l'erreur entre la prévision du réseau et la valeur réelle. Lors d'une tâche de classification, la variable aléatoire est discrète, car elle peut prendre uniquement la valeur 0 ou 1, représentant l'appartenance (1) ou non (0) à une classe. C'est pourquoi la fonction de perte la plus courante et la plus adaptée est la fonction d'entropie croisée (en anglais cross-entropy).

Celle-ci est issue du domaine de la théorie de l'information, et mesure la différence globale entre deux distributions de probabilité (celle de la prévision du modèle, celle du réel) pour une variable aléatoire ou un ensemble d'événements. Formellement, elle s'écrit :

$$\text{loss}(x, \text{class}) = - \sum_{\text{class} = 1}^C y_{x, \text{class}} \log(p_{x, \text{class}})$$

Avec y la probabilité estimée d'appartenance de x à la classe i , p la probabilité réelle d'appartenance de x à la classe i , sachant qu'il y a C classes.

La figure suivante montre les différentes couches ensembles:

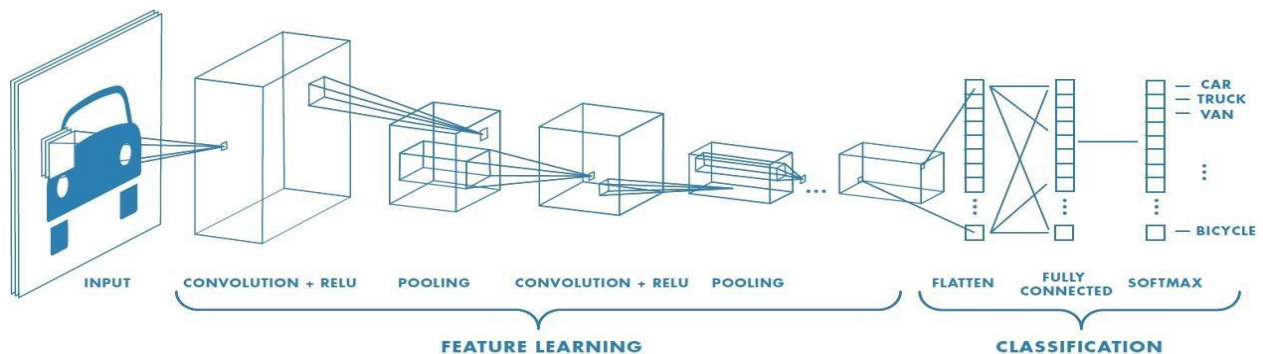


Figure 5: Réseau de Neurones Convolutif avec ses différentes couches. Source: [16]

Les hyper-paramètres

Les hyper-paramètres permettent de contrôler l'apprentissage automatique. Ils sont répartis en deux catégories :

- les hyperparamètres de modèles
- les hyperparamètres d'algorithme

Les **hyper-paramètres du modèle** sont principalement liés à la topologie et la taille du réseau de neurones, souvent prédéfinis par l'architecture de celui choisi (CNN, RNN, auto-encodeur). Généralement, ces paramètres sont peu ou pas changés, car l'architecture est souvent utilisée telle qu'elle est proposée par son auteur.

Les **hyper-paramètres d'algorithme** sont ceux sur lesquels on va le plus jouer pour contrôler la vitesse d'apprentissage du modèle. En deep learning, le modèle est encapsulé dans un algorithme d'apprentissage qui définit :

- le chargement des données en entrée du modèle,
- le déroulement de la phase d'entraînement,
- le déroulement de la phase de validation.

Pour le **chargement des données**, il y a :

- la taille du lot de données (en anglais **batch size**) fournit en entrée du réseau,
- la méthode de chargement des données (en anglais data loader) avec ou sans échantillonnage.

Pour les **phases d'entraînement et de validation**, il y a :

- le nombre d'itérations (en anglais **epoch**) qui définit le nombre de boucles d'apprentissage (i.e. entraînement-validation) que l'algorithme va réaliser pour permettre au modèle d'améliorer ses estimations
- la fonction de perte qui calcule la valeur de l'erreur existante entre l'estimation et l'observation
- l'optimiseur est la fonction d'optimisation utilisée pour la descente de gradient
- le taux d'apprentissage (en anglais **learning rate**) est le pas dans l'algorithme de descente de gradient.

Le **taux d'apprentissage** (ou le pas d'apprentissage) est le paramètre définissant la "vitesse" de la descente de gradient par l'algorithme d'optimisation. Le choix de sa valeur est difficile, car si elle est trop grande l'algorithme d'optimisation diverge, et si elle est trop petite la vitesse de convergence est trop faible.

Les différents **algorithmes d'optimisation** tendent tous à améliorer l'estimation du taux d'apprentissage en fonction des itérations passées de la descente de gradient. Dans ce projet de mémoire l'optimiseur qui est utilisé est l'optimiseur **ADAM**.

ADAM calcule le facteur multiplicateur du taux d'apprentissage par une combinaison de la moyenne des carrés et de la moyenne des moments d'ordre 2 (soit la variance) des gradients antérieurs.

1.2.3.5. Deep Learning Appliqué à la Reconnaissance Vocale

La plupart des applications audios d'apprentissage en profondeur utilisent des spectrogrammes pour représenter les audios. L'utilisation des spectrogrammes est un moyen efficace de capturer les caractéristiques essentielles des données audio sous forme d'image.

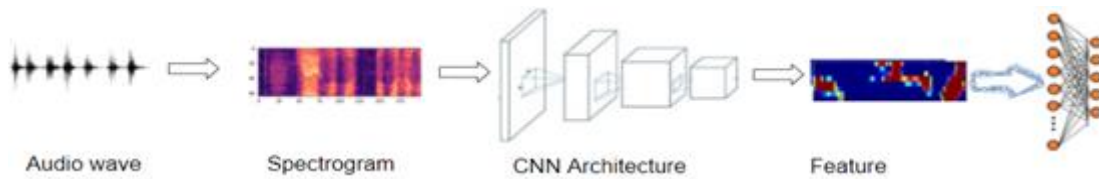


Figure 6: Étapes de la construction d'un modèle d'apprentissage audio en profondeur

Les étapes de la construction d'un modèle d'apprentissage audio en profondeur basé sur les images sont les suivantes :

- Commencer avec des données audio brutes sous la forme de fichiers wav
- Convertir les données audio en spectrogrammes correspondants
- Un nettoyage peut également être effectué sur les données audio brutes avant la conversion du spectrogramme.
- Utiliser des CNN pour les traiter, extraire des cartes de caractéristiques et créer le modèle.

L'étape suivante consiste à générer des prédictions de sortie à partir du modèle.

1.2.4. Son et Signal Sonore

Le son est une vibration mécanique d'un fluide, qui se propage sous forme d'ondes longitudinales grâce à la déformation élastique de ce fluide. Les êtres humains, comme beaucoup d'animaux, ressentent cette vibration grâce au sens de l'ouïe. Nous pouvons mesurer l'intensité des variations de pression et tracer ces mesures dans le temps. Les signaux sonores se répètent souvent à intervalles réguliers afin que chaque onde ait la même forme. La hauteur indique l'intensité du son et est appelée amplitude.

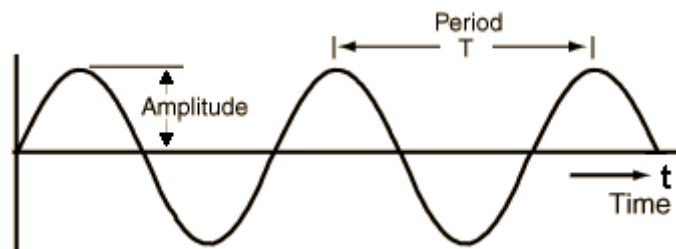


Figure 7: Signal répétitif simple montrant l'amplitude en fonction du temps. Source : [18]

Le temps mis par le signal pour terminer une onde complète est la période. Le nombre d'ondes produites par le signal en une seconde s'appelle la fréquence. La fréquence est l'inverse de la période. Tous les sons que nous entendons, y compris notre propre voix humaine, sont constitués de formes d'onde comme celles-ci.

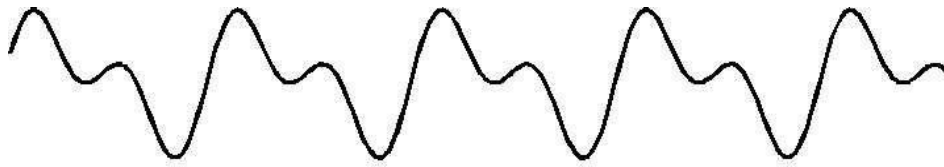


Figure 8: Forme d'onde musicale avec un signal répétitif complexe. Source : [19]

1.2.4.1. Représentation Numérique du Son

Pour numériser une onde sonore, nous devons transformer le signal en une série de nombres afin de pouvoir l'utiliser comme entrée dans nos modèles. Cela se fait en mesurant l'amplitude du son à des intervalles de temps fixes.

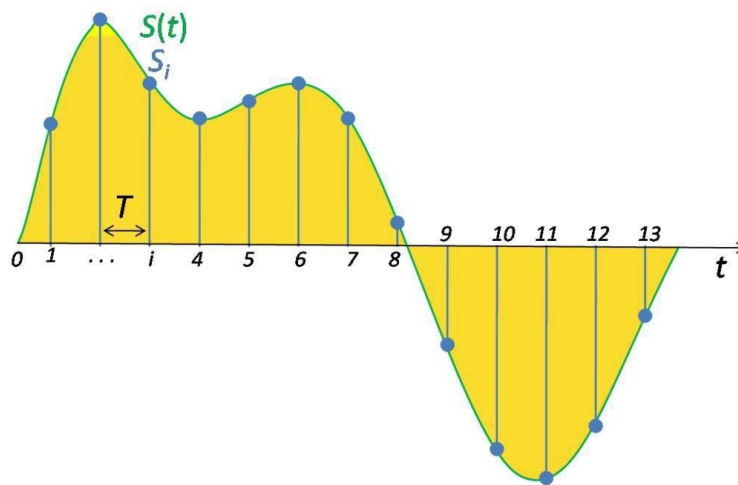


Figure 9: Mesures d'échantillons à intervalles de temps réguliers. Source : [20]

Chacune de ces mesures est appelée un échantillon et la fréquence d'échantillonnage est le nombre d'échantillons par seconde. Par exemple, un taux d'échantillonnage commun est d'environ 44 100 échantillons par seconde. Cela signifie qu'un audio de 10 secondes aurait 441 000 échantillons.

1.2.4.2. Parole Humaine

La parole est un flux continu constitué d'une suite de mots, eux-mêmes étant constitués d'un enchaînement de phonèmes et de bruits. Un phonème est l'unité distinctive de prononciation dans une langue.

- Exemple : /ε / et / ε: / dans père et paire

La parole humaine est très variable puisqu'un même phonème possède de nombreux paramètres qui varient suivant les locuteurs :

- Intensité de la voix
- Hauteur de la voix
- Type de son émis par le locuteur (chuchotement, chant, parole etc.)
- Emotion dans la voix du locuteur

1.2.4.3. Préparation des Données Audio pour un Modèle d'Apprentissage en Profondeur

Jusqu'à il y a quelques années, les applications d'apprentissage automatique de computer vision s'appuyaient sur des techniques traditionnelles de traitement d'images pour faire de l'ingénierie des fonctionnalités. Les algorithmes génèrent des caractéristiques et détectent les coins, les bords et les faces. Avec les applications NLP également, nous nous appuyerons sur des techniques telles que l'extraction de N-grammes et le calcul de la fréquence des termes.

Les applications d'apprentissage automatique audio dépendaient de techniques traditionnelles de traitement du signal numérique pour extraire des fonctionnalités. Par exemple, pour comprendre la parole humaine, les signaux audio étaient analysés à l'aide de concepts phonétiques pour extraire des éléments tels que des phonèmes. Tout cela nécessitait beaucoup d'expertise spécifique au domaine pour résoudre ces problèmes et régler le système pour de meilleures performances. Cependant, ces dernières années, alors que le deep Learning devient de plus en plus omniprésent, il a également connu un énorme succès dans la gestion de l'audio. Avec l'apprentissage en profondeur, les techniques de traitement audio traditionnelles ne sont plus nécessaires et on peut s'appuyer sur une préparation de données standard sans nécessiter beaucoup de génération manuelle et personnalisée de fonctionnalités.

Ce qui est plus intéressant, c'est qu'avec le deep Learning, on ne traite pas réellement les données audio sous leur forme brute. Au lieu de cela, l'approche courante utilisée consiste à convertir les données audio en images, puis à utiliser une architecture CNN standard pour traiter ces images. Cela se fait en générant des spectrogrammes à partir de l'audio.

Spectre

Comme on l'a expliqué précédemment, des signaux de fréquences différentes peuvent être additionnés pour créer des signaux composites, représentant n'importe quel son qui se produit dans le monde réel. Cela signifie que tout signal se compose de plusieurs fréquences distinctes et peut être exprimé comme la somme de ces fréquences. Le spectre est l'ensemble des fréquences qui sont combinées pour produire un signal par exemple la figure suivante montre le spectre d'un audio. Le spectre trace toutes les fréquences présentes dans le signal ainsi que la force ou l'amplitude de chaque fréquence.

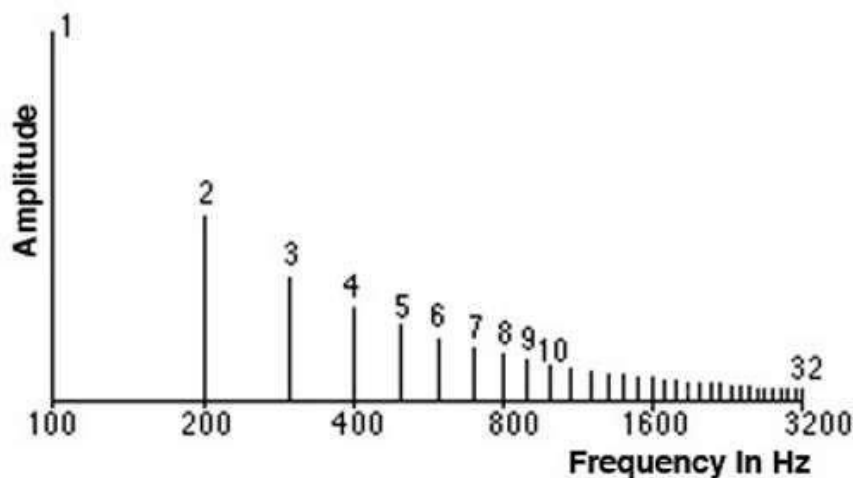


Figure 10: Spectre montrant les fréquences qui composent un signal sonore. Source : [21]

La fréquence la plus basse d'un signal est appelée fréquence fondamentale. Les fréquences qui sont des multiples entiers de la fréquence fondamentale sont appelées harmoniques. Par exemple, si la fréquence fondamentale est de 200 Hz, alors ses fréquences harmoniques sont de 400 Hz, 600 Hz, et ainsi de suite.

Domaine Temporel et Domaine Fréquentiel

Les formes d'onde que nous avons vues précédemment montrant l'amplitude par rapport au temps sont une façon de représenter un signal sonore. Étant donné que l'axe des x montre la plage de valeurs temporelles du signal, nous visualisons le signal dans le domaine temporel.

Le spectre est une autre façon de représenter le même signal. Il montre l'amplitude par rapport à la fréquence, et puisque l'axe des x montre la plage de valeurs de fréquence du signal, à un moment donné, nous visualisons le signal dans le domaine fréquentiel.

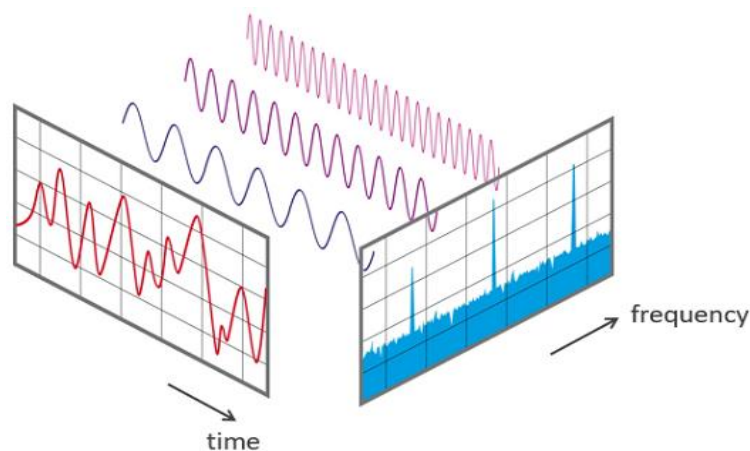


Figure 11: Domaine temporel et domaine fréquentiel. Source : [22]

Spectrogrammes

Le graphique qui affiche les données audio dans le domaine fréquentiel et temporel est appelé un spectrogramme.

Étant donné qu'un signal produit des sons différents au fur et à mesure qu'il varie dans le temps, ses fréquences constitutives varient également avec le temps. En d'autres termes, son spectre varie avec le temps.

Le spectrogramme d'un signal trace son spectre au fil du temps et ressemble à une photographie du signal. Il trace le temps sur l'axe des x et la fréquence sur l'axe des y. C'est comme si nous prenions le spectre encore et encore à différents moments dans le temps, puis les réunissons tous ensemble en une seule intrigue.

Il utilise différentes couleurs pour indiquer l'amplitude ou la force de chaque fréquence. Plus la couleur est brillante, plus l'énergie du signal est élevée. Chaque "tranche" verticale du spectrogramme est essentiellement le spectre du signal à cet instant et montre comment la force du signal est distribuée dans chaque fréquence trouvée dans le signal à cet instant.

Les spectrogrammes sont produits à l'aide de transformées de Fourier pour décomposer tout signal en ses fréquences constitutives.

Les figures ci-dessous présentent les spectrogrammes des mots Wolof « waaw » et « deideid » et le signal sonore de « waaw » pour des enregistrements de notre dataset.

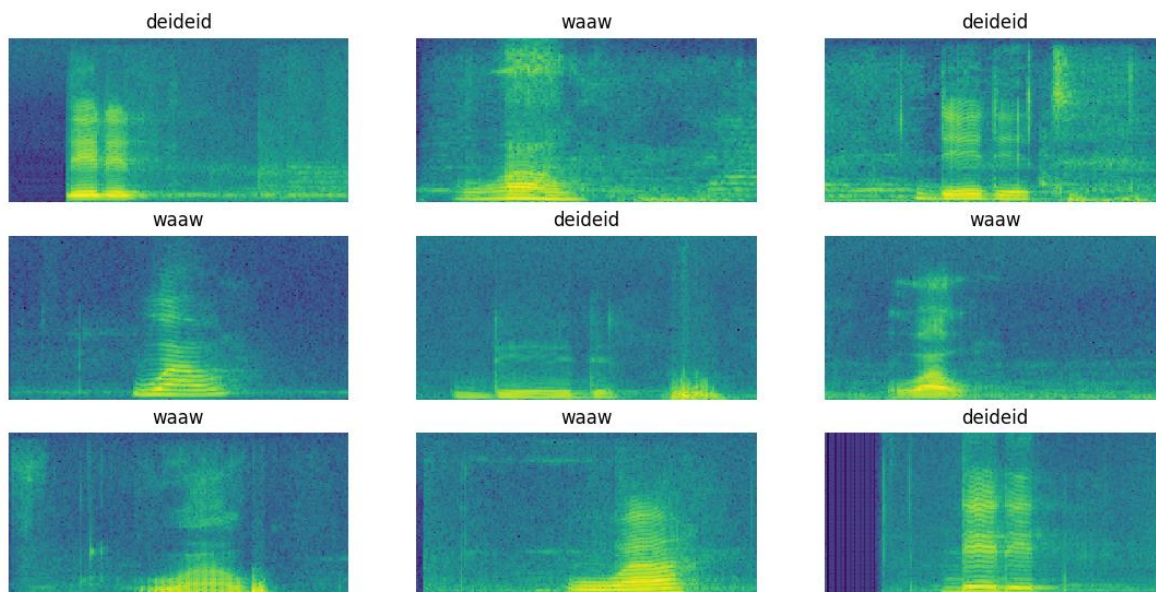


Figure 12: Spectrogrammes des mots Wolof « waaw » et « deideid ».

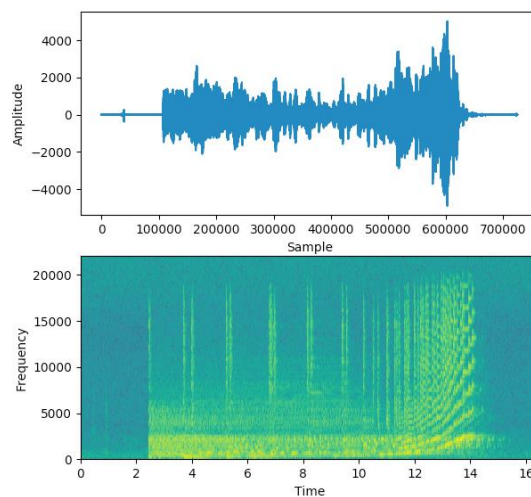

















Figure 13: Signal sonore du mot Wolof “waaw” et son spectrogramme.

CHAPITRE 2 : MISE EN OEUVRE

2.1. Outils Utilisés

Outil	Description	Utilisation
Python 	Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est le langage le plus utilisé pour la Science de données. Python permet également de faire du développement web avec des frameworks comme Django et Flask.	On a utilisé Python tout au long de l'implémentation de notre projet.
TensorFlow 	TensorFlow est une plateforme open source dédiée au machine learning développée par Google. Elle est particulièrement axée sur la formation et l'inférence de réseaux neuronaux profonds. TensorFlow peut fonctionner sur des systèmes CPU et GPU.	On a utilisé TensorFlow version 2.1 pour créer et entraîner notre modèle de reconnaissance vocale de mot Wolof
Keras 	Keras est une API de réseaux neuronaux de haut niveau écrite en Python et capable de s'exécuter sur TensorFlow. Elle a été développée dans le but de permettre une expérimentation rapide. Elle prend en charge à la fois les réseaux convolutifs et les réseaux récurrents, ainsi que les combinaisons des deux. Keras fonctionne de manière transparente sur des CPU et GPU.	Keras associée à TensorFlow nous a permis de créer notre modèle et de l'entraîner.
Scikit-learn 	Scikit-learn est une bibliothèque d'apprentissage statistique en Python. C'est le moteur de beaucoup d'applications de l'intelligence artificielle et de la science des données. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python notamment NumPy et SciPy.	Cette bibliothèque nous a permis de diviser nos données en ensemble de données d'entraînement, de validation et de test.
Numpy 	Numpy est une bibliothèque Python qui permet de manipuler les matrices ou tableaux multidimensionnels et les fonctions mathématiques opérant sur ces tableaux. Elle est libre et open source. Numpy fournit ndarray qui est un objet tableau homogène à n dimensions et des méthodes pour opérer efficacement dessus.	Nous avons utilisé cette bibliothèque pour convertir nos données en matrice afin qu'elles soient utilisables par le modèle.

 <p>Matplotlib</p>	<p>Matplotlib est une bibliothèque Python qui permet de tracer et de visualiser des données sous formes de graphiques. Il est libre et gratuit sous licence BSD.</p>	<p>Nous avons utilisé cette bibliothèque pour visualiser nos données sous plusieurs formes.</p>
 <p>Librosa</p>	<p>Librosa est une librairie Python pour le traitement de la musique et de l'audio. Il permet l'analyse et l'extraction des caractéristiques d'un signal audio.</p>	<p>Nous avons utilisé cette bibliothèque pour prétraiter nos données audios.</p>
 <p>PyAudio</p>	<p>Pyaudio est une librairie Python qui est liée à la technologie PortAudio (la bibliothèque d'E/S audio multiplateforme). Pyaudio permet d'utiliser facilement Python pour lire et enregistrer de l'audio sur diverses plateformes (Windows, Linux etc.).</p>	<p>Nous avons utilisé cette bibliothèque pour enregistrer de l'audio en ligne de commande.</p>
 <p>Flask</p>	<p>Flask est un framework d'application Web WSGI léger. Il est conçu pour rendre la mise en route rapide et facile, avec la possibilité de s'adapter à des applications complexes. Il a commencé comme un simple wrapper autour de Werkzeug et Jinja et est devenu l'un des frameworks d'applications Web Python les plus populaires.</p>	<p>Flask nous a permis de déployer notre modèle dans une plateforme web accessible pour les utilisateurs finaux du modèle.</p>
 <p>JSON</p>	<p>JavaScript Object Notation (JSON) est un format textuel de représentation de données basé sur la liste ordonnée et les paires clé-valeur. Il permet de stocker des données de manière logique et organisée. Il est également utilisé pour échanger des données entre les serveurs Web et les clients. Outre l'échange de données, il rend possible la migration de base de données, par exemple de JSON vers SQL.</p>	<p>Nous avons utilisé un fichier JSON pour y sauvegarder toutes les caractéristiques extraites des données audio lors de la phase de prétraitement de ces données. Nous avons également utilisé le format JSON pour envoyer des données du serveur au client.</p>
 <p>WhatsApp</p>	<p>WhatsApp est l'une des applications de messagerie les plus utilisées dans le monde. Il prend en charge l'envoi et la réception de divers fichiers média (photo, vidéo, audio), ainsi que la création de groupes.</p>	<p>Cette application nous a permis de collecter nos données de types audio dans plusieurs localités avec des locuteurs différents.</p>
 <p>Pycharm</p>	<p>JetBrains PyCharm est un environnement de développement de programmes en Python. C'est aussi un IDE Python pour la science des données et le développement web avec prise en charge intégrée pour Pandas, Numpy, Matplotlib et d'autres bibliothèques scientifiques, avec intelligence de code avancée, graphiques. Il est open source.</p>	<p>Nous avons utilisé cet environnement de développement pour implémenter tout ce qui concerne notre projet.</p>
 <p>Google Colab</p>	<p>Google Colab offre la possibilité d'exécuter du code écrit en langage Python directement depuis leur navigateur et d'éviter de devoir installer certaines bibliothèques Python directement sur leur ordinateur.</p>	<p>Nous avons utilisé Google Colab pour faire plusieurs notebook.</p>

 Audacity	Audacity est un éditeur audio libre, à la fois complet et simple d'utilisation. Il permet de manipuler les fichiers au format WAV, AIFF, OGG ou MP3 : suppression des silences, ajout d'écho ou d'effets spéciaux, suppression des bruits, mixage, etc. Grâce à l'éditeur intégré, on peut également copier, coller et assembler des extraits sonores pour créer des projets multipistes.	Nous avons utilisé ce logiciel lors du prétraitement de nos données de type audio.
 Fre:ac	Fre:ac (acronyme de Free Audio Converter) est un logiciel Open Source, en français, qui permet la conversion des musiques. Il est compatible avec Windows. Il convertit les formats les plus populaires et connus (MP3, M4A, WAV, WMA, FLAC, OGG).	Nous avons utilisé ce logiciel lors du prétraitement de nos données de type audio.

2.2. Implémentation du Système

Notre but principal était de mettre en place un système (modèle) de reconnaissance vocale de mots Wolof et de l'intégrer dans une plateforme web qui permet aux personnes d'être informées sur la COVID-19 et orientées en fonction de leurs réponses.

Cette plateforme est composée d'une partie qui présente une série de questions auxquelles l'utilisateur doit répondre oralement en Wolof et d'une partie qui fait de la reconnaissance vocale des réponses entrées par l'utilisateur. L'utilisateur ne peut répondre que par oui ou non en Wolof. A la fin de chaque auto-diagnostic, le système oriente et informe l'utilisateur sur sa susceptibilité d'être atteint ou non de la COVID-19 et sur les actions à prendre. La plateforme dispose également d'une autre partie permettant de s'informer sur la COVID-19 en langue Wolof.

Nous avons utilisé le deep learning associé aux réseaux de neurones convolutifs pour mettre en place le système de reconnaissance vocale. Ainsi, il est nécessaire d'avoir un jeu de données avec lequel le réseau de neurones à convolution sera entraîné. Pour ce faire, nous avons créé notre propre jeu de données en enregistrant des personnes prononçant les mots Wolof « waaw » et « deideid ».

2.2.1. Vue d'Ensemble du Système

L'ensemble des fichiers audio recueillis ont été prétraités. Le modèle a été entraîné et optimisé sur les données. A la fin de l'entraînement, le modèle est utilisé faire des prédictions sur de nouvelles audio.

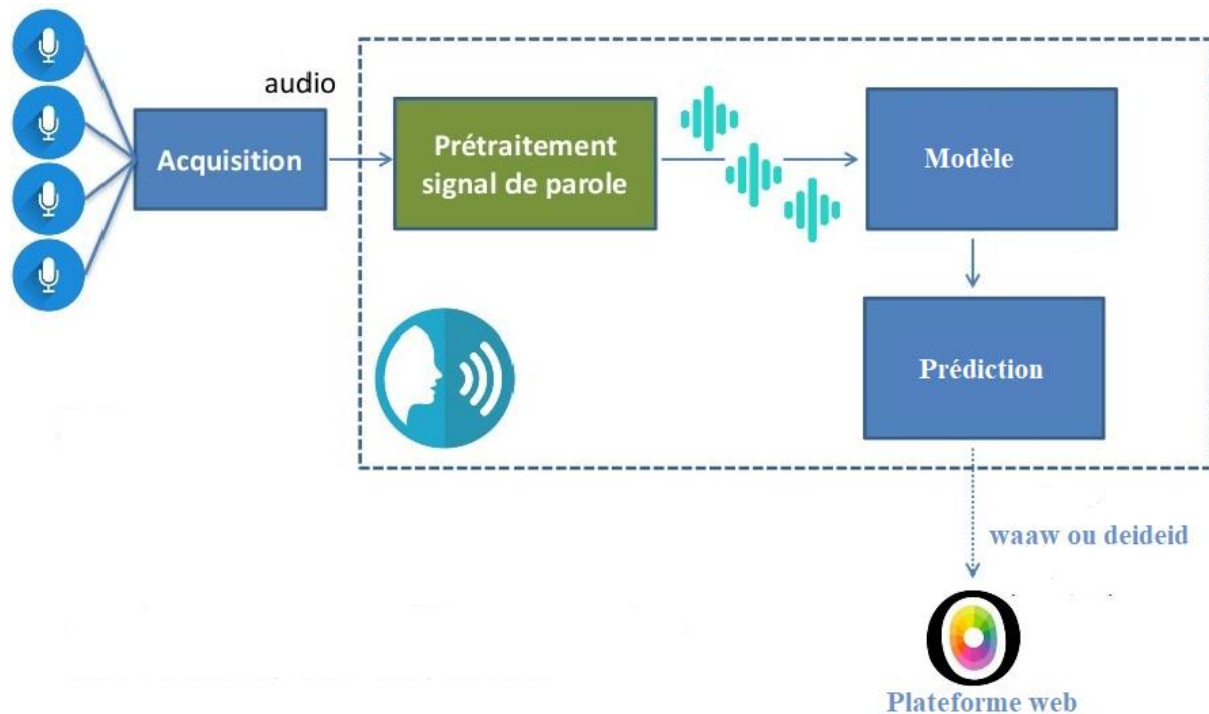


Figure 14: Architecture des différentes étapes de l'implémentation

2.2.2. Description des Données

L'ensemble de données sur lequel nous avons travaillé contient 310 enregistrements d'hommes, de femmes et d'enfants prononçant les mots Wolof "waaw" et "deideid". Nous avons utilisé l'application de messagerie WhatsApp pour collecter ces données audios. Au total, nous avons obtenu 155 fichiers audio pour chaque classe. Chaque enregistrement dure au maximum 1 seconde. Les personnes qui ont participé à la conception de mon dataset habitent dans les localités comme Touba, Saint-Louis, Dakar et dans d'autres villes du Sénégal. Par conséquent, les manières de prononcer les mots wolof "waaw" et "deideid" sont très variées. De plus, il y a différents bruits de fond dans les enregistrements. Ces conditions permettent au modèle d'être plus réaliste et généralisé.

2.2.3. Prétraitement des Données

2.2.3.1. Explication de la Méthodologie Utilisée

Tout projet d'apprentissage automatique comporte deux étapes principales : l'extraction des caractéristiques des données et l'entraînement du modèle.

Pour l'**extraction de caractéristiques audio** à des fins d'apprentissage automatique, des coefficients cepstraux à fréquence mel (MFCC) sont généralement extraits de l'audio et ces caractéristiques sont utilisées pour entraîner le modèle. L'extraction de caractéristiques MFCC est un moyen d'extraire uniquement les informations pertinentes qui sont les propriétés phonétiquement vitales du signal vocal de l'audio.

Pour mieux expliquer cela, lorsque nous représentons un fichier audio au format numérique, l'ordinateur le considère comme une onde avec l'axe x comme temps et l'axe y comme amplitude. Un exemple est fourni dans la figure ci-dessous.

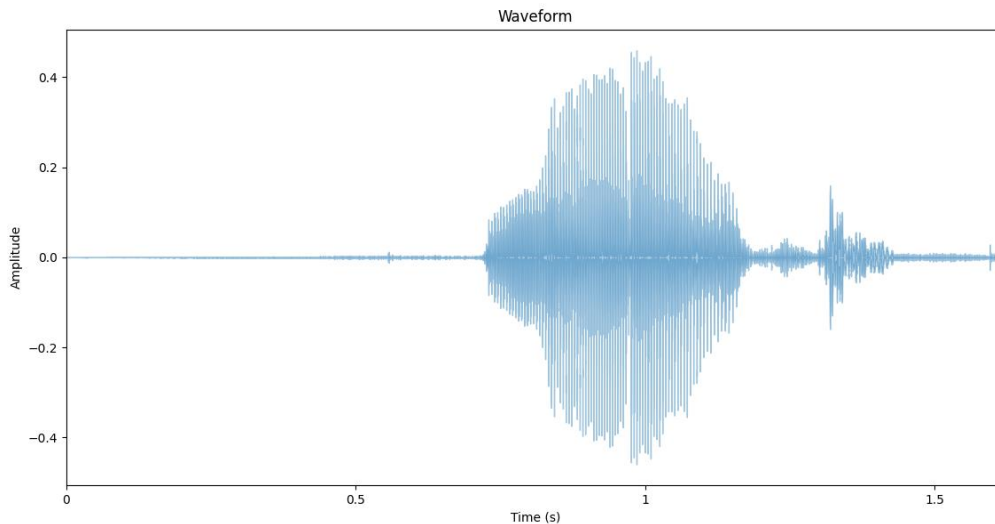


Figure 15: Waveform du mot Wolof « waaw ».

Ce format de représentation ne nous donne pas beaucoup d'informations sur l'audio, c'est pourquoi nous représentons l'audio dans le domaine fréquentiel, en utilisant une transformation de Fourier (**FFT**, Fast Fourier Transform). La FFT est un algorithme mathématique qui est utilisé pour convertir le domaine temporel en domaine fréquentiel.

En utilisant cette FFT, nous convertissons notre fichier audio et le représentons dans le domaine fréquentiel et temporel sous forme de spectrogramme.

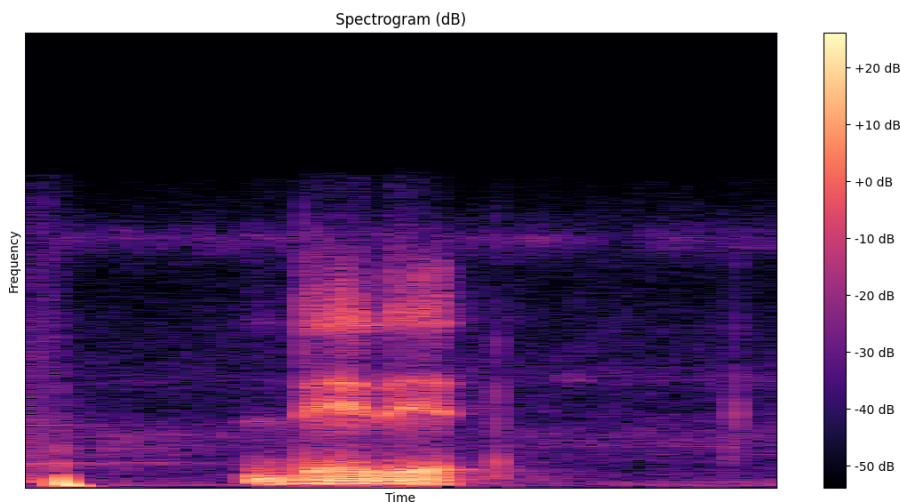


Figure 16: Spectrogramme du mot Wolof « deideid ».

MFCC est une technique conçue pour extraire des caractéristiques d'un signal audio. Il utilise l'échelle mel pour diviser les bandes de fréquences du signal audio, puis extrait les coefficients de chaque bande de fréquences individuelle, créant ainsi une séparation entre les fréquences. MFCC utilise la transformée en cosinus discrète (DCT) pour effectuer cette opération. L'échelle mel est établie sur la perception humaine du son, c'est-à-dire sur la façon dont le cerveau humain traite les signaux audio et différencie les différentes fréquences. Le MFCC utilise donc une échelle mel pour extraire les caractéristiques d'un signal audio, qui, lorsqu'il est représenté sous forme de graphique, s'avère être un spectrogramme mel. Donc, en un mot, ce qu'on voit sur un spectrogramme mel sont les caractéristiques exactes dont nous avons besoin pour entraîner notre modèle.

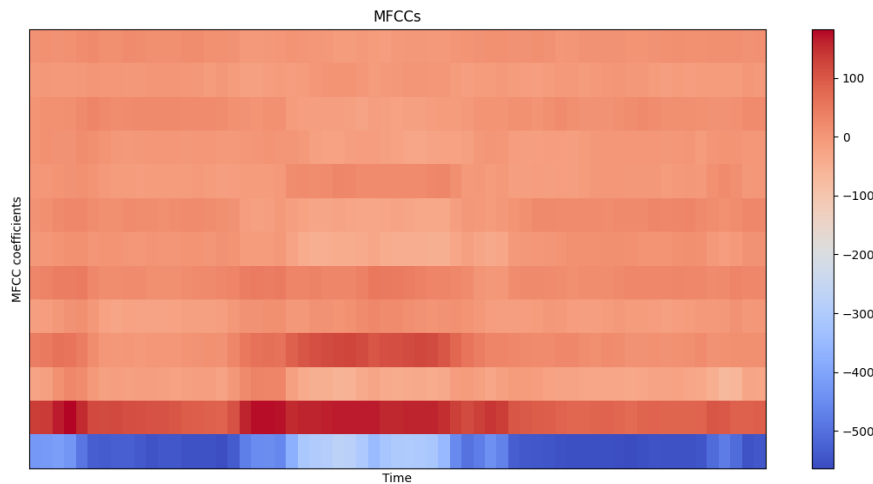


Figure 17: MFCC du mot Wolof « deideid ».

Après cette préparation des données, nous pouvons entraîner notre modèle en utilisant un CNN.

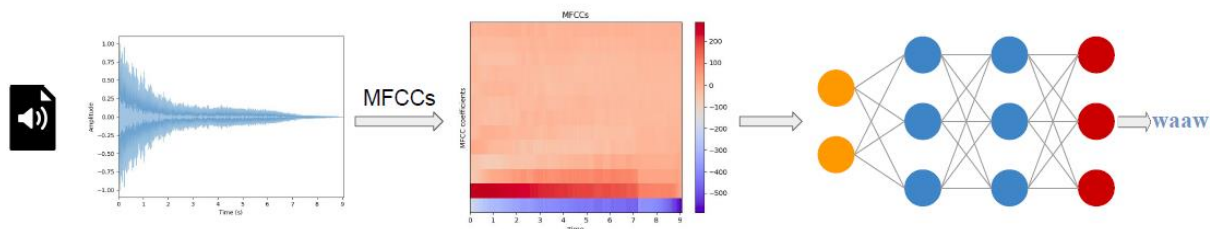


Figure 18: Du fichier audio à la prédiction du mot Wolof

2.2.3.2. Étapes de Prétraitement des Données

Les fichiers audio que l'on a collectés sont souvent au format opus, aac, m4a et autres, alors que pour le traitement et la manipulation des données audios, on utilise que des fichiers au format wav. Nous avons utilisé le logiciel Fre:ac convertir les fichiers audio collectés au format wav.

Nous avons utilisé le logiciel Audacity pour les normaliser les enregistrements de plus d'une seconde.

Nous avons ensuite extrait les caractéristiques qui conviendront pour alimenter notre réseau à l'aide de MFCC. Librosa est utilisé pour extraire les caractéristiques de chacun des segments audio. Nous avons créé un dictionnaire avec l'étiquette ou la catégorie du mot wolof comme clé et toutes les caractéristiques extraites des 155 fichiers audio comme un tableau de caractéristiques sous cette étiquette. Une fois que l'on a fait cela en boucle pour les 2 catégories, le dictionnaire est vidé dans un fichier JSON. Ce fichier JSON devient ainsi le jeu de données sur lequel le modèle sera entraîné.

L'utilisation de Librosa est illustrée dans l'extrait de code ci-dessous.

```
# extraire les MFCCs
MFCCs = librosa.feature.mfcc
```

Nous passons ensuite au codage pour le prétraitement de l'ensemble de données. Pour cela, nous avons d'abord défini la fréquence d'échantillonnage de chaque audio. La fréquence d'échantillonnage est nécessaire pour connaître la vitesse de lecture d'un son. Nous avons gardé

22050 pour chaque son. Nous avons choisi cette valeur car Librosa utilise 22050 Hz comme taux d'échantillonnage lors du chargement des données audios.

```
DATASET_PATH = "dataset"
JSON_PATH = "data.json"
SAMPLES_TO_CONSIDER = 22050
```

Nous avons ensuite créé une boucle dans laquelle nous ouvrons chaque fichier audio de chaque dossier de mot Wolof. Nous extrayons ensuite les fonctionnalités MFCC pour chacun de ces audios et les ajoutons au dictionnaire sous le nom du mot Wolof (qui est également le nom du dossier).

Ainsi le script Python extrait les caractéristiques et les vide dans le fichier **data.json**. Cette opération est illustrée dans l'extrait de code ci-dessous.

```
# enregistrement des données dans un fichier json
with open(json_path, "w") as fp:
    json.dump(data, fp, indent=4)
```

2.2.4. Création et Entraînement du modèle

Avant de construire le modèle, nous devons charger les données dans le programme et les diviser en ensemble d'entraînement, de validation et de test. Cela se fait en ouvrant le fichier JSON et en le convertissant en tableaux NumPy. Cette opération est illustrée dans l'extrait de code ci-dessous.

```
def load_data(data_path):
    with open(data_path, "r") as fp:
        data = json.load(fp)

    X = np.array(data["MFCCs"])
    y = np.array(data["labels"])
    print("Données chargées!")
    return X, y
```

Après avoir chargé les données, nous préparons les données et les séparons en ensembles d'entraînement, de validation et de test. Ceci est fait en utilisant la fonction **train_test_split** de **sklearn**. Cette opération est illustrée dans l'extrait de code ci-dessous.

```
def prepare_dataset(data_path, test_size=0.2, validation_size=0.2):
    X, y = load_data(data_path)

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size)
    X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train, test_size=validation_size)

    X_train = X_train[..., np.newaxis]
    X_test = X_test[..., np.newaxis]
    X_validation = X_validation[..., np.newaxis]

    return X_train, y_train, X_validation, y_validation, X_test, y_test
```

Reconnaissance de mots wolof à l'aide de CNN : Le cas d'une plateforme d'autodiagnostic COVID-19

Le réseau CNN est créé à l'aide de TensorFlow et Kera. Nous avons choisi de construire un modèle **Sequential** comme point de départ. L'extrait de code suivant montre la création du réseau de neurones.

```
def build_model(input_shape, loss="sparse_categorical_crossentropy", learning_rate=0.0001):  
  
    model = tf.keras.models.Sequential()
```

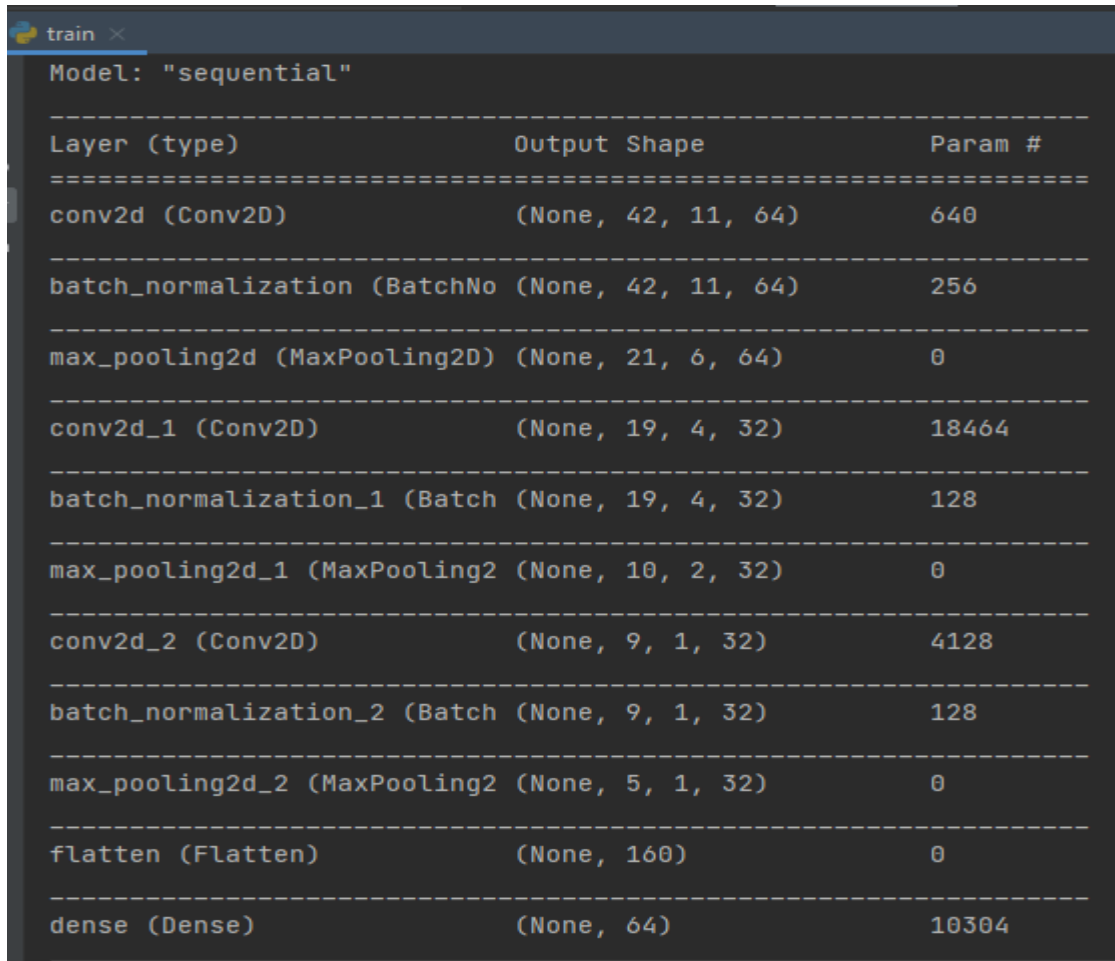
Des couches de **Convolution** avec **ReLU** comme fonction d'activation sont ajoutés sur le modèle de départ Sequential. Les autres couches ajoutées sont **BatchNormalization** pour normaliser les entrées, **MaxPooling** pour réduire l'échantillonnage et **Dropout** pour éviter le surajustement. Une couche **Flatten** et une dernière couche **Dense** de 2 neurones, une pour chaque classe (waaw, deideid) sont enfin ajoutées. Nous utilisons **Softmax** comme fonction d'activation de la dernière couche.

Les couches sont ajoutées avec la fonction **add()**. La taille de la couche d'entrée dépend de la taille du coefficient MFCC que l'on passe en argument « **input_shape** ».

L'ajout des couches du modèle est décrit dans l'extrait de code ci-dessous.

```
model.add(tf.keras.layers.Conv2D(64, (3, 3), activation='relu', input_shape=input_shape,  
                                kernel_regularizer=tf.keras.regularizers.l2(0.001)))  
model.add(tf.keras.layers.BatchNormalization())  
model.add(tf.keras.layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same'))  
model.add(tf.keras.layers.Conv2D(32, (3, 3), activation='relu',  
                                kernel_regularizer=tf.keras.regularizers.l2(0.001)))  
model.add(tf.keras.layers.BatchNormalization())  
model.add(tf.keras.layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same'))  
model.add(tf.keras.layers.Conv2D(32, (2, 2), activation='relu',  
                                kernel_regularizer=tf.keras.regularizers.l2(0.001)))  
model.add(tf.keras.layers.BatchNormalization())  
model.add(tf.keras.layers.MaxPooling2D((2, 2), strides=(2, 2), padding='same'))  
model.add(tf.keras.layers.Flatten())  
model.add(tf.keras.layers.Dense(64, activation='relu'))  
tf.keras.layers.Dropout(0.3)  
model.add(tf.keras.layers.Dense(2, activation='softmax'))
```

La figure suivante montre l'architecture de notre modèle avec ses différentes couches obtenues à la suite de l'appel de la fonction **summary()**.



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 42, 11, 64)	640
batch_normalization (Batch Normalization)	(None, 42, 11, 64)	256
max_pooling2d (MaxPooling2D)	(None, 21, 6, 64)	0
conv2d_1 (Conv2D)	(None, 19, 4, 32)	18464
batch_normalization_1 (Batch Normalization)	(None, 19, 4, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 10, 2, 32)	0
conv2d_2 (Conv2D)	(None, 9, 1, 32)	4128
batch_normalization_2 (Batch Normalization)	(None, 9, 1, 32)	128
max_pooling2d_2 (MaxPooling2D)	(None, 5, 1, 32)	0
flatten (Flatten)	(None, 160)	0
dense (Dense)	(None, 64)	10304

Figure 19: Architecture du Modèle.

Nous pouvons maintenant passer à la phase d'entraînement du modèle de classification.

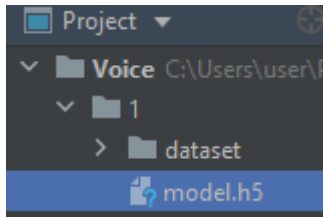
La fonction "**prepare_datasets**" est utilisée avec un pourcentage représentant les données de test et un pourcentage représentant les données de validation. Les données de validation sont une partie des données d'entraînement, avec lesquelles le modèle n'est pas entraîné et sont utilisées pour valider le modèle.

La fonction « **build_model** » construit le réseau CNN et le compile. La compilation permet d'ajouter l'optimiseur (qui définit le taux d'apprentissage) et la fonction de calcul des pertes. Nous avons utilisé la fonction mathématique catégorique d'entropie croisée.

Après la compilation, **model.fit ()** est utilisé pour entraîner le modèle sur les données.

```
# entraînement du modèle
history = model.fit(X_train, y_train, pochs=epochs, batch_size=batch_size,
                    validation_data=(X_validation, y_validation),
                    callbacks=[earlystop_callback])
```

Après l'entraînement, on enregistre le modèle dans le fichier **model.h5** (Hierarchical Data Format) afin de pouvoir l'utiliser dans notre plateforme pour prédire de nouvelles données.



Après entraînement, nous obtenons un modèle qui a atteint 96.72% de précision et 0.140 de perte. Les figures qui suivent représentent les courbes montrant la précision et la perte pour l'entraînement et la validation.

```
Test loss: 0.14045727252960205, test accuracy: 96.72130942344666
```

```
Process finished with exit code 0
```

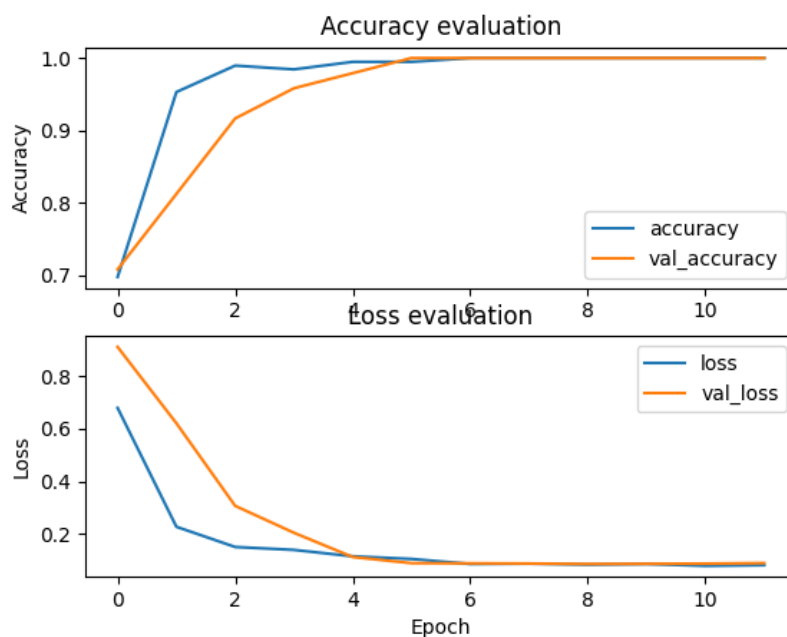


Figure 20: Courbes d'entraînement et de validation

Notre modèle est évalué sur l'ensemble de données d'entraînement et sur un ensemble de données de validation. Ici on remarque que la précision (accuracy) pendant l'entraînement et la validation est assez élevée alors que la perte lors de l'entraînement et la validation est assez faible. De plus, en visualisant les courbes, nous observons que notre modèle n'est pas surentraîné. En effet le surentraînement survient lorsque la perte pour l'entraînement continue de diminuer alors que la perte pour la validation est en hausse et la précision est plus élevée pour la phase d'entraînement que pour la phase de validation. Par conséquent on a un modèle qui généralise, c'est-à-dire, le modèle est capable de faire de bonnes prédictions sur de nouvelles données.

🚦 Les hyperparamètres Choisis:

Nous avons expérimenté avec les hyperparamètres et, après certains nombres d'optimisations, nous avons fini par conserver les hyperparamètres suivant : batch size= 20, nombre d'époques= 40, learning rate= 0,001 et comme optimiser Adam. En effet, le réglage de ces hyperparamètres nous a permis d'obtenir le meilleur modèle de précision et perte.

```
DATA_PATH = "data.json"
SAVED_MODEL_PATH = "model.h5"
EPOCHS = 40
BATCH_SIZE = 20
PATIENCE = 5
LEARNING_RATE = 0.001
```

2.2.5. Réaliser une Prédiction

Nous pouvons enfin réaliser des prédictions sur de nouveaux audios.

Nous configurons les constantes suivantes.

```
SAVED_MODEL_PATH = "model.h5"
SAMPLES_TO_CONSIDER = 22050
```

Nous lançons un serveur créé à l'aide de Flask.

```
2021-10-04 19:54:09.520231: I tensorflow/compiler/mlir/mlir_graph_optimization_pass
.cc:176] None of the MLIR Optimization Passes are enabled (registered 2)
127.0.0.1 - - [04/Oct/2021 19:54:09] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [04/Oct/2021 19:56:39] "POST /predict HTTP/1.1" 200 -
```

Nous pouvons faire des prédictions sur de l'audio enregistré à partir d'un microphone.

```
(venv) C:\Users\user\Desktop\MyVoice\1>python client.py
Predicted keyword: deideid

(venv) C:\Users\user\Desktop\MyVoice\1>
```

2.2.6. Validation du Modèle

Nous devons valider le modèle en observant comment il se comporte lorsque l'on prononce les mots Wolof « waaw » et « deideid » directement dans le microphone de l'ordinateur. PyAudio enregistre de nouveaux extraits audio inconnus à notre modèle. On observe ensuite les prédictions faites par le modèle.

Nous avons validé notre modèle après un certain nombre de tests avec différents locuteurs et l'obtention de bonnes prédictions

```
(venv) C:\Users\user\Desktop\MyVoice\1>python client.py
Enregistrement avec PyAudio...
Arreter l'enregistrement
Mot Prédit: deideid

(venv) C:\Users\user\Desktop\MyVoice\1>python client.py
Enregistrement avec PyAudio...
Arreter l'enregistrement
Mot Prédit: waaw
```


CHAPITRE 3 : DÉPLOIEMENT DU MODÈLE

Dans ce dernier chapitre, nous allons présenter l'architecture client-serveur de notre système d'autodiagnostic de la COVID-19.

3.1. Architecture Client-Serveur du Système

La figure suivante donne une vision de notre architecture.

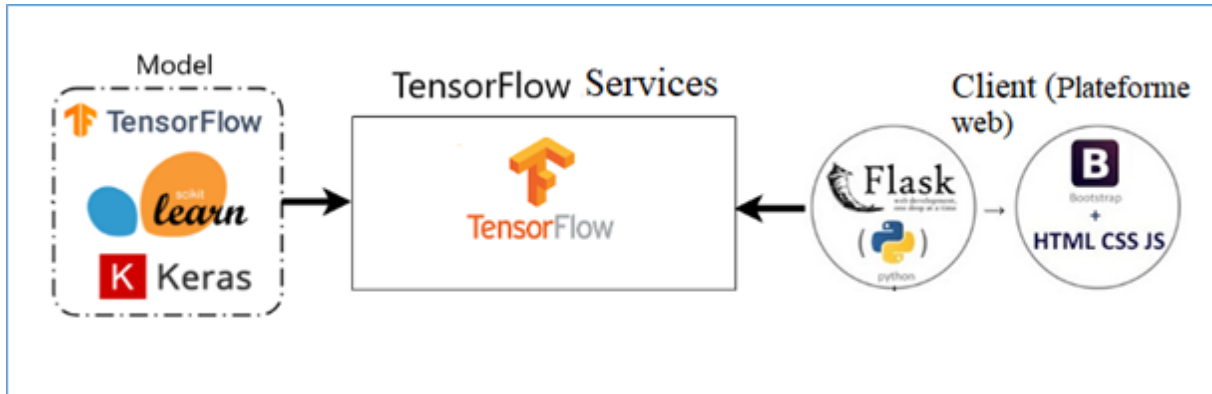


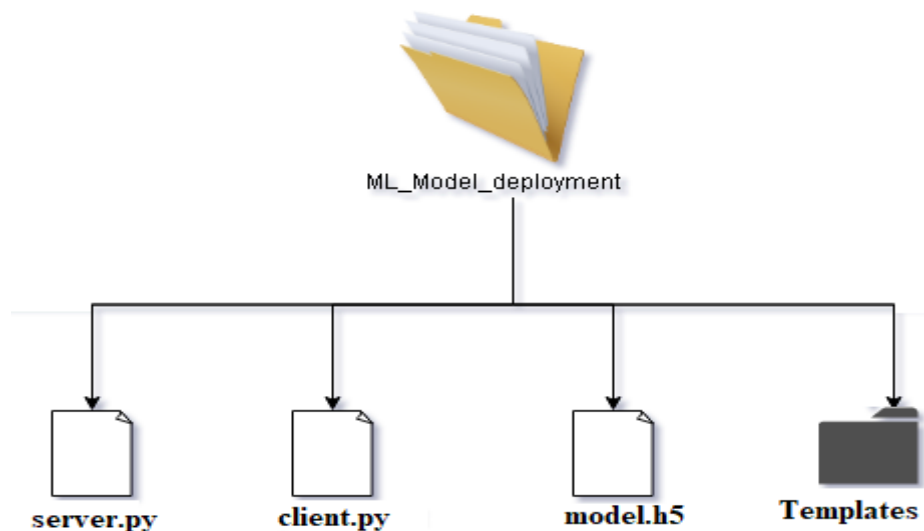
Figure 21: Architecture client-serveur du Projet

- TensorFlow associée à Kera et Scikit Learn: comme expliqué précédemment, ces bibliothèques sont utilisées dans notre projet pour la création et l'entraînement du modèle de deep learning.
- Flask: est utilisé pour déployer le modèle créé.
- Client web: est un navigateur web qui envoie des requêtes au serveur et reçoit des réponses venant du serveur.

Le micro-framework web Flask dédié à Python a été instrumental dans l'implémentation du système. Le web rend le système accessible à l'utilisateur final.

- **Structure du Répertoire de notre Projet**

La figure ci-dessous illustre l'organisation du répertoire de notre projet et donne une idée de l'étendue du projet.



- **Templates:** Ce dossier contient les fichiers html qui sont utilisés par notre fichier principal (server.py) pour générer les interfaces de notre plateforme.
- **server.py:** Ce fichier contient toutes les fonctionnalités importantes comme celle de prédiction. Il lie les différents fichiers et répertoires.
- **model.h5:** Ce fichier contient le modèle.
- **client.py:** Ce fichier permet de faire des tests en ligne de commande.

Le serveur Flask est créé comme décrit ci-dessous.

```
import random
import os
from flask import Flask, request, jsonify
from keyword_spotting_service import Keyword_Spotting_Service

app = Flask(__name__)
```

La capture ci-dessous montre notre serveur démarré.

```
server x
* Serving Flask app 'server' (lazy l
* Environment: production
WARNING: This is a development ser
Use a production WSGI server inste
* Debug mode: off
* Running on http://127.0.0.1:5000/
```

Le fichier server.py contient la fonction **index** qui permet de visualiser le fichier index.html et, donc, l'application web.

```
@app.route('/accueil/')
def index():
    return render_template('index.html')
```

3.2. Interfaces de la Plateforme Web

Reconnaissance de mots wolof à l'aide de CNN : Le cas d'une plateforme d'autodiagnostic COVID-19

La figure suivante représente l'interface qui permet à un utilisateur d'enregistrer un vocal. Cet audio sera la réponse à la première question, à savoir « Voulez-vous faire un autodiagnostic de la COVID-19 ? » en wolof. Ensuite, lorsque l'utilisateur validera sa réponse, celle-ci va être envoyée au serveur qui utilisera le modèle pour prédire la réponse de l'utilisateur (« waaw » ou « deideid »). La prédiction finale est retournée au système.

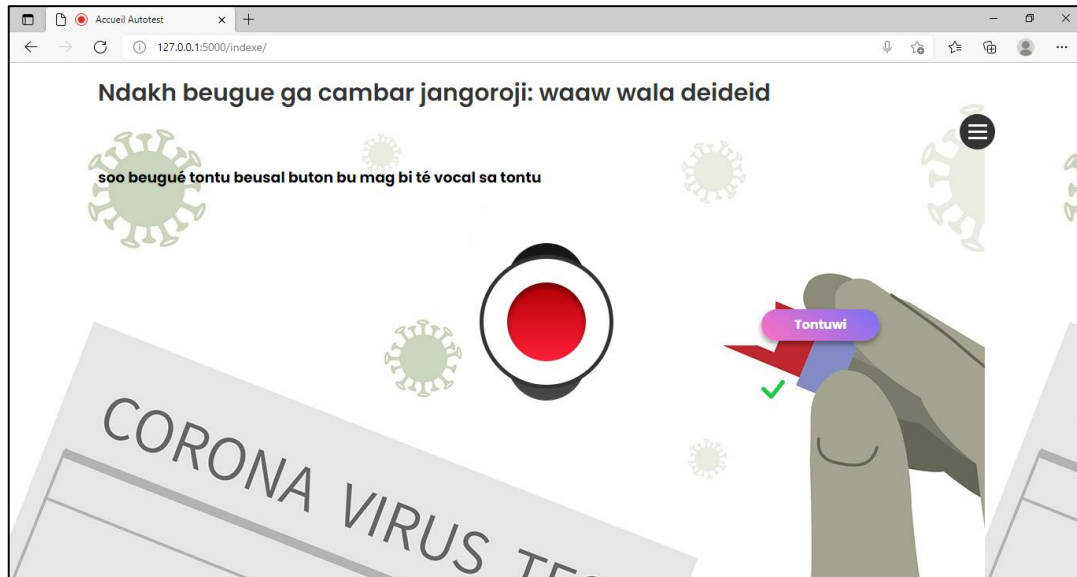


Figure 22 : Interface web qui permet la reconnaissance vocale

Le bouton rouge montre que l'utilisateur est en train d'enregistrer un audio. A la suite de l'enregistrement audio, si le modèle prédit le mot Wolof « deideid », il ne se passera rien. La page affichera seulement sa réponse (« **deideid** »).

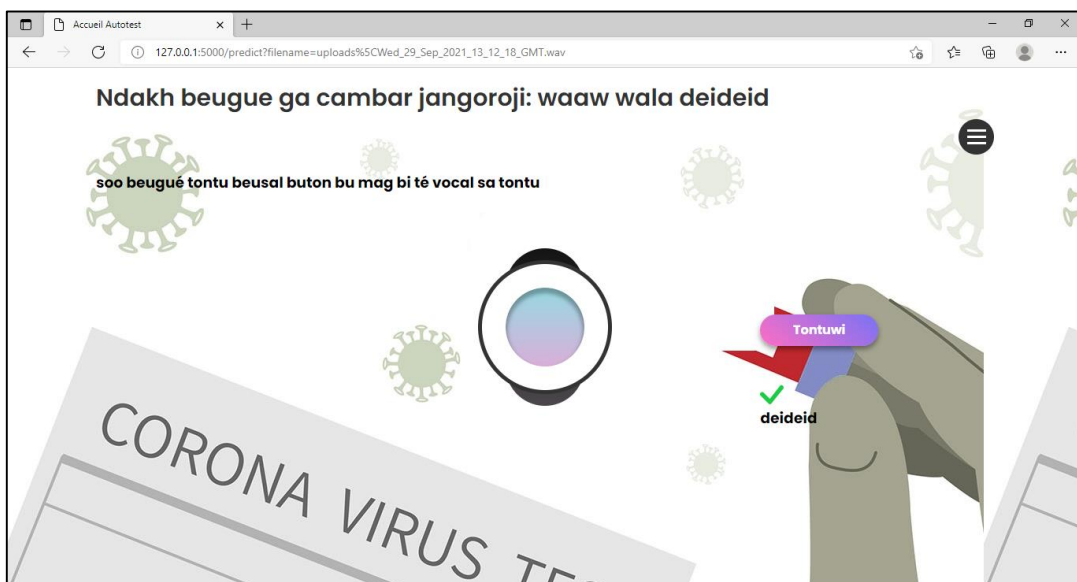


Figure 23: Cas où la réponse est "deideid"

Si le modèle prédit par contre le mot Wolof « waaw », dans ce cas l'utilisateur va être redirigé vers la page d'accueil pour l'autodiagnostic de la COVID-19. Cette page va permettre de démarrer l'autodiagnostic.

Reconnaissance de mots wolof à l'aide de CNN : Le cas d'une plateforme d'autodiagnostic COVID-19

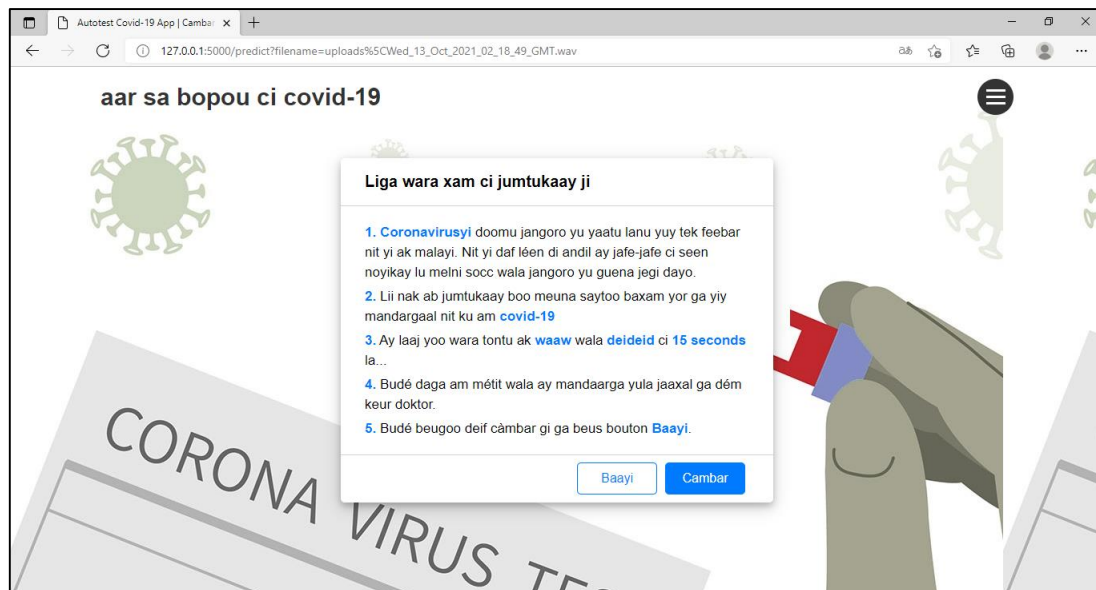


Figure 24: Page d'accueil pour démarrer l'autodiagnostic

L'autodiagnostic est constitué d'une série de questions auxquelles l'utilisateur devra répondre par les mots wolof « waaw » ou « deideid ». On s'est inspiré des questions qu'utilise la plateforme « AlloCovid » utilisée en France pour faire un autodiagnostic contre la COVID-19. Nous les avons traduites en Wolof pour notre cas d'utilisation. Voici un tableau qui contient l'ensemble des questions concernant l'autodiagnostic en Français et en Wolof.

Questions de AlloCovid en Français	Questions en Wolof
Avez-vous de la fièvre ?	Ndax sa yaram dafa tang
Avez-vous une toux?	Ndax daga socc wala dangay seuxeut ak di tissli
Avez-vous noté une perte ou une forte diminution de votre goût ou de votre odorat ces derniers jours ?	Ci fann yu muju yi ndax booy lék dafay léwét ba doo xam tiafka gi
Avez-vous un mal de gorge ou des maux de tête inhabituels ces derniers jours ?	Ndax daga am méti put wala méti bopu ci fann yu muju yi
Avez-vous de la diarrhée ces dernières 24 heures?	Ndax daga am bir bouy daww ci fann yu muju yi
Avez-vous une fatigue inhabituelle ces derniers jours ?	Ndax daga am yaram buy méti wala yaram bu dis ci fann yu muju yi
Cette fatigue vous oblige-t-elle à vous reposer plus de la moitié de la journée ?	Ndax daga am yaram buy méti bulay teural bisbi yeup
Êtes-vous dans l'impossibilité de vous alimenter ou de boire depuis 24 heures ou plus?	Ndax daga am jaffé-jaffé lék wala naan ci fann yu muju yi
Dans les dernières 24 heures, avez-vous noté un manque de souffle inhabituel lorsque vous parlez ou faites un petit effort ?	Ndax dagay yeuk coono ci sa yaram sooy wax wala yeugatu ci fann yu muju yi
Avez-vous une maladie respiratoire chronique (bronchopneumopathie obstructive, asthme sévère) ou la maladie de drépanocytose?	Ndax am ga jangoroju andak jaffé-jaffé nooyi wala jangoroju mélni cancer wala diabète wala drépanocytose

Reconnaissance de mots wolof à l'aide de CNN : Le cas d'une plateforme d'autodiagnostic COVID-19

La figure suivante montre la façon dont les questions sont présentées, ici, la question est : Avez-vous de la fièvre ?

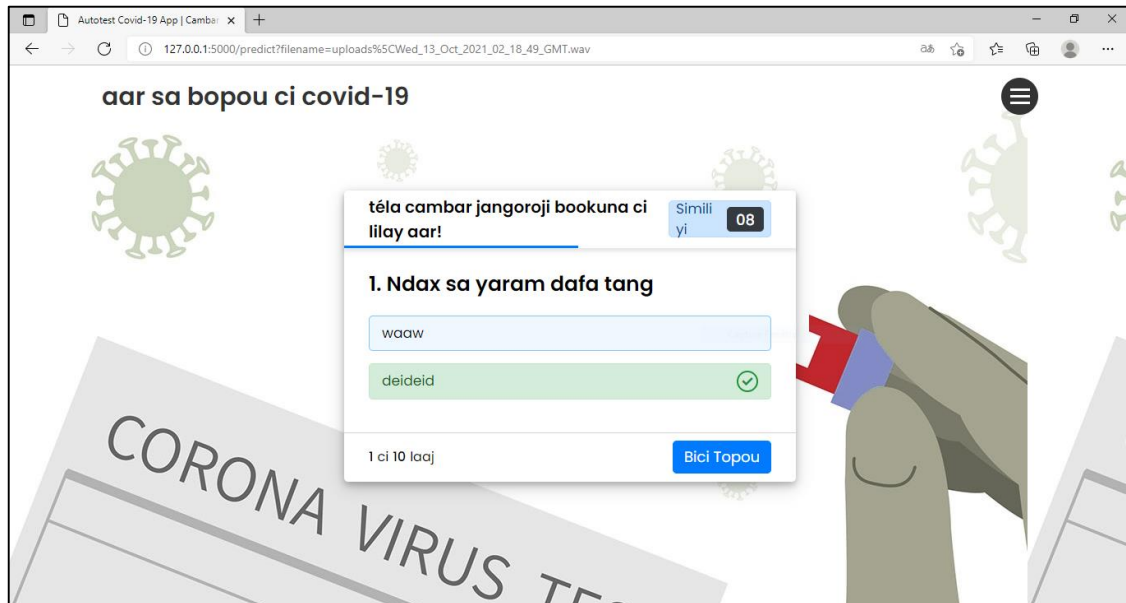


Figure 25: La première question de l'autodiagnostic

A la fin de la série de questions réponses, le système informe et oriente l'utilisateur sur sa susceptibilité d'être infecté par le virus ou non et sur les actions à prendre en cas d'infection. La figure suivante illustre un type de conclusion du système après autodiagnostic.

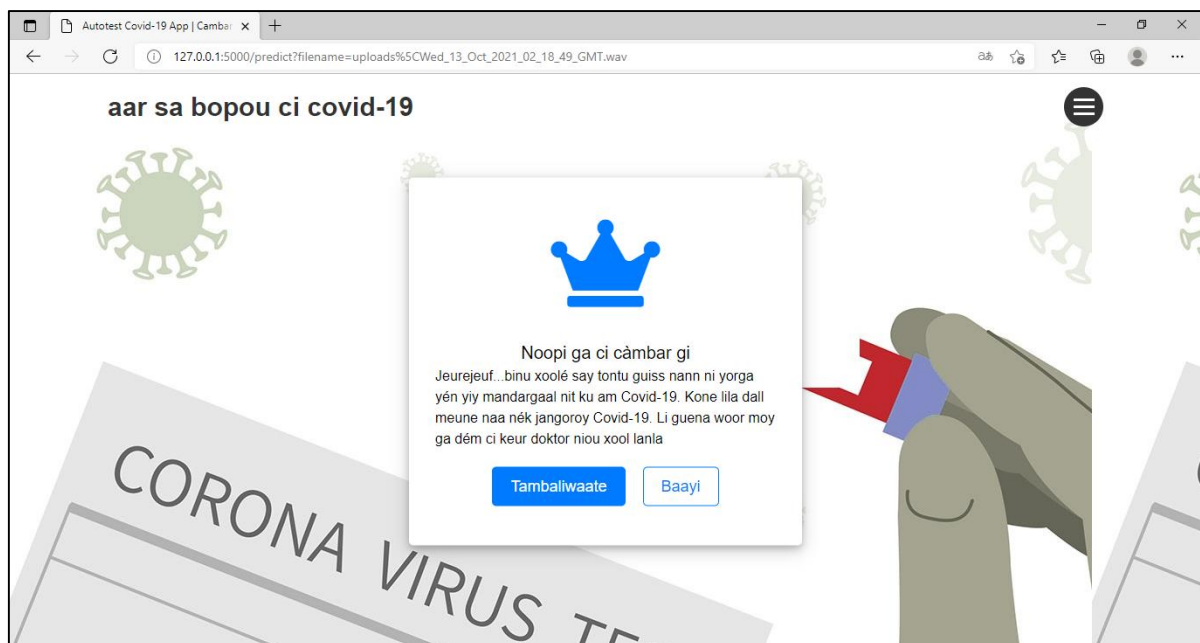


Figure 26: Type d'orientation

CONCLUSION GÉNÉRALE

Le but principal de ce mémoire était de mettre en place un système de reconnaissance vocale de mots Wolof. Pour y arriver, nous avons d'abord collecté des données de type audio auprès de personnes de différentes régions du Sénégal et créé un ensemble de données de 310 enregistrements audios. Nous avons ensuite formé un modèle de Deep learning. Ce mémoire s'est surtout intéressé à la construction de modèle pour la reconnaissance vocale des mots "oui" et "non" en Wolof et à son intégration dans une application d'autodiagnostic de la COVID-19.

Toute personne parlant le Wolof peut faire un autodiagnostic rapide de la COVID-19. L'autodiagnostic est constitué d'une série de questions auxquelles l'utilisateur devra répondre par "oui" et "non" en Wolof. Le système informe ensuite l'utilisateur et l'oriente sur les prochaines étapes.

En perspective, nous comptons élargir notre ensemble de données pour que notre modèle soit beaucoup plus puissant. Nous voulons également déployer notre modèle dans des applications mobiles et augmenter le vocabulaire reconnu en commençant par les symptômes de la COVID-19.

RÉFÉRENCES

- [1] James K. Tamgno, Pascal U. Elingui, Aristide T. Mendo'o, Morgan Richomme, Claude Lishou, Seraphin D. Oyono Obono. Speech Recognition and Text-to-speech Solution for Vernacular Languages. ICDT 2011 : The Sixth International Conference on Digital Telecommunications. ISBN: 978-1-61208-127-4.
- [2] Doctor Car.
<https://emergency-live.com/fr/Actualit%C3%A9s/S%C3%A9n%C3%A9gal%3A-une-voiture-de-combat-contre-Covid-19-L%27institut-polytechnique-de-Dakar-pr%C3%A9sente-au-robot-des-innovations-anti-Covid>. Consultée le 20 mars 2021.
- [3] La plateforme xel-xeeli. <https://xelxeeli.org>. Consultée le 20 mars 2021.
- [4] Le vélo Mobigel. <https://twitter.com/EnactusEsp/status/1256373511129571328>.
- [5] Levis, John & Suvorov, Ruslan. (2012). Automatic Speech Recognition. 10.1002/9781405198431.wbeal0066.
- [6] Ministère des Solidarités et de la Santé. Algorithme d'orientation Covid-19. France, mai 2020: <https://www.allocovid.com/home>
- [7] Mathieu Mangeot, Chantal Enguehard. DILAF : des dictionnaires africains en ligne et une méthodologie. Francophonie et Langues Nationales, Nov 2014, Dakar, Sénégal. fahal-01107550f.
- [8] Laurent Besacier, Elodie Gauthier, Mathieu Mangeot, Philippe Bretier, Paul Bagshaw, et al.. Speech Technologies for African Languages: Example of a Multilingual Calculator for Education. Interspeech 2015 (short demo paper), Sep 2015, Dresden, Germany. fahal-01170505f.
- [9] Tamgno, James & Barnard, Etienne & Lishou, Claude & Richomme, Morgan. (2012). Wolof Speech Recognition Model of Digits and Limited-Vocabulary Based on HMM and ToolKit. 10.1109/UKSim.2012.118.
- [10] NIANG CAMARA F.B., 2014. Dynamique des langues locales et de la langue française au Sénégal, 2014, Actes du XVIIe colloque international de l'AIDELF sur Démographie et politiques sociales, Ouagadougou, novembre 2012, 23 p. ISBN : 978-2-9521220-4-7
- [11] Mouhamadou Khoule, Mathieu Mangeot, El Hadji Mamadou Nguer, Mame-Thierno Cissé. iBaatukaay : un projet de base lexicale multilingue contributive sur le web à structure pivot pour les langues africaines notamment sénégalaises. Atelier Traitement Automatique des Langues Africaines TALAf 2016, conférence JEP-TALN-RECITAL 2016, Jul 2016, Paris, France. fahal-02054921.
- [12] Intégration des langues africaines avec de l'IA : le cas du Wolof: <https://baamtu.com/evenements/evenements-think-data-day/integration-des-langues-africaines-avec-de-lia-le-cas-du-oulof>
- [13] Baamtu et Weego: <https://web.facebook.com/baamtu/posts/961195550896181>
- [14] Elhadji Mamadou Nguer, Alla Lo, Cheikh Bamba Dione, Sileye Oumar Ba, Moussa Lo. UVS, UGB, UiB, Dailymotion, UVS. Senegal, Senegal, Norway, France, Sénégal.

- SENCORPUS: A French-Wolof Parallel Corpus. Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2803–2811 Marseille, 11–16 May 2020.
- [15] <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/deep-learning-vs-machine-learning/>. Consultée le 25 mars 2021.
- [16] <https://ichi.pro/fr/un-guide-complet-des-reseaux-de-neurones-convolutifs-la-methode-eli5-65776100198995>. Consultée le 25 mars 2021.
- [17] les réseaux de neurones convolutifs (CNN) : <https://www.imaios.com/fr/Societe/blog/Classification-des-images-medicales-comprendre-le-reseau-de-neurones-convolutifs-CNN>. Consultée le 25 mars 2021.
- [18] <https://qsstudy.com/physics/time-period-wave>. Consultée le 10 avril 2021.
- [19] https://www.phys.uconn.edu/~gibson/Notes/Section4_2/Sec4_2.htm. Consultée le 15 mars 2021.
- [20] https://commons.wikimedia.org/wiki/File:Signal_Sampling.svg
- [21] Spectre: <http://public.iutenligne.net/telecommunications/berthet/module-signaux-systeme/ExercicesSpectre/index.html>
- [22] <https://commons.wikimedia.org/wiki/File:FFT-Time-Frequency-View.png>