No d'ordre : 382/2021

Université Sultan Moulay Slimane
Faculté des Sciences et Techniques
Béni Mellal

Centre des Etudes Doctorales : Sciences et Techniques .
Formation doctorale : Mathématique et Physique appliquées (MPA).

THÈSE

Presentée par

Abdelkbir OUISAADANE
(Laboratoire: LIMATI)

Pour l'obtention du grade de

DOCTORAT NATIONAL

Discipline : Informatique
Specialité : Informatique

Contribution à l'amélioration du taux de reconnaissance automatique de la parole par des modèles hybrides

---

Soutenue le 11/12/2021 devant le jury composé de :

Mohamed FAKIR	: Professeur à la Faculté des Sciences et Techniques, Université Sultan Moulay Slimane, Béni Mellal,	Président du jury.
Jilali ANTARI	: Professeur à la Faculté polydisciplinaire, Université Ibn Zohr,Taroudante,	Rapporteur.
Khalid AUHMANI	: Professeur,à l'Ecole Nationale des Sciences Appliquées de l'Université Cadi Ayyad, Marrakech,	Rapporteur.
Ahmed BOUMEZZOUGH	: Professeur Habilité à la Faculté Polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal,	Rapporteur.
Belaid BOUIKHAENE	: Professeur à la Faculté Polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal,	Examinateur.
Abderezzak FARCHANE	: Professeur Habilité à la Faculté Polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal,	Examinateur.
Yassine SADQI	: Professeur Habilité à la Faculté Polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal,	Examinateur.
Said SAFI	: Professeur à la Faculté Polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal,	Directeur de la thèse.

Dédicace

Je dédie ce travail aux étoiles de ma vie A Ma trés chére mére ZAHRA, symbole de douceur, de tendresse, d'amour et d'affection. A mon trés cher pére MOHA, symbole de patience, de responsabilité, de force, de confiance et surtout d'espoir et d'amour. Je vous dédie entiérement cette thése à mon pére et à ma mére, mes piliers, mes exemples, mes premiers supporteurs et ma plus grande force. Merci pour votre soutien, votre amour, merci de n'avoir jamais douté de moi. Tout ce que j'espére, c'est que vous soyez fiers de moi aujourd'hui. Grâce à vous j'ai appris le sens du travail et de la responsabilité. Je voudrais te remercier pour ton amour, ta générosité, ta compréhension... Ton soutien fut une lumiére dans tout mon parcours. Aucune dédicace ne saurait exprimer l'amour l'estime et le respect que j'ai toujours eu pour toi. Ce modeste travail est le fruit de tous les sacrifices que tu as déployés pour mon éducation et ma formation. Je vous aime ***baba*** *et* ***yama*** *et j'implore Dieu tout-puissant pour qu'il vous accorde une bonne santé et une vie longue et heureuse.*
A Ma chére femme ***Siiham****, Je te remercie pour ton soutien dans les moments les plus difficiles. Je te remercie pour ta grande patience durant la préparation et rédaction de cette thése et de l'amour et la motivation que tu m'offres, Je t'aime.*

A mes deux petites princesses Alaa-Rahman et Firdaws, Merci pour le bonheur que vous me procure à chaque instant. Aucune ne dédicace ne saurait exprimer tout l'amour que j'ai pour vous. Votre gaieté me comble de bonheur. Puisse Dieu vous garder, éclairer votre route et vous aider à réaliser à votre tour vos vœux les plus chers, Je t'aime ***Alaa****, je t'aime* ***Firdaws****.*

A mes chers fréres et soeurs
A tous les membres de ma famille
A tous mes professeurs
A toutes mes amies
Abdelkbir Ouisaadane

Remerciements

Je tiens à remercier en premier lieu mon encadrant **SAID SAFI**, pour la patience dont il a fait preuve à mon égard, et pour l'orientation qu'il m'a donné à mon travail durant cette thèse. Je lui suis également reconnaissant d'avoir su me communiquer sa passion pour la recherche. Merci de m'avoir encadré, dirigé, encouragé tout au long de ma thèse, vos conseils et remarques m'ont été d'une très grande aide. Je le remercie également pour leurs efforts de relecture du manuscrit. Pour tout cela, je tiens à leurs témoigner toute ma gratitude.

J'aimerais remercier messieurs Jilali ANTARI, Khalid AUHMANI et Ahmed BOUMEZZOUGH pour avoir accepté d'être les rapporteurs de ma thèse. Je remercie également Mohamed FAKIR qui a accepté d'être le président du jury. Un grand merci également à messieurs Belaid BOUIKHAENE, Abderezzak FARCHANE et Yassine SADQI pour leur participation au jury comme examinateurs.

Mes remerciements sont adressés également mes collègues de labo **TAID** et **LIMATI** pour leurs aides et pour leurs participations à la construction du corpus DARIJA_MO.

Mes remerciements s'adressent à tous ceux qui m'ont apporté leur aide de prés ou de loin.

Les résultats rapportés dans cette thèse de doctorat résument les principaux efforts entrepris au cours des quatre dernières années. Merci à tous ceux qui nous ont contribué à l'une de ces étapes.

Nous espérons que la lecture a été enrichissante, que les fautes d'orthographe et de grammaire n'étaient pas trop nombreuses, que les références bibliographiques ne comportent trop de fautes et que les idées développées dans cette thèse sont intéressantes.

Nous sommes heureux de rejoindre le domaine de la recherche académique et scientifique. Cette thèse n'est que le début à la recherche scientifique professionnelle.

Table des Matiéres

Liste des figures

Liste des tableaux

Liste des abréviations et notations

ANN:	Artificial Neural Network
API:	Alphabet Phonétique International
ASR:	Automatic Speech Recognition
ARBDIGITS:	Arabic Spoken Digits Corpus
CNN:	Convolutional Neural Network
DCT:	Discrete Cosine Transform
DTW:	Dynamic Time Warping
DNN:	Deep Neural Network
FFT:	Transformée de Fourier Rapide (Fast Fourier Transform)
GMM:	Gaussian mixture model
HMM:	Hidden Markov Model
HTK:	Hidden Markov Model Toolkit
LPC:	Linear Predictive Coefficients
LPCC:	Linear Predictive Cepstral Coefficients
MFCC:	Mel Frequency Cepstral Coefficients
MLE:	Maximum Likelihood Estimation
NASCIW:	Noisy Arabic Speech Corpus for Isolated Words
PLP:	Perceptual Linear Predictive
SRAP:	Système de Reconnaissance Automatique de la Parole
SNR:	Signal-to-Noise Ratio
SDDN:	English Spoken Digits Database under Noise conditions
SVM:	Support Vector Machine
VQ:	Vector quantization
WER:	Word Error Rate
WRR:	Word Recognition Rate

Résumé

Résumé

L'amélioration des systèmes de reconnaissance automatique de la parole est l'un des défis actuels les plus importants, spécialement au milieu réel où des bruits ambiants élevés nous entourent de tous côtés, peuvent conduire donc à une dégradation des performances dans des conditions acoustiques défavorables. Pour résoudre ce problème, nous avons proposé et évalué dans le cadre de cette thèse la combinaison de deux grandes approches des systèmes RAP à savoir : l'approche d'entraînement multi-styles et les approches traditionnelles hybrides. Le but de l'approche d'entraînement bruités (multi-styles) est d'injecter de manière aléatoire des bruits à plusieurs niveaux SNR dans les données d'entraînement pour les systèmes de reconnaissance automatique de la parole (SRAP). Cette approche a été abordée spécialement pour la langue arabe standard et pour le dialecte marocain dans un milieu bruité. Nous avons construit dans un premier temps quatre corpus que nous avons utilisé dans les expériences. Deux ont été modifiés (SDDN et NASCIW) et deux nous les avions enregistré (ARADIGITS et DARIJA_Mo) dans des conditions différentes. Ensuite, ces bases de données ont été utilisé pour évaluer plusieurs systèmes (sept systèmes au total) avec plusieurs tests et par différents outils et techniques. Cette diversité des techniques de mise en œuvre de chaque module a été exploitée pour construire différents systèmes. Nous avons réalisé aussi en utilisant l'approche d'entraînement bruité une étude comparative entre quatre outils open source les plus connus (PocketSphinx, Kaldi, HTK et Matlab) et cinq modèles de classification (DNN-HMM, GMM-HMM, HMM, VQ-GMM et CNN).

De plus, nous avons présenté les étapes intermédiaires suivies pour la formation des modèles, y compris les modèles acoustiques et linguistique de chaque système. Nous avons analysé l'effet du bruit de fond à différents niveaux sur les corpus NASCIW et DARIJA_Mo. L'extraction des paramètres acoustique, y compris en cas de bruit, se fait par les coefficients cepstraux MFCC. Les résultats obtenus montrent que l'outils Kaldi et PocketSphinx sont plus robuste au bruits avec meilleurs performances par rapport aux outils HTK et Matlab selon l'entraînement avec le corpus NASCIW. Ensuite, les expériences présentées dans cette thèse confirment que l'approche d'entraînement bruité fonctionne bien pour le modèle hybride DNN-HMM. Des améliorations significatives du WRR ont été observées sur les systèmes RAP basés sur DNN-HMM par rapport aux autres classificateurs avec une augmentation d'environ 5 %.

Enfin, Les résultats obtenus suggèrent qu'une augmentation substantielle du taux de reconnaissance de mots (WRR) est obtenue lorsque l'entraînement se fait avec les données bruitées par rapport à celle fondé sur des données propres. Le meilleur taux de reconnaissance obtenu par le système basé sur le modèle DNN-HMM dans les conditions de test propres est de 97.10%.

Mots clefs : Système de Reconnaissance Automatique de la Parole (SRAP),Entraînement Multi-styles ,MFCC, DNN-HMM, GMM-HMM, HMM, VQ-GMM, CNN, PocketSphinx, Kaldi, HTK, Matlab.

Abstract

Abstract

Improving automatic speech recognition systems is one of the most important current challenges, especially in real environment where high ambient noise surrounds all sides, which can therefore to lead performance degradation under adverse acoustic conditions. we have proposed and evaluated in the framework of this thesis the combination of two main approaches of RAP systems namely: the multi-style training approach and the traditional hybrid approaches. The purpose of the noisy (multi-style) training approach is to randomly inject multi-level SNR noises into training data for automatic speech recognition systems (ASRs). This approach was implemented especially for standard Arabic speech and for the Moroccan dialect in noisy environment. We first built four corpora that we used in the experiments, two were modified (SDDN and NASCIW) and two we recorded (ARADIGITS and DARIJA_Mo) under different conditions. Then these databases were used to evaluate several systems (seven systems in total) with several tests and by different tools and techniques. This diversity of techniques for implementing each module has been exploited to build different systems. We also carried out using the noisy training approach a comparative study between four of the most famous open-source tools (PocketSphinx, Kaldi, HTK and Matlab) and five classification models (DNN-HMM, GMM-HMM, HMM, VQ GMM and CNN).
In addition, we have presented the intermediate steps followed for the formation of the models, including the acoustic and linguistic models of each system. We analyzed the effect of background noise at different SNR levels on the NASCIW and DARIJA_MO corpora. The extraction of the acoustic parameters, including in case of noise, is done by the MFCC cepstral coefficients. The results obtained show that the Kaldi and PocketSphinx tools are more robust to noise with better performance compared to HTK and Matlab tools depending on training with the NASCIW corpus. Then, the experiments presented in this thesis confirm that the noisy training approach works well for the DNN-HMM hybrid model. Significant improvements in WRR were observed on ASR systems based on DNN-HMM compared to other classifiers with an increase of approximately 5%.
Finally, the results obtained suggest that a substantial increase in the word recognition rate (WRR) is obtained when learning is done with noisy data compared to that based on clean data. The best recognition rate obtained by the system based on the DNN-HMM model under the specific test conditions is 97.10%.

Keywords : Automatic Speech Recognition (ASR),multi-style training, MFCC, DNN-HMM, GMM-HMM, HMM, VQ-GMM, CNN, PocketSphinx, Kaldi, HTK, Matlab.

Liste des publications

1) **Articles dans des journaux internationaux**

1. OUISAADANE, S.SAFI « A comparative study for Arabic speech recognition system in noisy environments ». International Journal of Speech Technology (2021), Published: 27 April 2021, https://doi.org/10.1007/s10772-021-09847-7; Publisher: Springer Nature;
ISSN:1381-2416. Int J Speech Technol : https://www.scopus.com/sourceid/26802;

2. OUISAADANE, S.SAFI, M.FRIKEL « Arabic digits speech recognition and speaker identification in noisy environment using a hybrid model of VQ and GMM». TELKOMNIKA, 2193-2204, DOI: http://dx.doi.org/10.12928/telkomnika.v18i4.14215,
August 2020. TELKOMNIKA :https://www.scopus.com/sourceid/21100256101;

1) **Proceedings de confrences internationales**

1. A. OUISAADANE, S. SAFI, M. FRIKEL « English Spoken Digits Database under noise conditions for research: SDDN ». 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), Publisher: IEEE, DOI: 10.1109/WITS.2019.8723698 , 30 May 2019;

2. A. Ouisaadane, S. Safi, M. Frikel, « Reconnaissance Automatique du dialecte marocain en milieu réel à l'aide de PocketSphinx ». Publication dans le colloque ICSAT'2020 (book_icsat_2020_fr), Caen, France. https://hal.archives-ouvertes.fr/hal-02974050/ .

Introduction Générale

Problématique

L'interaction homme-machine et devenue une forme de communication courante dans le monde actuel. Historiquement, cette interaction était effectuée par des périphériques tels qu'un clavier, une souris ou un écran tactile, mais la tendance actuelle est de les remplacer par des systèmes de reconnaissance automatique de la parole (Automatic Speech Recognition: ASR) dans des situations qui nécessitent une forme de communication plus naturelle [2]. De nos jours, il devient assez courant que les gens utilisent des assistants personnels intégrés dans leurs smartphones, communiquent automatiquement des systèmes de dialogue dans les centres d'appels ou contrôlent les appareils intelligents de leur maison et de leur bureau avec la voix. En outre, les systèmes de transcription automatique sont utilisés pour créer des sous-titres pour les émissions de télévision, pour indexer les archives audio ou pour transcrire des enregistrements personnels [3]. Ces progrès sont venus avec l'introduction d'algorithmes avancés de traitement du signal et d'apprentissage automatique, ainsi que grâce à l'augmentation massive des données disponibles et de la puissance de calcul grâce au développement des matériels et des outils technologiques. Par conséquent, les performances des systèmes de reconnaissance automatique de la parole ont atteint des niveaux très élevés, les taux d'erreurs de mots étant diminués. De nombreuses techniques ont été développées pour augmenter les performances des systèmes ASR [4]. Malheureusement, malgré ce succès de l'ASR au fil des décennies, les performances de celui-ci se dégradent dans les environnements incontrôlés [3]. La variabilité du discours augmente considérablement la difficulté de ces tâches. Les conditions d'enregistrement introduisent une variabilité supplémentaire qui dégrade la qualité des signaux et on dit souvent que la parole est déformée [3]. Le bruit soit de type stationnaire et non stationnaire est l'un des défis les plus difficiles dans la reconnaissance automatique de la parole [3]. Ce phénomène est particulièrement intéressant dans les applications réelles où le niveau de bruit change fréquemment. Pour contrer cette faible performance dans un environnement bruyant, la recherche sur la reconnaissance vocale au cours des dernières décennies s'est concentrée sur la création des systèmes robustes pour reconnaître la parole dans les environnements bruyants [5].

Motivation et objectif de cette thèse

Plusieurs recherches sont faites pour améliorer les technologies de traitement des signaux vocaux dans les divers domaines d'application comme la traduction, secteur militaire, secteur médical, sécurité, contrôle à distance, maison intelligente, recherche vocale, système de navigation automobile etc. Les systèmes de reconnaissance vocale existants fonctionnent bien pour les langues européennes comme l'anglais [2]. Malgré les progrès réalisés avec les systèmes de reconnaissance vocale, beaucoup ne comprennent toujours pas les dialectes arabes. Les recherches sont encore limitées, en particulier dans les conditions bruyantes [5]. La langue arabe fait partie des langues les plus parlées au monde avec environ 300 millions de locuteurs natifs. La présence de bruit de fond, ainsi que la diversité des dialectes arabes, sont considérées comme des défis pour la reconnaissance vocale en arabe. Etudier le dialecte arabe marocain, ce qui est très difficile, ce qui est même difficile dans les règles orthographiques, les multiples accents et le vocabulaire selon les régions du Maroc ainsi que les systèmes deviennent plus compliqués lors de l'ajout du facteur de bruit.
L'objectif principale de notre thése est de réaliser un systéme de reconnaissance automatique de la parole permettant une bonne résistance au bruit, en particulier pour la langue Arabe Standard et Dialecte Marocain dans des milieux bruités selon la combinaison de deux approches. La premiére est de faire l'entraînement avec des données bruitées (l'entraînement multi-styles ou multi-conditions) dans la phase d'apprentissage. La deuxiéme concerne à utiliser l'hybridation des modéles dans la phase de décodage. Par suite, de comparer les résultats de notre travail avec des travaux réalisés dans ce domaine. Une part importante du travail aussi, portera sur la constitution des corpus spécifiques pour étudier les performances des techniques proposées. Ensuite, nous avons développé des applications en temps-réel avec plusieurs outils implémentant des systémes de RAP basant sur les deux approches. Enfin, des évaluations en conditions simulées ou différents sont effectuées.

Structure du document

Cette thèse est organisée en quatre chapitres, après l'introduction générale, comme suit:
Dans le premier chapitre, nous présentons un bref aperçu de l'état de l'art sur la reconnaissance automatique de la parole, y compris la structure productive de la parole, les éléments principaux d'un système de reconnaissance et nous découvrirons les modèles phonétiques et les modèles de langages couramment utilisés. Nous donnons également une revue de la littérature. Le deuxième chapitre nous permet de présenter les techniques les plus utilisées dans la reconnaissance vocale : la quantification vectorielle, les modèles de Markov cachés et les réseaux de neurones artificiels. Nous décrivons aussi les librairies les plus utilisé dans le domaine de la RAP et que nous avons utilisé dans notre thèse telles que la plateforme open source PocketSphinx, Kaldi et le toolbox de Matlab (Deep Learning Toolbox et Voicebox). Ainsi les divers corpus des bruits accessibles au public (CHiME3, Aurora-4 et NOISEX-92) et certains d'entre eux sont les nôtres (ARABDIGITS,SDDN,NASCIW et DARIJA_Mo). Par la suite dans le chapitre 3, nous décrivons premiérement les déférents types de bruit stationnaires et non stationnaires. Deuxièmement, nous expliquons le problème que pose

le bruit additif dans les systèmes de RAP. Nous consacrons aussi à une revue des algorithmes et des techniques utilisées dans le domaine d'améliorations des systèmes de RAP. Par la suite, nous présentons les techniques adoptées dans cette thèse, pour débruiter les corpus utilisés surtout pour la reconnaissance de la parole arabe. Le dernier chapitre présente les résultats des expérimentations que nous avons effectués. Dans ce chapitre nous décrivons notre cadre expérimental incluant les corpus de parole bruités qui ont été mis en place pour tester les différentes techniques de reconnaissance vocale. Nous y présentons les différentes évaluations et les expériences comparatives que nous avons effectué en environnement calme et bruité ainsi que les discussions des résultats obtenus et les conclusions que nous en avons tiré. Enfin, en conclusion de ce travail, nous résumons les méthodes proposées et les résultats obtenus, et nous présentons des perspectives des travaux effectuées dans cette thése.

Chapter 1

État de l'art et contexte technique de la reconnaissance de la parole

1.1 Introduction

Au cours de ce premier chapitre, nous allons décrire brièvement les mécanismes entrant en jeu lors de la production et perception de la parole chez l'être humain. Ensuite nous nous intéresserons à l'architecture fonctionnelle d'un système de la reconnaissance automatique de la parole, nous verrons quels sont les fondements théoriques des différents algorithmes utilisés, les différentes caractéristiques d'un système de reconnaissance de la parole, la structure générale de ce dernier, les méthodes d'analyse du signal pour une paramétrisation .Nous terminerons enfin par un bref aperçu historique des systèmes de reconnaissance automatique de la parole dans les conditions bruitées en particulier pour la langue Arabe.

1.2 Production et perception de la parole

La parole est le moyen de communication le plus naturel entre les humains. Elle fait partie de notre vie quotidienne. Elle a une influence capitale, d'où l'importance de développer des techniques permettant une communication efficace dans plusieurs applications [1]. La communication humaine peut être considérée comme un processus de cinq éléments reposant sur un échange entre deux ou plusieurs personnes utilisant un code fermé de gestes, de mots ou d'expressions qui rend compréhensible une information formelle ou informelle transmise d'un émetteur à un récepteur. Le signal vocal qui se produit et se perçoit par les êtres humains est un sujet essentiel qui il doit être considéré avant de pouvoir poursuivre et décider quelle approche utiliser pour la reconnaissance vocale. La communication humaine est considérée comme un diagramme complet du processus de la production de la parole à la perception de la parole entre le locuteur et l'auditeur comme le montre la Fig. 1.1 [1]. Le processus de production de parole est un mécanisme très complexe qui repose sur une interaction entre le système neurologique et physiologique. Il y a une grande quantité d'organes et de muscles qui entrent dans la production de sons des langues naturelles. Le fonctionnement de l'appareil phonatoire

humain repose sur l'interaction entre les poumons, le larynx, et les cavités supra-glottiques.

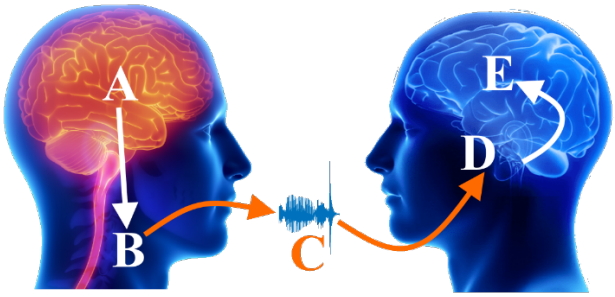

Figure 1.1: Représentation schématique de la production et de la perception de la parole [1]

Le schéma (Figure 1.1) représente le circuit de parole au moment où deux personnes communiquent entre elles. Il contient Cinque éléments :

- A : Production de la parole
- B : Appareil phonatoire du locuteur
- C : Propagation acoustique de l'onde sonore dans l'air
- D : Perception de la parole chez l'auditeur
- E : Compréhension de la parole

Le premier élément (A. Production de la parole) est associé à la production de la parole signalant l'esprit du locuteur. Cette production est utilisée par le mécanisme vocal humain (B. Appareil phonatoire du locuteur) pour produire des variations de pressions dans l'air. Ces variations de pression sont perçues comme du son par l'oreille humaine. La forme d'onde est transmise via l'air (C Propagation acoustique de l'onde) à l'auditeur. Pendant ce transfert, l'onde acoustique peut être affectée par des sources externes, par exemple du bruit, ce qui se traduit par une forme d'onde plus complexe. Lorsque l'onde atteint le système auditif de l'auditeur (les oreilles), l'auditeur perçoit la forme d'onde (D. Perception de la parole chez l'auditeur) et l'esprit de l'auditeur (E. Compréhension de la parole) commence à traiter cette forme d'onde pour comprendre son contenu afin que l'auditeur comprenne ce que le locuteur essaie de lui dire [1]. L'un des problèmes avec un système de RAP est de simuler la manière dont l'auditeur traite la parole produite par le locuteur. Plusieurs actions se déroulent dans le cerveau de l'auditeur et dans le système auditif pendant le traitement des signaux vocaux. Le processus de perception peut être considéré comme l'inverse du processus de production de la parole [1]. Pour le dialogue homme-homme, la parole est l'un des principaux moyens plus simple et populaire de communication. Néanmoins, cette simplicité (pour l'être humain) renferme un traitement très complexe fait par notre cerveau, de la production de la parole jusqu'à sa perception et sa compréhension, ce qui rend la parole difficilement automatisable pour une machine. L'avancement technologique et surtout de l'informatique a suscité le besoin de simuler de manière semi-parallèle le principe de dialogue homme-homme dans le dialogue homme-machine, des moyens qui libéreraient l'homme d'un contact constant avec la machine limitant ainsi l'utilisation du clavier et autres périphériques qui rendaient la communication avec la machine très difficile et très lente. La reconnaissance vocale présente de grands avantages pour l'interaction homme-machine. Il est facile d'obtenir des données vocales sans nécessite de compétences particulières telles que l'utilisation du clavier, la saisie de données en cliquant sur les boutons des programmes GUI, etc. Le transfert

de données textuelles dans des supports électroniques à l'aide de la parole est environ 8 à 10 fois plus rapide que l'écriture manuelle. De plus, l'utilisateur peut continuer à saisir du texte tout en se déplaçant ou en effectuant tout travail qui l'oblige à utiliser ses mains. Puisqu'un microphone ou un téléphone peut être utilisé. Il est possible aussi et économique de saisir des données à distance par téléphone.

1.3 La reconnaissance automatique de la parole (RAP)

1.3.1 Système de RAP

La reconnaissance automatique de la parole (RAP) (ASR en anglais) est un sujet important pour le traitement de la parole. Elle a considérablement progressé et a révélé d'innovants algorithmes et techniques pour le traitement statistique de la parole [2]. La RAP est une technologie qui permet à une plate-forme électronique comme un smartphone ou un ordinateur de reconnaitre les mots prononcés par des humains. C.-à-d, la tâche d'un système de RAP est de convertir l'énoncé parlé prévu en une forme textuelle. Les caractéristiques acoustiques, qui sont extraites du signal vocal, sont comparées avec les vecteurs formés de modèles linguistiques et acoustiques. Donc, le système de RAP rend l'interaction homme- machine plus flexible et très simple [2]. Parmi les différents domaines couverts par les systèmes RAP, c'est la reconnaissance d'un discours continu de vocabulaire plus de 5 000 mots différents, où les mots sont prononcés naturellement et pas de silence entre eux, plus le bruit ambiant [3][4].
Dans les dernières années, les chercheurs sont intéressés à surmonter certains défis tels que le bruit, canal de transmission, locuteur, l'écho, etc. Les performances de RAP sont considérablement dégradées lorsque le bruit ambiant est différent de celui des données d'apprentissage. Un système de RAP est utilisé de façon optimale lorsque ses conditions de test et d'apprentissage sont semblables. étant donné un système entraîné à partir d'un corpus de parole propre, l'objectif du filtrage de bruit est de prétraiter le signal bruité de test, afin de pouvoir l'utiliser comme entrée du système [2]. De nombreuses approches ont été introduites pour résoudre ce problème avec divers degrés de complexité et de taux d'amélioration. Le modèle général de résolution de ce problème se divise en plusieurs approches à savoir l'amélioration du signal audio, les approches basées sur le front-end qui améliorent le signal dans le domaine des fonctionnalités, les méthodes back-end [5], former un modèle acoustique général et transformer les modèles pour qu'ils correspondent aux vecteurs acoustiques bruités. Les défis récents tels que le défi REVERB [6] et CHiME3 [7] ont démontré l'efficacité de ces approches. Nous allons approfondir dans cette partie en examinant les solutions d'apprentissage avec les vecteurs acoustiques bruités et celles fondées sur des modèles statistiques avant d'examiner les solutions hybrides.
Les systèmes modernes de reconnaissance automatique de la parole ont été introduits par Jelinek[8]. Le but d'un système RAP est de fournir la transcription textuelle d'un signal audio X fourni en entrée en une séquence de mots la plus probable. Il peut être décomposé typiquement en cinq modules, comme illustré dans la figure 1.2 :

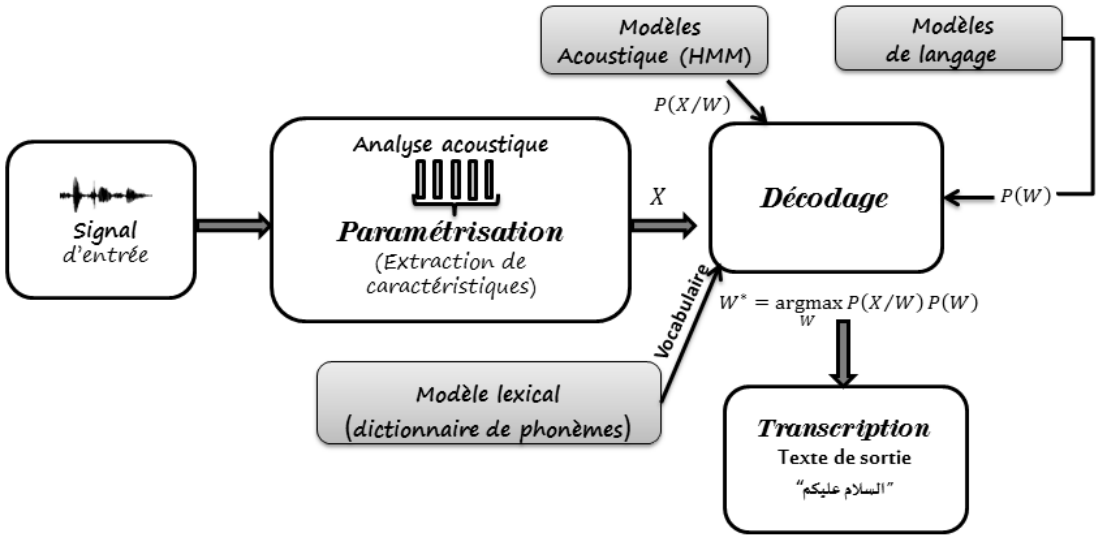

Figure 1.2: L'architecture générale d'un système de reconnaissance automatique de la parole

a) Un module de paramétrisation et de traitement du signal : permet d'extraire l'information utile à la caractérisation de son contenu linguistique en réduisant la redondance du signal de la parole. Le signal sonore brut est converti en une séquence des vecteurs acoustiques adaptée à la reconnaissance.

b) Des modèles acoustiques : modélisant un ensemble réduit d'unités de sons élémentaires d'une langue donnée. Ces unités acoustiques sont plus petits que les mots par rapport au nombre d'échantillons. Ce sont des modèles phonétiques statistiques (HMM) estimés à l'aide d'une grande quantité de données de parole.

c) Un modèle lexical : fourni la transcription de mots de la langue modélisée par un simple dictionnaire phonétique. Les plus développés sont construits à partir des automates probabilistes, capables de représenter chaque mot d'un dictionnaire par une probabilité.

d) Un module de langage : introduit la notion de contraintes linguistiques par un modèle statistique utilisant une grande base de données textuelles pour estimer les probabilités d'une suite de phonèmes, de manière automatique. Il permet de guider le décodeur vers les suites de mots les plus probables.

e) Un module de décodage : consiste à sélectionner, parmi l'ensemble des phrases possibles, celle qui correspond le mieux à la phrase prononcée. Le décodage de la parole s'effectue à l'aide de tous les modules déjà présentés.

Comme décrit dans la figure 1.2, le système de reconnaissance automatique de la parole statistique s'appuie sur l'extraction des caractéristiques acoustiques $X = x_1, x_2, ..., x_T$ et sur deux modèles probabilistes (un modèle de langage et un modèle acoustique) afin d'appliquer la fonction ***argmax*** et trouver la séquence de mots hypothèse $\hat{W} = w_1, w_2, ..., w_N$ les plus probables. Le modèle de langage renvoie la probabilité de Wet le modèle acoustique estime la probabilité de $P(X|W)$. La qualité du système dépend essentiellement de la qualité de ces deux modèles probabilistes.

La séquence de mots hypothèse $\hat{W}$est obtenue comme suit :

$$\hat{W} = \arg\max_{W} P(W|X) \tag{1.1}$$

En appliquant la formule de Bayes, l'équation (1.1) devient :

$$\hat{W} = \arg\max_{W} \frac{P(X|W)\ P(W)}{P(X)} \tag{1.2}$$

Avec :
-$P(X|W)$: La probabilité d'observer la séquence X des vecteurs acoustiques connaissant la suite de phonèmes W. Cette probabilité est estimée par les modèles acoustiques.
-$P(W)$: La probabilité a priori d'observer la suite de phonèmes W, indépendamment du signal. Elle est déterminée par le modèle de langage.
-$P(X)$: La probabilité d'observer la séquence de vecteurs acoustique X. Elle est identique pour chaque suite de phonèmes ($P(X)$ ne dépend pas de W). Elle n'est pas utile et peut donc être ignorée.
Alors Comme, les paramètres acoustiques Xsont fixes, l'équation peut être simplifier comme suite :

$$\hat{W} = \arg\max_{W} P(X|W)\ P(W) \tag{1.3}$$

Cette méthode statistique permet de représenter, de manière flexible, les niveaux acoustiques et linguistiques dans le même processus de reconnaissance.

1.3.2 Extraction des caractéristiques

1.3.2.1 Aperçu générale

L'extractions des fonctionnalités et des caractéristiques ou bien paramétrisation du signal vocal est la partie la plus importante du processus de reconnaissance. étant donné que le signal acoustique contient des informations autres que les mots prononcés par un locuteur (bruit, silence, écho, etc.), la paramétrisation du signal de la parole consiste à caractériser le signal de parole en entrée afin de trouver les informations pertinentes pour les exploiter dans le système de reconnaissance et produire la séquence de mots W appelés des vecteurs acoustiques. Le but de cette opération est d'obtenir une nouvelle représentation qui est plus compacte et plus appropriée à la modélisation statistique. Avant l'extraction des paramètres, un prétraitement qui consiste à détecter les zones de silence est effectué afin de n'utiliser que les zones d'activité acoustique. Cette opération est très difficile à mener à cause de la présence de bruit qui change les caractéristiques du signal de parole. Les techniques les plus utilisées pour résoudre ce problème se basent sur le taux de passage par zéros et l'amplitude moyenne des trames courtes. Cette technique est une version modifiée de l'algorithme de Rabiner et Sambur [9]. L'objectif principal est de produire des représentations vectorielles qui caractérisent le signal de parole via des fenêtres glissantes de 10 à 30 ms. Différentes techniques sont utilisées dans la littérature pour extraire les paramètres acoustiques à partir d'un signal de parole brut qui peuvent être enrichis avec leurs dérivées premières et secondes : Mel-Frequency Cepstral Coefficients (MFCC) [10], Perceptual Linear Prediction (PLP) [11] et Linear Prediction Ceptral Coefficients (LPCC) [12], Relative Spectral Filtering (RASTA)[13], etc. La plupart des techniques de paramétrisation consis-

tent à décrire l'enveloppe du spectre à court terme dans le domaine fréquentiel. D'autres techniques peuvent être utilisées comme l'analyse en ondelette. Malgré les efforts qui ont été investis dans la conception de paramétres acoustiques plus robustes aux distorsions acoustiques, les paramétres MFCC et PLP restent jusqu'à ce jour les deux méthodes de paramétrisation prépondérantes en RAP. Les MFCCs sont plus discriminants, plus robustes au bruit ambiant, donc sont plus préférés. Pour ces raisons, nous optons pour la paramétrisation MFCC dans le cadre de cette thése.Les différentes étapes de l'analyse MFCC sont détaillées dans dans la section suivante.

1.3.2.2 Coefficients MFCC

La paramétrisation MFCC (Mel Frequency Cepstral Coefficients) est basée sur la perception humaine de son : l'échelle de Mel, basant sur l'évidence connue que les renseignements portés par les composantes de la fréquence basse du signal de parole sont plus importants phonétiquement pour les humains que les composantes à haute fréquence [14]. Les coefficients cepstraux sur l'échelle Mel (MFCC, Mel-Frequency Cepstral coefficients) est la plus courante méthode d'extraction de caractéristiques utilisé dans les systèmes de reconnaissance de la parole et du locuteur. Les coefficients MFCC sont plus discriminant, plus robustes au bruit ambiant et moins corrélés entre eux. L'analyse acoustique est divisée en trois étapes, le filtrage analogique, la conversion numérique et le calcul de coefficients, nous donnerons brièvement les étapes d'une analyse MFCC dans la figure 1.3.

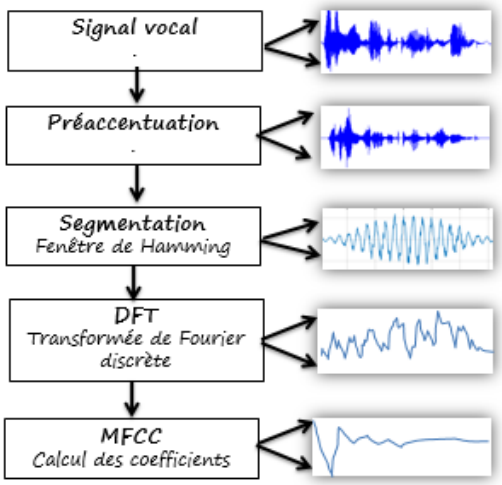

Figure 1.3: Etapes d'extraction des paramètres MFCC

Le signal de la parole est variant au cours du temps. Pour cette raison, il doit être divisé en trames de faible durée (typiquement 20 à 30 ms) où le signal sonore peut être considéré comme quasi-stationnaire, avec un pas de décalage entre deux trames successives de l'ordre de 10 ms. Un vecteur cepstral est extrait pour chaque trame. Le signal de la parole $S(n)$ est pré-accentué à l'instant n pour relever les hautes fréquences par l'équation 1.4, pour une valeur classique α de 0.97 (α peut prendre une valeur comprise entre 0.9 et 1).

$$S(n) = S(n) - \alpha S(n-1) \tag{1.4}$$

Pour compenser cette décroissance, on effectue une préaccentuation en utilisant un filtre passe haut. La fenêtre la plus utilisée dans la reconnaissance de la parole est la fenêtre de Hamming [14] à réponse impulsionnelle finie et pour rendre proche de zéro les extrémités de la trame temporelle. Chaque trame temporelle est ensuite fenêtrée avec la fenêtre de Hamming pour éliminer les discontinuités au niveau des bords [15], les coefficients $w(n)$ d'une fenêtre de Hamming de longueur n sont calculés selon la formule :

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) & ;0 \leq n \leq N-1 \\ 0 & autrement \end{cases} \tag{1.5}$$

N est le nombre total d'échantillons et n l'échantillon courant. Après le fenêtrage, la transformée de Fourier rapide (Fast Fourier Transform: FFT) est calculée sur chaque trame pour extraire des composantes fréquentielles du signal dans le domaine temporel. Les coefficients MFCC sont basés sur l'estimation de l'enveloppe spectrale dans un échelle perceptive [16], [17] . Les échelles perceptives les plus utilisées sont l'échelle Mel ou l'échelle Bark [18], [19],[20]. Dans notre cas nous avons fait le choix d'utiliser l'échelle Mel. Celle-ci peut être définie par la relation suivante entre la fréquence de la parole en Hertz et sa correspondance en l'échelle de Mel, elle s'établit comme suit [21]:

$$Mel(f) = 2595\ \log_{10}\left(1 + \frac{f}{700}\right) \tag{1.6}$$

La dernière étape de calcul des coefficients MFCCs est le calcul de la transformation en cosinus discrète (Discrete cosine Transform: DCT) sur les sorties du banc de filtre. Les n premiers coefficients cepstraux C_k (en gé n éral n est choisi entre 10 et 15) sont calculés directement à partir du logarithme des énergies m_i sortant d'un banc de F filtres en échelle de fréquences non linéaire Mel, se calcule par la relation (3.4) [21], [22], [23].

$$C_k = \sum_{i=1}^{p} \log(m_i)\ \cos\left(k\left(i - \frac{1}{2}\right)\frac{\pi}{p}\right) \quad ;1 \leq k \leq n \tag{1.7}$$

Où p est le nombre de coefficients spectraux calculés précédemment et n est le nombre de coefficients cepstraux que nous voulons calculer $(n \times p)$. Finalement, ceci conduit à l'obtention des coefficients MFCCs. Ainsi, pour chaque trame de parole, un ensemble de coefficients MFCCs est calculé. Cet ensemble est appelé vecteur acoustique et représente les caractéristiques phonétiquement importantes et très utiles pour une analyse plus approfondie dans le traitement de la parole.

1.4 Modèle acoustique

Le modèle acoustique est le composant le plus important dans un système de RAP. Son objectif est d'estimer la probabilité $P(X|W)$, comme le montre la figure (1.2), quiétablit une relation statistique entre les caractéristiques acoustiques et les unités vocales, c.-à-d. il conduit à la réalisation de modèles dont le rôle est d'estimer la probabilité qu'une séquence d'unités linguistiques (phonèmes, diphones, syllabes, mots, etc.…) particulière a généré le signal émis. Le décodeur utilise les informations disponibles de toutes les autres composantes pour déduire la transcription la plus probable du signal d'entrée [24]. Dans la section précédente (1.3), nous avons décrit comment extraire les

vecteurs acoustiques X du signal de parole qui caractérisent la suite de phonèmes W.
Dans la littérature deux approches fondamentales sont couramment utilisées pour la modélisation acoustique : les approches statistiques et les modèles probabilistes. Des années 70 à nos jours, ces approches ont connu une nette amélioration, des performances remarquables, une robustesse au bruit et à la variabilité des locuteurs [25]. Plusieurs approches de modélisation acoustique ont été proposées dans la littérature, mais les principales sont actuellement les modèles de Markov cachés (HMM – Hidden Markov Models), les réseaux de neurones profonds (DNN - Deep Neural Network) et les modèles hybrides HMM-DNN et HMM-GMM. Ce sont les modèles que nous avons adopté dans nos expériences, nous les détaillerons au chapitre 2, donc, différents modèles acoustiques seront évalués surtout pour le facteur de la robustesse au bruit. Les modèles de Markov cachés sont aujourd'hui les plus utilisés pour la modélisation statistique acoustique, même pour les autres approches comme les réseaux de neurones qu'ils s'utilisent pour estimer la vraisemblance de chaque phonème et l'algorithme Viterbi trouve le chemin maximisant ces vraisemblances et de modéliser chaque unité de parole par un MMC [26]. Le formalisme des MMCs, a permis de développer, de nos jours, des algorithmes performants de modélisation acoustique. Ces algorithmes ont prouvé leur efficacité dans de nombreux domaines de la RAP et sont utilisés dans les outils de construction de systèmes de RAP les plus répandus tels que CMUSphinx[1] , KALDI[2], Julius[3] , HTK[4] , MATLAB[5] etc. Nous détaillons ces techniques dans le chapitre suivant.

1.5 Modèle de langage

Le modèle de langage (Language Model: LM) accompagne le modèle acoustique pour créer une cohérence linguistique entre les différents éléments acoustiques prononcés. Il définit l'existence de chaque mot du modèle de langue dans un dictionnaire de prononciation. Il va permettre de sélectionner, parmi toutes les séquences de mots possibles, celle qui a la plus grande probabilité d'apparition. Les modèles de langage attribuent des probabilités aux séquences de mots, c'est-à-dire qu'ils contiennent des informations sur les mots susceptibles de se reproduire. Donc, le but du modéle de langage est d'estimer la probabilité a priori de toutes les séquences de mots qu'il est possible de construire à partir du lexique. En utilisant ce modèle, le dispositif de reconnaissance vocale impose également les informations linguistiques au sein du processus de la recherche et du décodage. Il en résulte une sortie qui est trouvée non seulement sur la base des informations acoustiques, mais également de la grammaire de la langue cible. De nombreux modèles de langage déterministes et probabilistes sont proposés dans la littérature [3]. Les modèles de langage déterministes tels que les grammaires hors-contextes sont principalement utilisés pour les petits vocabulaires, pour la reconnaissance vocale continue de grand vocabulaire (LVCSR), les modèles de langage probabiliste sont adoptés, les plus souvent et les plus simples de LM sont sous la forme d'un modèle N-gramme [27],[3]. Les N-grammes donnent la probabilité qu'un mot apparaisse en fonction de $n-1$ des mots précédents, les probabilités

[1] *https://cmusphinx.github.io*
[2] *http ://kaldi − asr.org/*
[3] *https://julius.osdn.jp/en_index.php*
[4] *http://htk.eng.cam.ac.uk*
[5] *https://www.mathworks.com/products/audio.html*

de N-gramme peuvent être calculées simplement en observant les fréquences de différentes séquences de mots. Divers outils ont été développés pour la modélisation du langage. L'une des boîtes à outils les plus utilisées est la boîte à outils SRILM[6]. Pour plus de détails, sont présentés dans les références [3],[27]. La probabilité a priori de la séquence de M mots W : $P(W)$ exprimée en section (1.3) est estimée par le modèle du langage. Cette probabilité peut être décomposée en un produit de probabilités conditionnelles comme le montre la formule (1.8):

$$P(W) = \prod_{i=1}^{M} P(w_i | w_1, w_2, ..., w_{i-1}) \tag{1.8}$$

Cette formulation suppose qu'un mot w_i peut être prédit uniquement à partir de l'historique des mots qui le précèdent.

1.6 Reconnaissance automatique de la parole arabe au milieu bruité

Les systèmes de RAP, qui sont entraînés dans des conditions d'utilisation bien contrôlées fonctionnent bien dans des conditions similaires, mais les performances de ces systèmes se dégradent considérablement lorsque les conditions d'apprentissage et d'utilisation d'un système sont différentes, ainsi des énoncés cibles corrompus par le bruit de fond (notamment en ce qui concerne le type et le niveau de bruit). Ce décalage entre les conditions d'apprentissage et de test a un impact direct sur le taux de la reconnaissance en raison des variations fortes de la structure spectrotemporelle de la parole. En effet, d'après Olivier Siohan [28], si près de 97.8% de taux de reconnaissance peut être obtenu par un système de reconnaissance multilocuteurs (1011 mots) entraîné et testé dans un milieu calme en parole très peu bruitée (RSB = 36 dB), ce même système n'obtient pas plus de 3% de reconnaissance si on ajoute un bruit blanc gaussien (avec un rapport signal/bruit (RSB) de 0dB) au signal de test (milieu très bruité). Par conséquent, les modèles acoustiques entraînés à partir de données vocales obtenues dans des conditions propres ne peuvent pas modéliser efficacement les caractéristiques acoustiques bruyantes [28],[29]. De plus, un système de RAP dont les modèles acoustiques ont été entraînés dans le même milieu que le milieu de test, même bruité, donnera toujours de meilleures performances qu'un système de RAP entraîné dans un milieu calme. Par conséquent, certaines de nos expériences porteront sur l'entraînement des modèles acoustiques dans les différentes conditions réelles, donc c'est parmi les approches importantes que nous adoptons dans notre thèse.
Les systèmes actuels sont donc dans l'ensemble très peu robustes aux variations même si celles-ci peuvent paraître assez faibles à l'oreille. Les sources de variabilité de la parole peuvent être classées en trois catégories, selon leur provenance [29]:

1. **L'environnement du locuteur** : bruit corrélé à la parole (réverbération, réflexion) ou additif (bruit ambiant, etc.) ;

[6] *http : //www.speech.sri.com/projects/srilm/*

2. **Le locuteur lui-même** : selon son état et son mode d'expression : essoufflement, stress, effet Lombard (qui amène un locuteur à modifier sa voix lorsqu'il est placé dans une ambiance très bruitée), rythme d'élocution, fatigue, etc. ;

3. **Les conditions d'enregistrement** : type de microphone, distance au microphone, canal de transmission (distorsion, écho, bruit électronique, etc.).

L'effet de type de bruit produit un décalage entre les données d'apprentissage et celle de test utilisées avec les systèmes de reconnaissance. Par conséquent, si le taux d'erreur de reconnaissance dans des environnements bruyants doit être réduit, il faut trouver une fonction qui réduira les différences entre ces deux environnements. Cela peut être fait de deux manières, en modifiant les paramètres du modèle de la parole pour qu'ils correspondent à l'environnement vocal ou en transformant les données de reconnaissance vocale de l'environnement bruité vers l'environnement dans lequel les modèles ont été entraînés.
De nombreuses techniques ont été proposées pour augmenter la robustesse des systèmes, notamment en ce qui concerne leur résistance aux bruits. Ces techniques peuvent être divisées sur les approches de base suivantes[4], [28] , [29], [30]:
-**Les techniques paramétriques qui sont basées sur l'extraction des caractéristiques robustes**. Elles offrent une résistance intrinsèque au bruit, par exemple les méthodes RASTA [31], MFCC, etc. Ces méthodes ont pour but principal le débruitage du signal de parole et après de l'analyser. Ces techniques consistent à ne pas différencier la parole bruitée de la parole non bruitée et à considérer le système de RAP comme étant indépendant des conditions de bruit. Ces techniques conduisent à l'utilisation de mesures de distances et à l'extraction d'indices acoustiques dont la résistance au bruit est connue et sûre. Elles permettent ainsi de ne pas modifier l'étape suivante du décodage acoustique-phonétique. On peut citer, au rang ces méthodes, la normalisation par moyenne cepstrale, qui est très populaire [32]. Le cepstre n'obtient cependant pas de bons résultats en milieu bruité, certaines recherches ont été menées pour trouver un opérateur différent du logarithme pour effectuer la déconvolution. Il est alors possible de parler d'analyse cepstrale généralisée, cette analyse peut être faite de différentes manières [33], [34]. Il existe également des méthodes fondées sur l'analyse linéaire discriminante selon un échelle Mel [35], [28], les méthodes RASTA-PLP [31]. Cette dernière technique permet d'obtenir de bons résultats dans le cadre de la reconnaissance de la parole en milieu bruité puisqu'elle permet d'effectuer une séparation sonore de manière aveugle. Dans certains travaux de notre thèse nous avons choisi cet approche avec MFCC plus robuste [36].
- **Les techniques permettant d'estimer la parole propre, c'est à dire débarrassée du bruit environnemental (Speech Enhancement Techniques : Amélioration de la parole)** : Ces techniques permettent d'estimer la parole propre utilisant le principe général de l'estimation d'un spectre du bruit qui est ensuite soustrait au spectre du signal bruit, principe permettant d'améliorer le rapport signal-sur-bruit (SNR). Elles consistent à transformer le signal de la parole bruité en un signal le moins bruité possible qui soit le plus proche possible en qualité d'un signal de la parole non bruité. Ces techniques essaient donc d'effectuer une amélioration qualitative du signal d'entrée. Le bruit est donc réduit avant que le signal de parole ne soit traité par le système de reconnaissance. Cette réduction peut se faire dans le domaine spectral ou dans le domaine cepstral, par soustraction ou filtrage du signal original. On peut citer parmi ces plusieurs méthodes : Les méthodes qui se

réclamant de la soustraction dans le domaine spectral, la soustraction spectrale (SS) est l'une des techniques de suppression de bruit les plus couramment utilisées, est fréquemment appliquée dans de nombreux schémas de traitement frontal robuste au bruit, elle élimine le bruit de fond additif, car elle est basée sur la soustraction d'une composante additive dans le domaine spectral. Le principe de cette technique est simple et donne généralement des résultats satisfaisants. D'autre part, le niveau de suppression du bruit ainsi que la distorsion possible de la parole nettoyée dépendent fortement d'une estimation correcte du spectre d'amplitude du bruit de fond [37], [38], [30]. Certaines méthodes qui s'appliquent dans le domaine cepstral [30],[39].
Il existe plusieurs algorithmes particuliers de soustraction spectrale diffèrent principalement dans la procédure d'estimation, par exemple la soustraction spectrale multi-bande, le filtrage de Wiener, le filtrage adaptatif modifié de Wiener, détection d'activité vocale (VAD), la soustraction spectrale itérative et la soustraction spectrale basée sur les propriétés perceptives [40],[41]. Certaines méthodes permettent d'obtenir de meilleurs résultats en faisant une combinaison entre eux. Des détails supplémentaires décrivant ces techniques peuvent être trouvés dans [30],[40],[42],[43].Dans notre travail, nous avons utilisé l'algorithme de filtrage de Wiener pour tester son efficacité en utilisant des donnés bruités.
- **Les techniques fondées sur des traitements pouvant s'accommoder d'un signal bruité (appelés aussi techniques basées sur des modèles) :** Ces techniques essaient de transformer les modèles de références de la parole de l'environnement d'origine, où a été fait l'apprentissage, en des modèles tenant compte du bruit de l'environnement effectif. Elles consistent à entraîner les modèles dans le bruit. Ainsi, les différences entre conditions d'entraînement et de test seraient totalement éliminées. Cette technique effectue donc une adaptation, ou compensation, des modèles au bruit. Elle effectue une décomposition des connaissances lors de l'apprentissage, modélisant d'une part la parole propre et d'autre part le bruit, et recombine cette connaissance lors de la phase de reconnaissance [45]. Cette recombinaison a été mise en oeuvre avec les modèles de Markov cachés et les réseaux de neurones et d'autres méthodes de mondialisation acoustiques (voir le chapitre 2) car ils permettent de modéliser un bruit avec une architecture très simple. Contrairement à ce qui est fait par les techniques d'amélioration du signal, le bruit n'est pas affaibli et sera présent lors de l'étape de reconnaissance puisqu'il est considéré comme une partie du signal à traiter. L'adaptation peut se faire par utilisation de modèles en parallèle, par utilisation de prototypes de bruit ou adaptation directe des paramètres du système de décodage de la parole, par régression linéaire ou ajustement stochastique [45]. Nous avons utilisé cette technique aussi dans l'un de nos travaux pour l'apprentissage des mots bruités dans le corpus modifié SDDN [44].
Ces trois approches constituent aujourd'hui l'état de l'art de la RAP en milieu bruité. Elles présentent cependant des lacunes puisque tous les bruits ne peuvent pas encore être traités. Par conséquent, la recherche est continue de manière intensive afin d'améliorer la qualité des systèmes et d'augmenter leurs robustesses. On peut citer parmi les dernières méthodes récentes dans ce domaine (plus de détails dans le livre [30]):
-Techniques basées sur l'extraction des caractéristiques robustes.
-Techniques basées sur des modèles (Adaptation Techniques, Retraining, …)
-Techniques conjointes (Noise Adaptive Training, Joint front-end and back-end training, Uncertainty Decoding – Missing Feature Theory,) [46].

-Apprentissage multi-référentiel ou multi-condition - Extraction de la parole à partir des signaux de mélange (Extraction of Speech from Mixture Signals) - Réseaux à grand nombre de microphones (Microphone Arrays) [47]. - Noise robust front ends (Kalman Filtres, Correlation Features, ...). - Noise robust Back ends (Parallel Model Combination, Model Adaptation, …). - Deep Noise Suppression Challenge

- REVERB challenge[7], CHiME-1st, 2nd, 3rd, 4th, 5th and 6th Challenge[8], Interspeech2020[9], ICASSP2020[10]. Ce sont des conférences dont le thème principal était la robustesse des systèmes de RAP dans des conditions bruitées.

Notons qu'il est possible de combiner certaines de ces techniques pour obtenir de meilleurs résultats en tirant parti des qualités de chacune des techniques choisies. Nous reviendrons sur certaines de ces techniques dans les chapitres 2 et 3.

Pour étudier la robustesse des systèmes de reconnaissance vocale pour la langue arabe, quelques travaux sont faits dans ce domaine, on mentionne par exemple les recherches de Touazi et Debyeche [48] qui ont présenté ARADIGIT-2 une base de données de reconnaissance de chiffres arabe basée sur la boîte à outils (HTK) et l'indépendante des locuteurs arabes, ils ont l'utilisé pour l'évaluation des systèmes robustes au bruit. Les tests se faits dans différentes conditions et type de bruit. Les résultats obtenus ont donnés un WER de 0,44% dont l'apprentissage avec des donnés propre et de 0,58% dont l'apprentissage avec des donnes multi-conditions. Certains systèmes vocaux ont été proposés pour la dialecte marocain de tamazight. Un exemple notable est le système de reconnaissance des 33 lettres de l'alphabet Amazigh réalisés par neuf locuteurs, conçu par Meryam Telmem et Youssef Ghanou [49]. Ils ont proposé une architecture basée sur des modèles de Markov cachés (HMM) avec la librairie open source CMUSphinx, et ont atteint une précision d'environ 82%.

Amrous et al [50],[51] ont proposé deux contributions pour tenter de résoudre les deux problèmes majeurs responsables en grande partie de la performance ainsi que de la robustesse des systèmes de RAP. Le premier c'est de trouver l'ensemble des caractéristiques qui représentent le mieux le signal de parole et le deuxième est lié à la différence entre les environnements d'apprentissage et ceux de test. Donc la solution pour le premier problème d'après Amrous et al est d'exploiter la richesse du signal de parole par la combinaison de différentes sources d'information acoustique et de l'information prosodique. Ces deux informations sont fusionnées pour aboutir au vecteur de caractéristiques du signal de parole à base des modèles de Markov Cachés, ainsi, pour résoudre le deuxième problème, ils se déclinent dans l'utilisation des Réseaux Bayésiens Dynamiques pour traiter le problème de la robustesse des systèmes de RAP. Les deux solutions ont permis d'améliorer d'une manière significative les taux de reconnaissance dans les milieux bruités.

On peut aussi mentionner quelques travaux qui portent sur la reconnaissance de la parole Arabe/Dialecte/Amazigh avec différentes approches dans les milieux bruités par exemple : [52], [53], [54], [55], [56].

Pour notre cas de cette thèse, on travaille avec les deux approches principales pour augmenter la robustesse des systèmes de RAP en particuliers pour la langue arabe traditionnelle et le dialecte marocain dans les conditions réelles. La première consiste à améliorer les paramètres et les caractéristiques

[7] *https* : *//reverb*2014.*dereverberation.com*
[8] *https* : *//chimechallenge.github.io/chime*6/
[9] *http* : *//www.interspeech*2020.*org*
[10] *https* : //2020.*ieeeicassp.org*

acoustiques par des méthodes d'amélioration et de réduction du bruit. La seconde approche consiste à estimer un modèle du bruit et à le combiner avec un modèle de parole non bruitée. Pour faire l'adaptation des modèles acoustiques et avoir la combinaison parallèle de modèles et d'apprentissage des systèmes dans des conditions différentes et réelles.

1.7 Conclusion

Dans ce chapitre, nous avons décrit brièvement les principes de fonctionnement des systèmes de RAP, en définissant la structure globale d'un système de RAP et les différents modules qui constituent ce dernier. Commençant par le signal acoustique de la parole, qui présente une grande variabilité qui complique la tâche des systèmes RAP. Cette complexité provient de la combinaison de plusieurs facteurs, comme la redondance du signal acoustique, la grande variabilité intra et interlocuteurs, les effets de la coarticulation en parole continue, ainsi que les conditions d'enregistrement. Différentes approches sont envisagées pour la reconnaissance de la parole telles que les méthodes analytiques, globales et les méthodes statistiques. Actuellement la majorité des systèmes de RAP sont construits selon la méthode statistique en utilisant les modèles de Markov cachés. Nous avons décrit brièvement le principe de fonctionnement des systèmes de RAP basés sur les modèles HMM ainsi que leur mise en œuvre pratique qui sera décrit dans le chapitre suivant. Ensuite, Nous avons mis l'accent sur trois modules principaux, à savoir : le module d'extraction des paramètres et le module acoustique et le module de langage. Nous avons terminé ce chapitre par présenter la problématique du bruit dans les systèmes de reconnaissance et les différents axes de solution qui existent dans la littérature, tout en positionnant nos solutions proposées. Dans le chapitre suivant, nous continuerons l'état de l'art des méthodes de modélisations acoustiques, les outils et l'environnement de travail que nous avons utilisé dans cette thèse.

Chapter 2

Approches et outils utilisés dans la reconnaissance de la parole

2.1 Introduction

Dans ce chapitre, nous décrivons l'ensemble des techniques qui sont fréquemment utilisées dans les étapes de classification et d'apprentissage des systèmes de RAP lesquels nous avons utilisé dans nos travaux. Parmi ces techniques on cite : les modèles de Markov cachés (HMM), les réseaux de neurones artificiels (ANN) ou les modèles combinés (modèles hybrides) (GMM-HMM, GMM-VQ, HMM-DNN), la quantification vectorielle (VQ), l'alignement temporel (DTW), les supports à marge vaste (SVM), etc. [57],[58]. Ces techniques peuvent s'appliquer afin de résoudre un probléme de reconnaissance des formes. Elles se distinguent principalement, par la manière dont les formes de références sont créées, modélisées et par la méthode qui sert à classer les formes inconnues. La plupart de ces approches sont basées sur la modélisation stochastique, qui fournit un cadre pour décrire statistiquement des modèles et formaliser le processus de prise de décision de telle sorte que la perte moyenne par décision soit la plus faible possible. Par la suite nous survolons les principes des différents modèles acoustiques et linguistiques ainsi que les algorithmes liés à leur apprentissage et de décodage.

Le problème de reconnaissance vocale peut se formuler selon un problème de classification. Différentes approches ont été développées, néanmoins on peut les classer en trois grandes familles:

- **L'approche vectorielle ou déterministes** : la voix est représentée par un ensemble de vecteurs de paramètres dans l'espace acoustique. Ses principales techniques sont la reconnaissance à base de la technique de programmation dynamique (Dynamic time warping: DTW) et par quantification vectorielle (VQ).
- **L'approche statistique** : consiste à représenter chaque mot par une densité de probabilités dans l'espace des paramètres acoustiques. Elle couvre les techniques de modélisation par les modèles de Markov cachés, par les mélanges de gaussiennes et par des mesures statistiques du second ordre.
- **L'approche connexionniste** : consiste principalement à modéliser la parole par des réseaux de neurones.

Nous proposons ici un aperçu des méthodes déterministes (quantification vectorielle), statistiques (modèles à mélange de distributions gaussiennes (GMM), modèles de Markov cachés) et connexionnistes (réseaux de neurones convolutionnels (CNN). Nous proposons certaines méthodes pour adapter le système afin qu'il prenne en compte le bruit (compensation) ou qu'il utilise plusieurs modèles combinés (modèles hybrides). Ensuite, nous présentons les outils et les librairies open source de reconnaissance vocale que nous avions utilisé pour la simulation de nos expériences, parmi eux : Kaldi, CMUSphinx, HTK et Matlab.

2.2 Modèle de Markov caché

2.2.1 Principe du HMM

Le rôle principale d'un système de RAP est d'inverser les informations de base implicitement intégrées dans une séquence d'observations acoustiques. L'efficacité des systèmes de RAP dépend fortement du choix de l'application, du modèle acoustique (indépendante ou dépendante du locuteur) et du modèle de langage. Cependant, deux problèmes majeurs rendent la tâche difficile pour un système de RAP. Premièrement, la mise en correspondance des symboles avec la parole n'est pas une procédure individuelle, car différents symboles sous-jacents peuvent représenter des sons de parole similaires. En raison de certaines contraintes telles que l'environnement, la variabilité du locuteur, l'émotion, l'âge du locuteur, le sexe, l'électronique, l'accent régional, phonologie, qualité du microphone, etc. Deuxièmement, les limites des symboles ne peuvent pas être déterminées avec précision à partir de la forme d'onde de la parole. Par conséquent, nous ne pouvons pas considérer le signal vocal comme une séquence de motifs statiques concaténés. Nous devons utiliser un bon outil statistique pour la modélisation acoustique, car il est nécessaire de pouvoir modéliser les variabilités et les dépendances de chaque unité en fonction de son contexte. Donc, l'entrainement de distributions statistiques fondé sur les modèles de Markov cachés représente la meilleure approche pour modéliser la variabilité observée sur des exemples réels [59].
Le modèle de Markov caché (HMM) est depuis longtemps la méthode préférable et fréquemment utilisée pour modéliser des séquences dans les systèmes de RAP. Cette section tente de donner un bref aperçu de la manière dont ils sont appliqués pour la reconnaissance de la parole. Une brève explication des chaînes de Markov est d'abord nécessaire, sans entrer trop profondément dans la théorie, car, ce n'est pas notre but dans cette thèse. Le signal de parole est supposé être produit par un automate stochastique fini, construit à partir d'un ensemble d'états stationnaires régis par des lois statistiques [59]. En d'autres mots, le formalisme des modèles HMM suppose que le signal de parole est formé d'une séquence de segments stationnaires. Tous les vecteurs associés à un même segment stationnaire étant supposés avoir générés par le même état HMM. Chaque état de cet automate est caractérisé par une distribution de probabilité décrivant la probabilité d'observation des différents vecteurs acoustiques. Les chaînes de Markov modélisent des séquences d'états observables. Une chaîne de Markov de premier ordre, qui est la plus couramment utilisée, est une chaîne où la probabilité d'un état donné ne dépend que de l'état précédent. C'est l'hypothèse de Markov qui permet d'avoir une

forme simple de modélisation temporelle. Les chaînes de Markov du premier ordre sont décrites par un ensemble de probabilités de transition d'état, représentées par la matrice$A = \{a_{ij}\}$où l'élémenta_{ij}est la probabilité de passage de l'état i à l'état j. Chaque état d'un modèle HMM est capable de générer un nombre fini de sorties possibles. En produisant un mot, le système fait passer le vecteur d'observation d'un état à un autre. Chaque état contribue à la sortie jusqu'à ce que le mot entier soit généré. HMM est un processus stochastique doublement intégré, où les états sous-jacents ne sont pas observables (cachés), mais ne peuvent être vus qu'à travers la séquence d'observation produite (vecteurs acoustiques). Chaque état contient une densité de probabilité d'émission $b_j(x_t)$ qui permet de mesurer la probabilité pour un élément $x_t$de la séquence d'observations d'être associé à (émis par) cet état. Dans le cas d'un processus markovien (d'ordre 1), la probabilité a_{ij} de passer de l'état i à l'état j à l'instant t en émettant l'observation x_t ne dépend pas des états parcourus aux instants précédents. La figure 2.1 représente un simple HMM à 3 états de gauche à droite, composé de trois états émetteurs et de deux états non émetteurs (entrée et sortie). Chaque état peut émettre une observation x_t selon une probabilité d'observation à l'instant t régie par $b_j(x_t)$ [59], [60] ,[61].

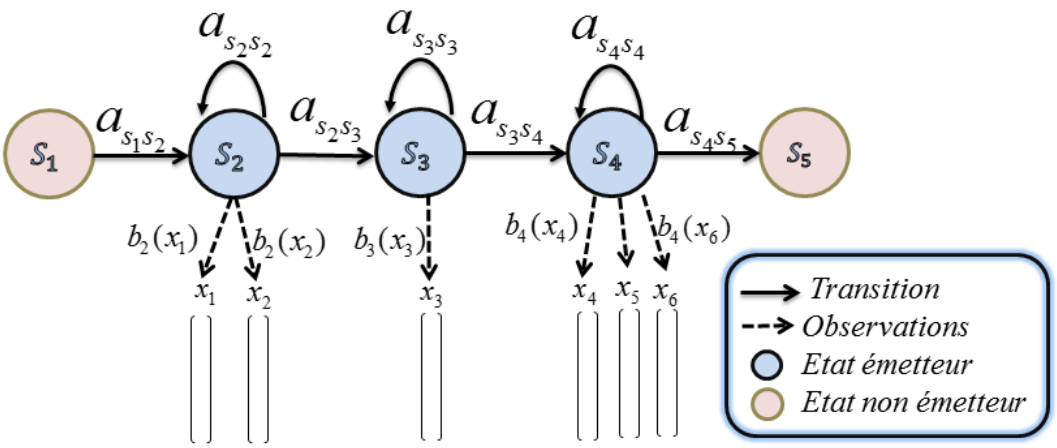

Figure 2.1: Un simple Modèle de Markov caché de gauche à droite à 5 états.

Un modèle de Markov caché est complètement défini par les paramètres λ : $\lambda = (N, A, B, \pi)$où

$\sim$ La séquence d'états $S = \{S_1, S_2, ..., S_N\}$ (dans notre cas on travail parfois avec $N = 5$ et parfois avec $N=3$)

$\sim$ $\boldsymbol{N}$ est le nombre d'états du modèle: $\{i; i = 1, ..., N\}$ $\sim A = \{a_{ij}\}$ est la matrice des transitions entre les états, a_{ij}représente la probabilité de transition pour aller de l'état i à l'état j.

$$a_{ij} = P(S_t = j/S_{t-i} = i) \forall i, j \in \{1, N\} \tag{2.1}$$

Cette matrice n'est pas pleine et on peut soit se déplacer à droite soit rester dans l'état courant. Les coefficients de cette matrice A doivent vérifier la propriété suivante: $\forall i, \ \sum_{j=1}^{N} a_{ij} = 1$ Pour la plupart des système de RAP utilisant des HMMs, $a_{ij} = P(s_i|s_j)$. Ces HMMs sont dits d'ordre 1. Dans ce cas, les probabilités de transition entre deux états ne dépendent ni du temps ni de l'historique des états.

On utilise le plus souvent des HMMs dits de Bakis [59].C'est un modèle qui peut être gauche-droite ou inverse (droite-gauche). Ce modèle est appelé ainsi car il n'autorise pas de transition d'un état vers un autre d'indice inférieur comme l'exprime dans la formule (2.2) [59][60].

$$a_{ij} \begin{cases} = 0 & \forall j < i \\ > 0 & \forall j > i \end{cases} \tag{2.2}$$

De plus, le vecteur des probabilités initiales a la propriété suivante :
$\pi_i = 0 \ si \ i \neq 1 \ et \ \pi_i = 1 \ si \ i = 1$. Toute séquence d'états d'un modèle de Bakis commence donc à l'état 1 et se termine à l'état N.C'est-à-dire qu'aucun saut ne peut être effectué vers un état précédent.Les longueurs des transitions sont généralement limitées à une longueur maximale, généralement deux ou trois :$a_{ij} = 0, si \ j > i + \delta \ avec \ \delta = 1, 2, ...$
Le modèle de Bakis se caractérise par le fait que $\delta = 2$. Ce modèle permet de modéliser des signaux qui évoluent avec le temps, par exemple, en reconnaissance de la parole, il est souhaitable d'utiliser ce modèle qui modélise les observations de manière successive car c'est la propriété de la parole.
$\sim$ La séquence d'observations $X = \{x_1, x_2, ..., x_N\}$associée à la séquence d'états S.
$\sim$ $\pi_i$la probabilité initiale, c'est à dire la probabilité d'être dans l'état i à l'instant initial.
Tel que la distribution initiale des états π_i est exprimé par : $\pi_i = P(s_1 = i) \ ; \ i = 1, ..., N$
$\sim$ $B = b_i(x_t)$ est l'ensemble des probabilités d'émission, c'est-à-dire la probabilité d'observer le vecteur x_t sachant que le processus Markovien est dans l'état i:

$$b_i\,(x_t) = P(x_t|S_t = i), \begin{cases} \forall\, i \in \{1,\, N\} \\ \forall\, t \in \{1,\, T\} \end{cases} \tag{2.3}$$

Dans le cas des HMMs continus, La probabilité d'émission $b_i\,(x_t)$ des observations continues x_t est généralement calculée par une somme pondérée de G_i qui est gaussienne, $N\,(\mu,\Sigma)$ appelé aussi modèle de mélange de gaussiennes (Gaussian Mixture Model- GMM), chaque gaussienne est caractérisée par un vecteur moyen $\mu_{i,k}$et une matrice de covariance $\Sigma_{i,k}$ pondérées par $w_{i,k}$. La probabilité d'émission $b_i\,(x_t)$est alors définie par la formule suivante (2.4) [60]:

$$\boldsymbol{b_i\,(x_t)} = \sum_{\boldsymbol{k=1}}^{\boldsymbol{M}} \boldsymbol{w_{i,k}}\ N\left(\mathbf{x_t}, \mu_{\mathrm{i,k}}, \Sigma_{\mathrm{i,k}}\right) \quad \boldsymbol{;\ i = 1, 2, ..., N} \tag{2.4}$$

Où M représente le nombre de gaussiennes de l'état i, $w_{i,k}$ représente le poids de pondération de la k^{ème} gaussienne dans l'état i, donné par:

$$\sum_{\boldsymbol{k=1}}^{\boldsymbol{M}} \boldsymbol{w_{i,k}} = \boldsymbol{1} \ \ \boldsymbol{;\ i = 1, 2, ..., N} \tag{2.5}$$

Il est clair d'après la discussion ci-dessus qu'une spécification complète d'un HMM nécessite deux paramètres du modèle (N et M), des séquences d'observation et les trois mesures de probabilité tels que n, A et B. Pour plus de simplicité, la notation $[\lambda = (A, B, \pi)]$ est utilisée pour décrire l'ensemble complet des paramètres du modèle.

2.2.2 Application du Modèle HMM à la reconnaissance de la parole

Après avoir défini les paramètres inconnus A, B, M, N et π, la génération d'un HMM est faite, en utilisant la séquence d'observation $X = \{x_1, x_2, ..., x_T\}$, où chacune des x_i est l'une des M observations et T représente la longueur de la séquence d'observation.
Afin d'obtenir un modèle utile, il y a trois problèmes fondamentaux et ceux-ci doivent être résolus pour les applications de reconnaissance vocale. Nous ne traiterons pas, dans ce document, toute la théorie car cette étude qui dépasse le cadre de notre recherche. Plus de détails dans la référence originale [61]. Les trois problèmes avec leurs solutions sont expliqués ci-dessous:

- **Problème d'Evaluation:** Etant donnée une séquence d'observations $X = \{x_1, x_2, ..., x_T\}$ et le modèle $\lambda = (A, B, \pi)$, comment calculer $P(X|\lambda)$?
- **Solution :**

Le premier problème peut être considéré comme un problème d'évaluation. Chaque modèle entraîné représente un événement acoustique, alors le but est de trouver quel modèle est le plus susceptible de produire une sortie correcte pour une séquence d'observation donnée. Donc, la solution de ce problème est d'appliquer les deux algorithmes d'estimation "forward-backward".
Le premier est l'algorithme directe "Forward", c'est la variable directe $\alpha_t(i)$, définie comme la probabilité d'observer la séquence $(x_1, x_2, ..., x_t)$et d'être à l'état i à l'instant t connaissant le modèle λ. Où :

$$\alpha_t(i) = P(x_1 x_2 ... x_t, S_t = i/\lambda) \tag{2.6}$$

Le deuxième est l'algorithme rétrograde " Backward ", c'est la variable $\beta_t(i)$ correspond à la probabilité d'observer la séquence $(x_{t+1}, ..., x_T)$et d'être à l'état i à l'instant t connaissant le modèle λ. Où :

$$\beta_t(i) = P(x_{t+1} ... x_T, S_t = i/\lambda) \tag{2.7}$$

Après le calcul de la probabilité avant $\alpha_t(i)$et la probabilité arrière $\beta_t(i)$ par les deux algorithmes, cela se traduit par le calcul de probabilité:

$$P(X|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\, \beta_t(i)\, et \beta_T(i) = 1 \tag{2.8}$$

- **Problème de Décodage:** Etant donnée une séquence d'observations $X = \{x_1, x_2, ..., x_T\}$et les paramètres de modèle$\lambda = (A, B, \pi)$, comment choisir une séquence d'états optimale $S = \{s_1, s_2, ..., s_T\}$ afin de donner la meilleure explication des observations X ?
- **Solution :**

Dans le deuxième problème, nous essayons d'exposer la partie cachée du modèle. Il doit être clair que pour tous sauf le cas des modèles dégénérés, il n'y a pas de séquence d'états « correcte » à trouver. Par conséquent, des critères optimaux représentent la meilleure option pour traiter ce problème. L'algorithme de Viterbi qui est un algorithme de programmation dynamique similaire à l'algorithme "forward" est utilisé pour cet objectif, il permet de trouver la séquence d'états optimale qui maximise $P(X, S|\lambda)$.

- **Problème d'Apprentissage:** Comment déterminer les paramètres du modèle $\lambda = (A, B, \pi)$ pour maximiser $P(X|\lambda)$?
- **Solution :**

Le troisième problème est le problème d'apprentissage. Pour une séquence d'entraînement de donnée, un modèle est créé pour chaque événement acoustique. Le problème d'apprentissage est le plus important pour la plupart des applications HMM, car une adaptation optimale des paramètres du modèle en fonction des données d'apprentissage observées, c'est-à-dire, pour créer les meilleurs modèles de phénomènes réels. Ici, l'objectif principal est d'ajuster les paramètres du modèle (A, B,) pour maximiser la probabilité de la séquence d'observation étant donné le modèle. Il n'y a pas de solution directe (analytique) connue pour satisfaire le critère d'optimisation. Cependant, Baum et ses collègues ont utilisé une technique itérative pour traiter ce problème connue sous le nom de méthode Baum-Welch [61]. Il s'agit d'une procédure itérative, cas particulier de l'algorithme d' Estimation-Maximisation (EM) (description disponible dans [61] et [62]).

2.3 Quantification vectorielle

La quantification vectorielle (VQ) est une technique non-paramétrique qui permet de décrire un ensemble de données par un faible nombre de vecteurs formant un dictionnaire associé aux données. Le dictionnaire est en général calculé de telle façon que la distance moyenne entre un vecteur issu des données et son plus proche voisin dans le dictionnaire soit la plus petite possible. La quantification vectorielle est une méthode de groupage qui est d'autant plus adaptée que les données présentent naturellement des "points d'accumulation" autour desquels la densité de vecteurs issus des données est importante [63]. Pour la reconnaissance du locuteur, la mesure de similarité entre deux ensembles de vecteurs acoustiques consiste à évaluer la distance moyenne d'un des deux ensembles de vecteurs acoustiques en utilisant le dictionnaire optimisé pour l'autre ensemble de vecteurs acoustiques par quantification vectorielle.
Nous utiliserons donc l'approche VQ, dans notre thèse, en raison de sa mise en œuvre la plus simple avec une grande précision. Cette technique consiste à extraire des vecteurs de caractéristiques de petite taille représentatifs comme moyen efficace et de caractériser les caractéristiques spécifiques du locuteur. En utilisant ces données d'entraînement, les fonctionnalités sont regroupées pour former un livre de codes pour chaque locuteur. Il s'agit d'un processus de mappage de vecteurs d'un grand espace vectoriel vers un nombre fini de régions dans cet espace [64].
La quantification scalaire consiste à représenter une valeur d'un échantillon de signal pas forcement audio avec une précision réduite. Par exemple la représenter avec une valeur appartenant à un ensemble plus petit que l'ensemble original. C'est le cas typique de la conversion analogique/digitale.
Lorsque ce principe est appliqué par bloc d'échantillons (*vecteurs*), on peut parler de quantification vectorielle. La quantification vectorielle est alors une généralisation de la quantification scalaire. Mais, pendant que la quantification scalaire est dans sa forme la plus simple juste une conversion analogique/digitale, la quantification vectorielle est une méthode de codage/compression puissante. Elle est souvent utilisée dans les télécommunications pour le codage de la source, ou dans la com-

pression des données notamment dans la compression des images. Elle est aussi un puissant outil de classification. En 1980, Linde, Buzo et Gray (LBG) ont proposé un algorithme de conception VQ basé sur une séquence d'apprentissage. L'utilisation d'une séquence d'apprentissage contourne le besoin d'intégration multidimensionnelle. La VQ conçu à l'aide de cet algorithme est mentionné dans la littérature en tant que LBG-VQ [63]. Elle repose sur un partitionnement de l'espace acoustique en sous-espaces. Chaque sous-espace est associe à leur vecteur particulier appelé "centroïde" (vecteur moyen) (voir figure 2.2). Ce vecteur est lié à la distance minimale intra-classe. Donc, la quantification vectorielle est définie par un doublet : un ensemble de vecteurs représentatifs appelés mots $C = c_1c_2....c_M$ qui forme un dictionnaire (*codebook* en anglais) et un critère de distorsion $d(.,.)$[64] ,[65].

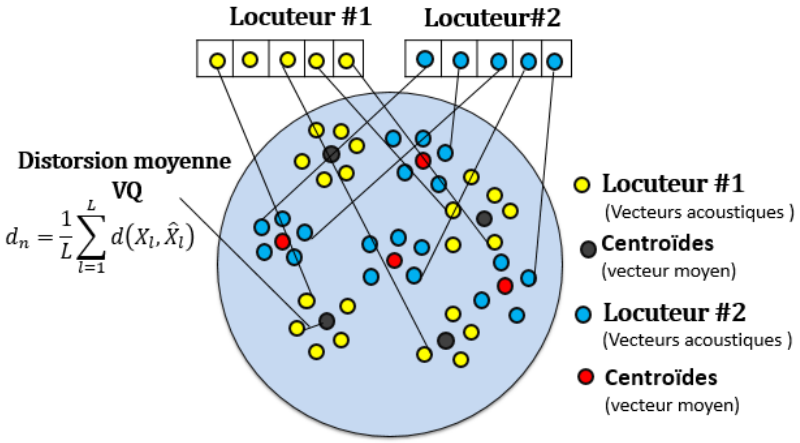

Figure 2.2: Exemple de quantification vectorielle

Lors de la phase de reconnaissance, une distance est calculée entre un vecteur de test et chaque vecteur centroide du dictionnaire. La distance minimale est assignée au vecteur de test. La distance d'une séquence de vecteurs de test est obtenue par moyenne des distances minimales attribuées à chacun des vecteurs de test.

La figure 2.2 montre un schéma conceptuel pour illustrer ce processus de reconnaissance. Un locuteur peut être distingué d'un autre en fonction de l'emplacement des centroïdes. En phase d'apprentissage, en utilisant l'algorithme de clustering LBG, cet algorithme d'optimisation itératif fonctionne à partir d'un dictionnaire initial [63]. Chaque spectre est alors reconnu par rapport au répertoire de prototypes. De cette façon, au lieu de représenter le mot par une séquence de vecteurs, il est représenté par une séquence de nombres (ou centroïdes) correspondant aux prototypes. Une mesure de distorsion peut être obtenues en calculant la distance moyenne entre les spectres et les spectres prototypes les plus voisins. Dans figure 2.2, nous nous limitons à présenter deux locuteurs et leurs vecteurs acoustiques. Les cercles jaunes font référence aux vecteurs acoustiques du locuteur1 tandis que les cercles bleus sont du locuteur2. Le dictionnaire VQ spécifique au locuteur est généré pour chaque locuteur connu en regroupant ses vecteurs acoustiques d'apprentissage. Les mots (centroïdes) sont représentés dans la figure 2.2 par des cercles noirs pour le locuteur 1 et des cercles rouges pour le locuteur 2. La distance entre un vecteur et le « centroïde » le plus proche d'un « *codebook* » est appelée distorsion VQ.

En phase de test, une voix inconnue, après extraction des vecteurs caractéristiques vocales, sera comparée au « *codebook* » de chaque locuteur dans la base de données des locuteurs et la distorsion sera calculée. La distorsion VQ illustre la distance par rapport au dictionnaire le plus proche, calculé dans la phase de test du système de reconnaissance du locuteur. Le locuteur « approprié » correspond à une distorsion VQ minimale, il est donc sélectionné et vérifié [64].
Cette technique a l'avantage d'être simple et peu coûteuse en temps de calcul. Cependant, elle n'offre qu'une représentation discrète de la conversion.

2.4 Réseaux de neurones artificiels

2.4.1 Introduction historique

Les réseaux de neurones artificiels (ou Artificiel Neural Network en anglais : ANN), sont des systèmes informatiques qui s'inspirent du fonctionnement du cerveau humain pour apprendre. Ils présentent plusieurs aspects du connexionnisme, qui voulaient s'inspirer du cerveau humain dans ses qualités de distribution et du parallélisme du traitement de l'information, et ses capacités d'apprentissage. L'idée est alors de pouvoir effectuer presque toutes les fonctions arithmétiques. Il s'agit là d'une variété de technologie Deep Learning (apprentissage profond : DNN), qui fait, elle-même partie, de la sous-catégorie d'intelligence artificielle du Machine Learning (apprentissage automatique). Donc, Les réseaux de neurones artificiels permettent aux ordinateurs de résoudre des problèmes de façon autonome et renforcent leurs capacités d'une manière générale.
Le concept des réseaux de neurones artificiels fut inventé en 1943 par deux chercheurs de l'Université de Chicago [66], le neurophysicien Warren McCullough, et le mathématicien Walter Pitts. Les deux chercheurs présentent leur théorie selon laquelle l'activation de neurones est l'unité de base de l'activité cérébrale [66]. En 1949, Donald Hebb va contribuer à révolutionner la perception des neurones artificiels, il a proposé ce que l'on appelle désormais la règle de Hebb c'est lorsque deux neurones se déclenchent ensemble, la connexion entre les neurones est renforcée, cette activité est l'une des opérations fondamentales nécessaires à l'apprentissage et à la mémoire [67]. En 1958, le Perceptron fut inventé par le psychologue Frank Rosenblatt [68]. Il s'agit du plus ancien algorithme de Machine Learning, conçu pour effectuer des tâches de reconnaissance de patterns complexes. C'est cet algorithme qui permettra plus tard aux machines d'apprendre à reconnaître des objets sur des images, sa première implémentation était dans un logiciel pour l'IBM 704, il a ensuite été implémenté dans du matériel personnalisé sous le nom de « Mark 1 perceptron» [68]. Malgré cette évolution au niveaux théorique à l'époque, les réseaux de neurones étaient limités par les ressources techniques. En effet, cette approche connexionniste a atteint ses limites technologiques, par exemple, les ordinateurs n'étaient pas assez puissants pour traiter les données nécessaires au fonctionnement des réseaux de neurones, compte tenu de la puissance de calcul. C'est la raison pour laquelle la recherche dans le domaine des Neural Networks est restée en sommeil durant de longues années. à partir de 1982, les réseaux de neurones formels ont connu un regain d'intérêt ; ils sont devenus opérationnels depuis les années 1990, grâce aux progrès effectués dans la compréhension des systèmes non linéaires et aux performances accrues des ordinateurs. Le début des années 2010, avec l'essor du Big Data

et du traitement massivement parallèle, pour que les Data Scientists disposent des données et de la puissance de calcul nécessaires pour exécuter des réseaux de neurones complexes. En 2012, lors d'une compétition organisée par ImageNet, un Neural Network est parvenu pour la première fois à surpasser un humain dans la reconnaissance d'image [69]. Récemment, les réseaux de neurones connaissent un grand d'intérêt sous l'appellation d'apprentissage profond (Deep Learning). L'augmentation de la taille des bases de données, notamment celles des données issues d'internet, associée à la puissance de calcul disponible, permettent d'estimer les millions de paramètres du perceptron accumulant des dizaines voire des centaines de couches de neurones aux propriétés très spécifiques. A présent, les réseaux de neurones artificiels ne cessent de s'améliorer et d'évoluer de jour en jour. Ce succès est la conséquence des résultats spectaculaires obtenus par ces réseaux en reconnaissance d'image, de jeux, de la parole etc.[70],[71], [72].

2.4.2 Principe des réseaux de neurones

La conception des réseaux de neurones artificiels (ANN) s'appuie sur la structure des neurones biologiques du cerveau humain. En règle générale, un réseau de neurones repose sur un grand nombre de processeurs opérant en parallèle et organisés en tiers. Le premier tiers reçoit les entrées d'informations brutes, un peu comme les nerfs optiques de l'être humain lorsqu'il traite des signaux visuels. Par la suite, chaque tiers reçoit les sorties d'informations du tiers précédent. On retrouve le même processus chez l'homme, lorsque les neurones reçoivent des signaux en provenance des neurones proches du nerf optique. Le dernier tiers, quant à lui, produit les résultats du système. Il existe de nombreux modèles qui peuvent être utilisés dans la construction des réseaux de neurones artificiels [72]. Le plus simple est le modèle du perceptron, c'est un classifieur linéaire. Il s'agit d'un neurone formel muni d'une règle d'apprentissage qui permet de déterminer automatiquement les poids synaptiques de manière à séparer un problème d'apprentissage supervisé. C'est le même principe du neurone biologique (un exemple représentatif dans la Figure 2.3[1]).

De façon très réductrice, un neurone biologique est une cellule qui se caractérise par:

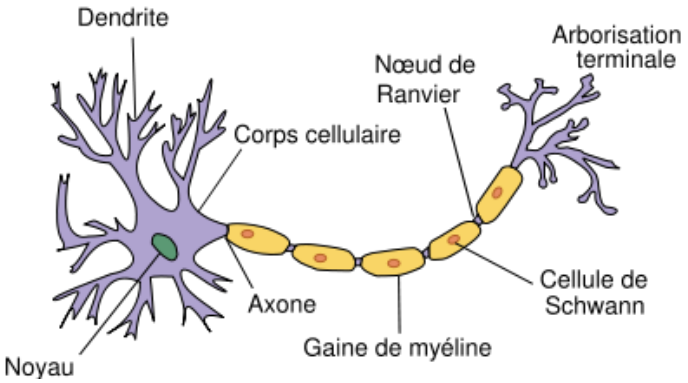

Figure 2.3: Schéma légendé de la forme d'un neurone biologique

- Des synapses, les points de connexion avec les autres neurones, fibres nerveuses ou musculaires;
- Des dendrites ou entrées du neurone;

[1] *ttps : //commons.wikimedia.org/wiki/File : Neuron − ro.svg*

- Les axones, ou sorties du neurone vers d'autres neurones ou fibres musculaires;
- Le noyau qui active les sorties en fonction des stimulations en entrée.

Par analogie, le neurone formel, est un modèle qui se caractérise par un état interne s ∈ S. Il prend la somme pondérée $(w_1, w_2, ..., w_n)$ de ses signaux de n entrée $(x_1, x_2, ..., x_n)$, une fonction d'activation et la sortie *(y)* est déterminée si la somme donnée dépasse une valeur prédéfinie valeur seuil (ou biais) (θ) ou non et elle est définie par la donnée de n poids (ou coefficients synaptiques) $(w_1, w_2, ..., w_n)$. Ceci est similaire au fonctionnement des neurones dans la nature (illustré à la figure 2.4.) : Le neurone reçoit son entrée via les récepteurs répartis sur ses dendrites, si le neurone se déclenche ou non dépend alors du potentiel du neurone s'il dépasse un seuil donné (θ). En d'autres termes, la sortie du neurone naturel est régulée par la somme de ses entrées et pouvant être utilisée comme l'entrée d'autres neurones. Il est défini par l'équation (2.9).

$$y = f(\sum_{i=1}^{n} w_i x_i) = \begin{cases} 1 & si \quad \sum_{i=1}^{n} w_i x_i > \theta \\ 0 & sinon \end{cases} \tag{2.9}$$

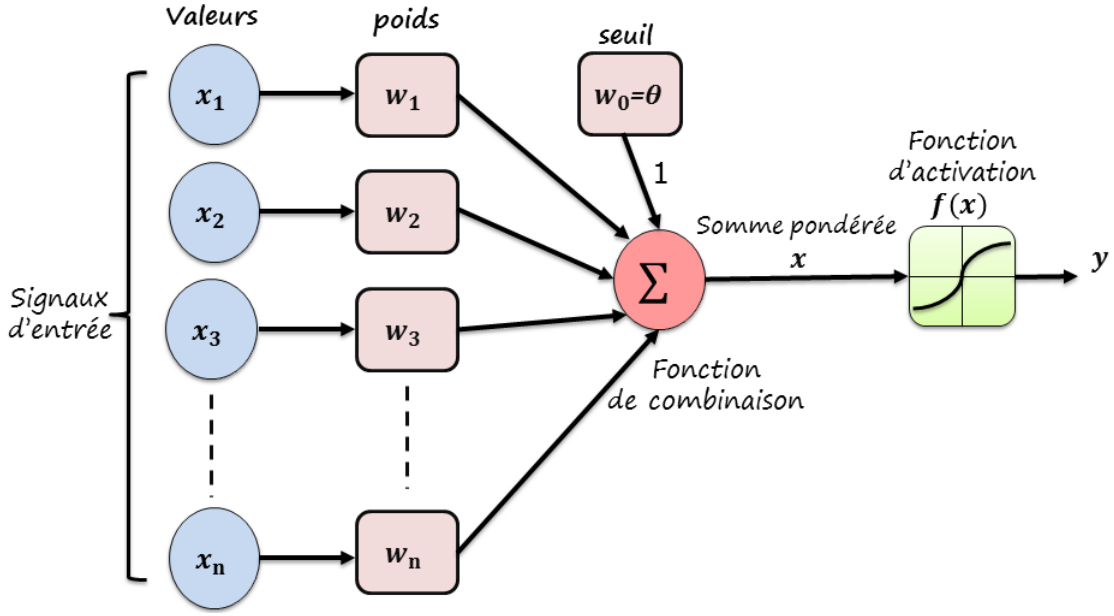

Figure 2.4: Structure d'un neurone formel

La figure 2.4 représente le fonctionnement d'un neurone formel : il s'agit d'une combinaison des déférentes valeurs d'entrée x_i, pondérées par des poids w_i appelés (poids synaptiques). La combinaison est opérée par une fonction spécifique, la sortie du neurone formel est conditionnée par une fonction d'activation f. Ainsi, la valeur de sortie du neurone y en fonction des valeurs d'entrée x_i et avec une fonction de combinaison une somme pondérée est trouvée en utilisant la formule (2.9). De nombreuses fonctions ont été proposées et utilisées comme fonction d'activation. Historiquement, les fonctions : *sigmoïde*, *softmax* et ReLU sont les fonctions d'activations les plus anciennes et les plus populaires, elles sont bien adaptées aux algorithmes des problèmes d'apprentissage impliquant[73], [74]. Les principaux types sont définies comme suit :

- linéaire f est la fonction identité (annulation de l'activation),
- La fonction sigmoïde $sig\,(x) = \frac{1}{1+e^{-x}}$, c'est une fonction qui produit une courbe en forme de

S. Bien que de nature non linéaire, il ne tient toutefois pas compte des légères variations des entrées, ce qui entraîne des résultats similaires.

- La fonction tangente hyperbolique : $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, Il s'agit d'une fonction supérieure comparée à sigmoide. Cependant, elle rend moins bien compte des relations et elle est plus lente à converger.
- La fonction ReLU $f(x) = max(0, x)$ (Rectified Linear Unit), Cette fonction converge plus rapidement, optimise et produit la valeur souhaitée plus rapidement. C'est de loin la fonction d'activation la plus populaire utilisée dans les couches cachées.
- La fonction *softmax* $f(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$ pour tout $k \in \{1...K\}$, utilisé dans la couche de sortie car elle réduit les dimensions et peut représenter une distribution catégorique.
- La fonction seuil $f(x) = 1_{[0,+\infty]}(x)$,
- ...

Les réseaux de neurones artificiels peuvent prendre différentes formes selon l'objet de la donnée qu'il traite, selon sa complexité et la méthode de traitement de la donnée, selon le type de la fonction d'activité (sigmoïde, échelon, fonction linéaire, …), le nombres des couches cachés, la fonction de combinaison utilisée (somme pondérée, distance pseudo-euclidienne…), l'algorithme d'apprentissage (rétro propagation du gradient, cascade corrélation, …)[73]. Le perceptron peut être vu comme le type de réseau de neurones le plus simple parce qu'il ne dispose que de deux couches ; la couche en entrée et la couche en sortie (Mono-couche). Lorsque, un réseau de neurones est composé d'une série des couches cachées, de sorte que tous les neurones de chaque couche précédente se connectent aux neurones de la couche suivante. Alors on parle du perceptron multi-couches (PMC) [73] (Feed-Forward ou Multi Layer Perceptron (MLP) en anglais).

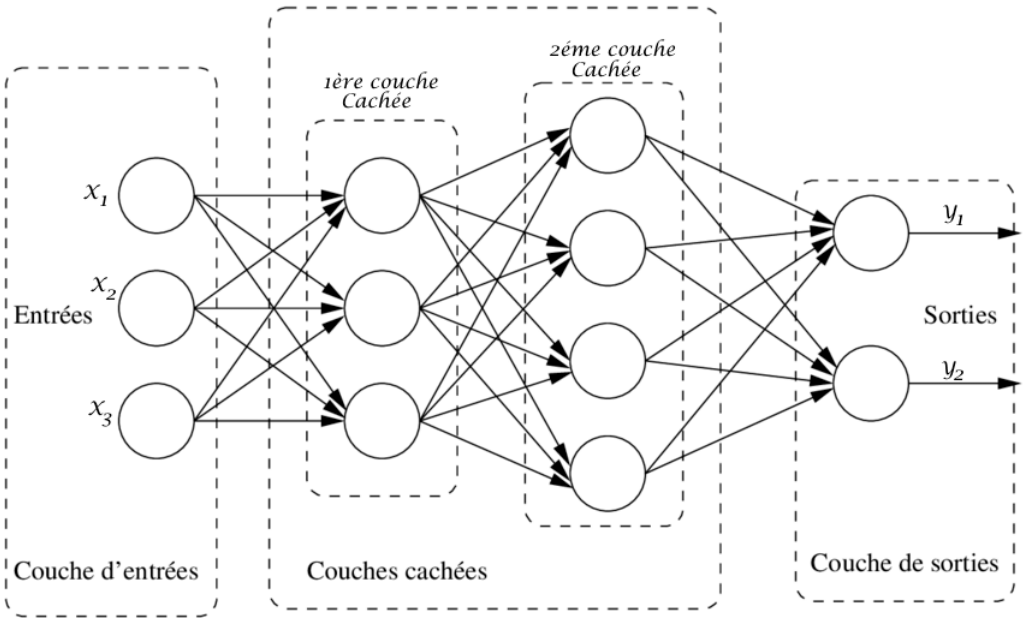

Figure 2.5: Exemple de perceptron multicouche élémentaire avec deux couches cachées

Dans la figure 2.5 nous avons présenté le schéma d'un exemple de réseau de neurones de 4 couches, les neurones de la dernière couche cachée transmettent les données de sortie en entrée des neurones de la couche de sortie. A partir de cela les neurones de la couche de sortie produisent les données de sortie finales.

2.4.3 Types des ANN et leurs application en RAP

Il existe de nombreux types de réseaux de neurones qui s'appliquent dans la reconnaissance de la parole. Parmi les autres types de ANN que le perceptron multi-couches, on peut citer:

- **Réseau de neurones récurrent – Recurrent Neural Network (RNN):** un réseau de neurones récurrent est un réseau dont le graphe de connexion contient au moins un cycle, il est possible de faire passer l'information dans des boucles de rétroaction, et ainsi de la faire revenir vers une couche précédente. Ces rétroactions permettent au système de se constituer une mémoire. Les réseaux de neurones récurrents peuvent fonctionner avec différentes longueurs d'entrée et de sortie et nécessitent une grande quantité de données, donc ce mode d'apprentissage est un peu plus complexe. Ils sont utilisés par exemple en matière de reconnaissance vocale, de traduction, l'analyse de sentiments, de reconnaissance d'écriture manuscrite, l'analyse de séquence ADN, la traduction automatique [71], [75]. Les réseaux de neurones récurrents étaient basés sur les travaux de David Rumelhart en 1986 [76] et de John Hopfield en 1982 [77]. En raison de ces capacités, les RNN, en particulier les LSTM, ont eu un impact énorme dans la reconnaissance vocale, par conséquent, ils sont incorporés dans les systèmes RAP récents [78].
- **Mémoire à court et long terme - Long-Short Term Memory (LSTM):** c'est une architecture particulière de RNN qui peut traiter des données ponctuelles uniques (par exemple une image) et des séquences complètes de données (par exemple la parole ou la vidéo) [75].
- **Réseaux de neurones de convolution – Convolution Neural Network (CNN):** Un réseau neuronal convolutif est un type de réseau multicouche. Il est composé d'un minimum de cinq couches. Il est procédé sur chacune de ces couches à une reconnaissance de motif. Le résultat obtenu sur chaque couche est transmis à la couche suivante. Ce réseau repose sur des filtres de convolution (matrices numériques). Les filtres sont appliqués aux entrées avant que celles-ci ne soient transmises aux neurones. C-à-d., au lieu d'utiliser des couches cachées entièrement connectées, le CNN introduit une structure de réseau spéciale, qui consiste en une alternance de couches dites de convolution. Ce type de réseau de neurones artificiel est utilisé en matière de reconnaissance d'images ou de l'audio-visuels. La couche de convolution dans les CNN agit comme un banc de filtres basé sur les données capables de capturer des représentations à partir de la parole [75], [79]. Nous avons utilisé ce type de réseaux de neurones dans nos travaux pour entraîner un réseau neuronal convolutif à reconnaître un ensemble donné de commandes[44].
- **Réseaux de neurones profonds - Deep Neural Networks (DNN) :** Les réseaux de neurones dits "profonds" (DNN) sont des perceptron multi-couches (PMC) avec un nombre de couches supérieur à trois. Les DNN se composent de plusieurs couches, y compris une couche d'entrée, des couches cachées et une couche de sortie, d'unités de traitement appelées « neurones ». Ces neurones dans chaque couche sont étroitement connectés aux neurones des couches adjacentes. Le but des DNN est d'approximer une fonction f. Par exemple, un classificateur DNN mappe une entrée x à une sortie se composent de nœuds y en utilisant une fonction de mappage $y = f(x; \theta)$et apprend la valeur des paramètres θqui résultent de la meilleure approximation de fonction. Chaque couche d'un DNN effectue un apprentissage de représentation basé

sur l'entrée qui lui est fournie. Par exemple, dans le cas d'un classificateur, toutes les couches cachées à l'exception de la dernière couche (***softmax***) apprennent une représentation pour les données d'entrée pour faciliter la tâche de classification. Un réseau DNN bien entrainé apprend une hiérarchie de représentations distribuées [75], [80]. Ces représentations se sont avérées très utiles dans la conception de différents systèmes basés sur la parole. Pendant longtemps les méthodes d'apprentissage pour ce type de réseaux de neurones acycliques ne permettaient pas de converger vers un réseau de neurones performant. Des avancées majeures sur les méthodes d'entraînement et le choix de la fonction de transfert (ReLU) qui minimise l'impact de la dilution du gradient dans les couches basses du réseau ont permis d'utiliser des réseaux de neurones de plus en plus gros. Historiquement, l'idée de réseaux de neurones profonds (DNN) est une extension d'idées émergeant de la recherche sur les réseaux de neurones artificiels (RNA) [75]. Les réseaux de Feed-Forward Neural Networks (FNN) ou perceptron multi-couches (PMC) avec plusieurs couches cachées sont en effet un bon exemple d'architectures profonds [75].

- **Auto-encodeur - Autoencoders (AEs):** Un encodeur-décodeur classique (ou auto-encodeur) ne cherche pas à passer d'une représentation en entrée à une représentation différente en sortie, ses prédictions consistent à reproduire ses propres entrées. L'intérêt de ce type d'architecture se trouve dans les représentations latentes apprises par l'auto-encodeur au niveau de la couche cachée, capables de capturer suffisamment d'informations pour reconstruire les entrées, ce type s'applique principalement à la détection d'anomalie et à la réduction de dimension [75].
- **Réseaux antagonistes génératifs - Generative Adversarial Network (GANs):** Les réseaux antagonistes génératifs ou GANs (Generative Adversarial Network), sont des algorithmes d'apprentissage non supervisé à base de réseaux de neurones artificiels, qui permettent de modéliser et d'imiter n'importe quelle distribution de données. Ils peuvent être utilisés dans différents domaines (traitement d'images, de texte, de sons, …). Depuis leur invention en 2014 par Goodfellow [81], les GANs ont suscité un grand intérêt et plusieurs chercheurs ont souligné leur potentiel. Ce type est un cadre théorique très puissant pour la génération de données et robuste au surapprentissage. Ils peuvent apprendre des représentations démêlées qui conviennent parfaitement à l'analyse de la parole [75].

Dans la littérature, l'utilisation des réseaux de nouerons en RAP a connu une grande progression. Hinton et al [82] ont donné un aperçu des progrès réalisés par quatre groupes de recherche différents qui ont obtenu de bons résultats dans l'utilisation de réseaux de neurones profonds dans la reconnaissance vocale. Les chercheurs ont fait des expériences avec la base de données TIMIT[2], le corpus de discours transcrits phonétiquement et lexicalement d'anglophones américains de sexes et de dialectes différents. Les résultats montrent que les DNN fonctionnent bien sur toutes les tâches par rapport aux systèmes basé sur HMM. Tous les groupes ont réussi à obtenir des meilleurs résultats sur la base de données TIMIT en utilisant un réseau hybride bayésien profond (DBN) / HMM-DNN avec huit couches cachées après avoir testé un nombre différent de couches de 1 à 8 couches avec le même nombre d'unités par couche en utilisant [512, 1024, 2048 ou 3072] unités. Ensuite, le modèle a été testé sur un ensemble de données différents et comparé ses résultats avec un modèle HMM / GMM hautement réglé.

[2] *https* : *//catalog.ldc.upenn.edu/LDC*93*S*1

Abraham[83] a proposé un modèle de RAP en utilisant une structure de réseau neuronal avec une mémoire à court et long terme (LTM) qui s'inspire de la mémoire à long terme du cortex humain. L'ensemble de données qu'a utilisé se compose de 8 800 échantillons d'énoncés recueillis auprès de 88 locuteurs, où chacun des locuteurs a répété chaque chiffre de 0 à 9 dix fois. La technique d'extraction de caractéristiques utilisée est MFCC pour produire une bonne représentation du signal vocal. Les caractéristiques extraites sont ensuite introduites dans le réseau neuronal avec des cellules LTM qui peuvent apprendre les séquences. Les résultats montrent que le modèle LTM développé avec les paramètres MFCC est précis à 99% dans la reconnaissance des ensembles de données vocales des chiffres arabes.

Ahmed Ali[84] a présenté dans sa thèse un système de reconnaissance de la parole multi-dialecte arabe. Il s'est basé sur un corpus arabique multi-dialectal des pays arabes (Maroc, Algérie, Tunisie, Egypte, …) collecté de la chaîne de télévision « Al Jazeera » et à partir de YouTube. Il est composé de cinq classes comprenant les dialectes (MGB-2, MGB-3, ...). IL a exploré deux groupes principaux de caractéristiques, à savoir les caractéristiques acoustiques et les caractéristiques linguistiques. Il a utilisé les classificateurs génératifs et discriminatifs, en plus des approches d'apprentissage en profondeur, à savoir le réseau de neurone profond (DNN) et le réseau de neurone convolutif (CNN). Le système global est une combinaison de cinq modèles acoustiques (AM): mémoire unidirectionnelle à long terme (LSTM), LSTM bidirectionnel (BLSTM), réseau neuronal à retard temporel (TDNN), couches TDNN avec couches LSTM (TDNN-LSTM) et enfin les couches TDNN suivies des couches BLSTM (TDNN-BLSTM). Leur système a obtenu une erreur absolue moyenne (MAE) de 12,3% WER et une erreur quadratique moyenne (RMSE) de 16,9% WER sur 1400 phrases. Le WER global estimé était de 22,9% pour l'ensemble de test de 3 heures, tandis que le WER réel était de 28,5%.

D'après cette brève revue de la littérature, et d'après l'article de synthèse de Wajdan Algihab et al [85] qui contient certains des systèmes proposés publiés au cours des 5 dernières années. La remarque principale presque de tous ces travaux c'est l'utilisation d'un corpus de petite taille, à l'exception de (MGB)[84] et (SAD)[86]. De plus, la majorité des études se concentrent sur l'étape de classification, et une attention limitée a été portée à l'étude des méthodes d'extraction de caractéristiques, bien que cette étape ait un impact significatif sur les performances du système. Les MFCC sont adoptés comme méthode d'extraction de caractéristiques dans presque tous les systèmes proposés. Une variété de techniques de classification a été appliquée, mais HMM et MLP sont les approches les plus largement adoptées. D'un autre coté, rarement, parmi les systèmes proposés la plupart sont concentrées sur l'étude de la robustesse des systèmes de RAP surtout dont les données vocales contaminés par le bruit.

Dans notre système de reconnaissance automatique de la parole arabe au milieu bruité, nous allons appliquer des DNN aux caractéristiques acoustiques des MFCC avec un nombre variable de couches cachées, des poids initiaux et des paramètres de réseau pour mesurer leur efficacité par rapport à d'autres techniques avec des donnes bruités dans la phase d'apprentissage. De plus, nous allons essayer des méthodes hybrides d'un DNN avec les modèles de Markov caché dans les autres simulations pour faire une comparaison au niveau de robustesse au bruits. Nous testons également d'autres types des réseaux de neurones à savoir le réseau de neurone convolutif (CNN).

2.5 Approche hybride : GMM-HMM

Le modèle de mélange gaussien (GMM) est un modèle statistique génératif capable de modéliser efficacement des caractéristiques acoustiques de la parole, tandis qu'un HMM est un modèle statistique capable de modéliser la séquence temporelle des signaux de parole [87]. Ainsi, la combinaison de ces deux composantes crée un modèle capable de décrire à la fois les caractéristiques spectrales et temporelles de la parole. L'utilisation du modèle hybride GMM-HMM pour la modélisation de la parole implique le choix d'une structure convenable. Ce modèle hybride est largement utilisé dans de nombreux systèmes de reconnaissance vocale. C'est l'un des modèles acoustiques les plus courants. Où chaque état HMM est associé à un GMM qui modélise les caractéristiques acoustiques pour cet état. Chaque Gaussien est un Gaussien multivarié sa taille dépend du nombre de MFCC dans le vecteur de caractéristiques [87].
Pour résumer, un modèle acoustique GMM-HMM est défini par les étapes suivantes pour chaque état HMM:

- Poids de pondération $\boldsymbol{w_{i,k}}$.
- Les vecteurs moyennes $\mu_{i,k}$et les matrices de variances $\Sigma_{i,k}$des Gaussiennes multivariées dans le GMM.
- Mélange des poids de Gaussiens multivariés dans le GMM.

Le GMM-HMM est construit avec la fonction de densité de probabilité des valeurs observées en utilisant le modèle de mélange gaussien basé sur la technique HMM originale, donc dans un modèle GMM-HMM, chaque état i est modélisé comme une somme pondérée de M Gaussiens, définie par l'équation précédente (2.4).

$$\boldsymbol{b_i(x_t)} = \sum_{\boldsymbol{k=1}}^{\boldsymbol{M}} \boldsymbol{w_{i,k}}\, N\left(\boldsymbol{x_t}, \mu_{\mathrm{i,k}}, \Sigma_{\mathrm{i,k}}\right) \quad ; \; \boldsymbol{i = 1, 2, ..., N} \; et \; \sum_{\boldsymbol{k=1}}^{\boldsymbol{M}} \boldsymbol{w_{i,k}} = \boldsymbol{1} \tag{2.10}$$

Dans l'équation (2.10), $N(.,\mu,\Sigma)$ est une distribution gaussienne multivariée avec le vecteur moyen $\boldsymbol{\mu}$ et la matrice de covariance $\boldsymbol{\Sigma}$, définie par l'équation suivante (2.11):

$$N(\boldsymbol{x};\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^{\mathrm{d}}\,|\Sigma|}} \exp\left\{-\frac{1}{2}(\mathrm{x}-\mu)^{\mathrm{T}}\,\Sigma^{-1}\,(\mathrm{x}-\mu)\right\} \tag{2.11}$$

où d est la dimensionnalité du vecteur de caractéristiques acoustiques: $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d}}$ Il faut noter que ce modèle est défini par l'ensemble des vecteurs moyens et des matrices de covariance de ses composantes ainsi que par le poids de chacune de ces composantes dans la somme pondérée. La figure 2.6 décrit un exemple de modèle acoustique hybride utilisant un GMM pour estimer les probabilités postérieures pour les états HMM.
Dans cette thése, nous avons choisi d'utiliser les caractéristiques utilisées dans la construction du modèle ayant les coefficients MFCC, de dimension égale à 39. Il est constitué des 13 premiers coefficients MFCC augmentés de leurs premières et secondes dérivées pour obtenir un vecteur de 39 caractéristiques par trame. Ce choix est basé sur les premier résultats expérimentaux qui ont démontré de meilleurs taux de reconnaissance (parti des résultats pour SRAP3). Le nombre d'états liés et le nombre de mélanges par état sont réglés à l'aide d'un ensemble de données d'entraînement.

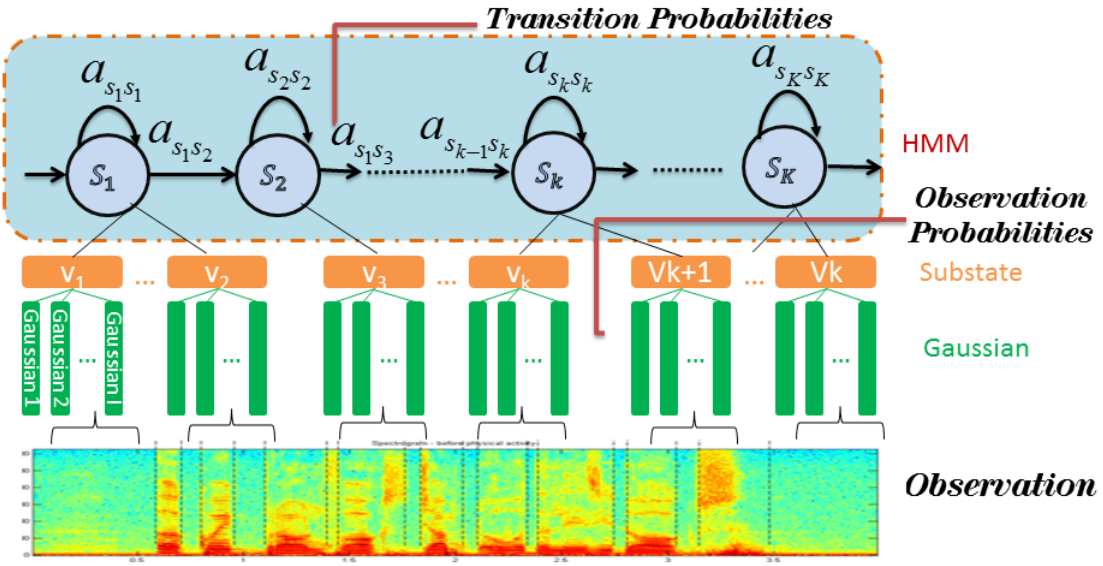

Figure 2.6: Architecture du système hybride GMM-HMM.

Ces traits caractéristiques ont été extraits en utilisant l'utilitaire PocketSphinx[3]. Nous avons évalué différents mélanges par état (4, 8, 16, 32, 64, 128 et 256). Plus le nombre de gaussienne est élevé, plus il est possible de modéliser les accents dans le modèle acoustique et bien sûre plus il faut des données pour entraîner le modèle acoustique. Le système GMM-HMM est utilisé dans la modélisation acoustique pour la reconnaissance des mots arabe à l'aide de la librairie PocketSphinx[3] développé par CMU.

2.6 Approche hybride : DNN-HMM

Récemment, les modèles DNN-HMM sont considérés comme l'une des techniques les plus couramment utilisées dans la modélisation acoustique des systèmes de RAP. Le système hybride qui combine le HMM et le DNN est adopté pour la modélisation acoustique en particulier dans la reconnaissance de la parole bruité, la plupart des études ont conclu que les modèles acoustiques DNN/HMM obtiennent de meilleures performances en comparaison avec les modèles acoustiques HMM/GMM dans de nombreuses tâches de reconnaissance de la parole, par exemple dans [80] ,[82],[88]. Une revue de la littérature des travaux récents utilisant le modèle hybride DNN-HMM et de l'approche basée sur l'apprentissage profond est donnée dans [75], [71], [80], [89], [90]. En 2011, Yu Dong et al. de l'Institut de recherche Microsoft ont proposé un modèle acoustique de Markov caché combiné à un réseau de neurones profonds basé sur le contexte qui est nommé context-dependent (CD) -DNN-HMM, ce modèle a réussi les tâches de reconnaissance de la parole continue à grand vocabulaire (LVCSR), et comparé aux systèmes GMM-HMM traditionnels, ce système a amélioré les performances de plus de 20%[91].

La différence principale entre le modèle DNN-HMM et GMM-HMM est l'utilisation de DNN (au lieu de GMM) pour estimer les probabilités d'observation. Dans le modèle acoustique, $p(x_t|s_t)$est la probabilité d'observation, qui est généralement représentée par GMM. La distribution de probabilité

[3] *https : //cmusphinx.github.io/wiki/tutorialpocketsphinx/*

postérieure de l'état caché $p(s_t|x_t)$peut être calculée par la méthode DNN. On utilise en fait le DNN pour modéliser $p(s_t|x_t)$, la probabilité postérieure de l'état donné au vecteur d'observation à, ce qui est possible puisque $p(s_t)$est facile à estimer à partir d'un alignement initial au niveau de l'état de l'ensemble d'apprentissage. Ces deux calculs différents de $P(X|S)$aboutissent à deux modèles différents, à savoir GMM-HMM et DNN-HMM[82], [90]. Le DNN est formé pour prédire les probabilités postérieures de chaque état dépendant du contexte avec des observations acoustiques données. Pendant le décodage, les probabilités de sortie sont divisées par la probabilité a priori de chaque état formant une « pseudo-vraisemblance » qui est utilisée à la place des probabilités d'émission d'état dans le HMM [82]. Une architecture générale du système hybride DNN-HMM est présentée à dans la figure 2.7. La figure 2.7 illustre l'architecture d'un modèle acoustique de type DNN/HMM. La

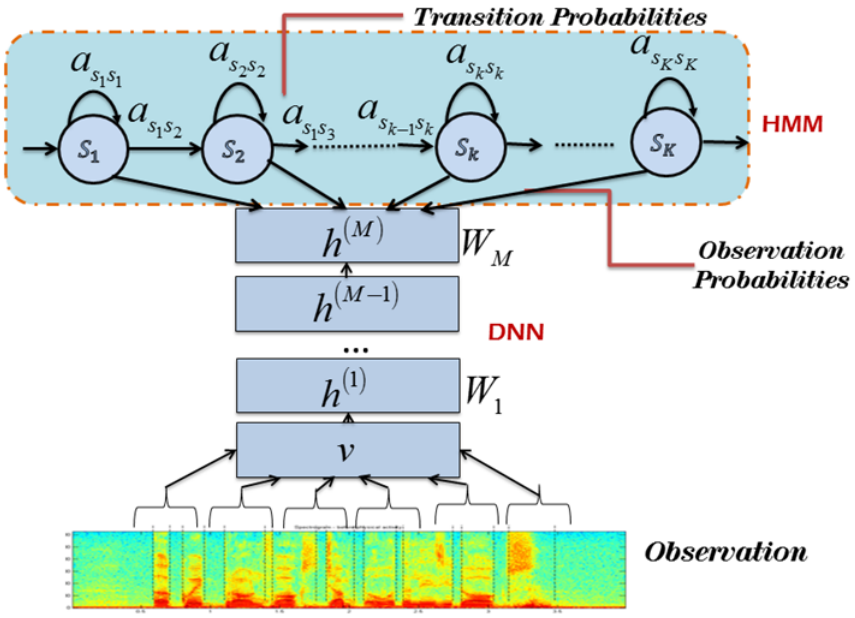

Figure 2.7: Architecture du système hybride DNN-HMM[92]

couche de sortie du DNN utilise la fonction *Softmax* pour calculer la probabilité de chaque état j du HMM connaissant l'observation x_t à l'instant t. Pour une étude détaillée à propos des modèles acoustiques DNN-HMM, il est intéressant de se reporter à [4]. Dans notre cas, nous présentons un ensemble d'expériences sur le HMM basé sur DNN pour la langue arabe, développé en utilisant la configuration DNN Kaldi de Karel. La première étape de l'apprentissage du modèle DNN-HMM consiste à entraîner le modèle GMM-HMM à l'aide des données d'apprentissage. Notre système de reconnaissance est fondé sur la boîte à outils Kaldi pour la modélisation acoustique basée sur le DNN qui comprend les étapes suivantes: extraction de caractéristiques (13 MFCC peuvent être utilisées comme caractéristiques), formation d'un modèle de monophone, formation d'un modèle de triphone avec des caractéristiques delta et formation du DNN-HMM final modèle. Les DNN sont formés de telle manière ou l'estimation des probabilités postérieures des états HMM à partir de l'observation donnée qui est représentée sur la figure 2.7.

2.7 Bases des données bruitées

Le bruit est généralement considéré comme un signal inutile. Un bruit attaque les informations concernant la source, l'environnement dans lequel il se propage. Les sources de bruit ou de distorsions sont de divers types, notamment [93]:

(a) **Le bruit électronique** qui comprend le bruit thermique et le bruit de tir.
(b) **Le bruit acoustique** qui se produit généralement en source vibrante ou en collision comme par les machines tournantes, le vent en mouvement, la pluie, etc.
(c) **Le bruit électromagnétique** est essentiellement l'interférence entre la réception de la voix, de l'image, des données et sa transmission, sur le spectre des radiofréquences,
(d) **Un bruit électrostatique** se produit en raison de la tension.
(e) **Bruit de quantification** lorsque les paquets sont perdus en raison de la congestion sur le réseau.

Donc, lorsque divers changements indésirables se produisent en raison des caractéristiques non idéales telles que l'écho, la réverbération, les réflexions multiples, etc., qui s'appelées distorsion du signal. Alors, sur la base de ses caractéristiques temporelles et de son spectre de fréquences, les bruits se classent en différent catégories.
Cette section décrit les types de bruit de fond utilisés pour fournir le bruit de fond pour la construction de nos bases de données lesquelles nous avons utilisés dans nos simulations.

2.7.1 provenant du corpus NOISEX-92

Nous utilisons dans notre thèse le corpus NOISEX-92 [94] pour bruiter artificiellement les données d'entraînement et de test. Le but de ce corpus est de fournir un ensemble de bruits standard pouvant servir de base de comparaison pour les différentes méthodes de traitement et de reconnaissance de la parole lorsqu'elles sont mises en présence de bruits additifs. Tel qu'indiqué, le nom de la base de données utilisée pour les signaux de bruits. Ce corpus est composé de 15 types de bruit d'une durée de 3min 56s chacun, représentés dans différents SNR. Il contient des bruits avec des statistiques stationnaires et non-stationnaires.
Le corpus Noisex-92 a été conjointement mis au point, en 1992, à partir du corpus Noise-Rom-0 par l'Institut TNO pour l'étude de la perception et par l'équipe de recherche sur la parole de la « Defense Research Agency » anglaise. Seuls certains bruits ont été sélectionnés par rapport à l'ensemble de ceux disponibles dans le corpus Noisex. En complément de ces bruits sont fournis des signaux de parole dans différentes conditions de bruits et, ce, pour tous les bruits du corpus : parole non bruitée et parole bruitée à des SNR de 18, 12, 6, 0 et -6 décibels. Tous les fichiers des bruits de la base Noisex-92 [94] sont enregistrés sous format '.wav' avec une fréquence d'échantillonnage de 20 kHz et quantifié à 16 bits. Ils ont été décimés à 8 et 16 kHz pour être correctement mélangés avec les segments de paroles. Le tableau 3.1 contient une brève description des différents types de bruit de la base Noisex-92.

Table 2.1: Description des bruits de la base Noisex-92

Type de bruits	Description
HF radio channel noise, rose et blanc (Pink and white).	Bruit générer par générateur de bruit blanc /rose analogique. HF bruit de canal radio hautes fréquences
Bruits de conversations (Babble)	Bruit de murmures de 100 personnes dans un restaurant.
Bruits d'usine (Factory)	Bruit d'une usine de production de voitures
Bruits de voiture (Volvo)	Bruit de voiture volovo340 à 120km/h en 4ème vitesse sur une route goudronnée
Divers bruits militaires (Buccaneer M109, F16, machine gun, destroyer,...)	Bruits d'avion de chasse, de char d'assaut et de mitraillette, char de combat,...

2.7.2 Bruits provenant du CHiME3

La base de données CHiME-3[7] a été développée dans le cadre du $3^{éme}$ défi CHiME[4] de séparation et de reconnaissance de la parole dans des environnements réels et contient environ 342 heures de discours et de transcriptions en anglais provenant d'environnements bruyants et 50 heures d'audio dans un environnement bruyant. Elle se compose de phrases de Wall Street Journal (WSJ0) prononcées par des locuteurs dans des environnements réels et enregistrées à l'aide d'une antenne de 6 microphones sur la même tablette. Le corpus d'origine comporte quatre catégories d'environnements : café (CAF), croisement de rue (STR), transports en commun (BUS) et zone piétonne (PED). Il est livré avec un outil de simulation de données, qui mélange des phrases WSJ0 non réverbérés originales au bruit de fond, garantissant la même distribution de rapport signal-sur-bruit que dans les enregistrements réels. Un signal de bruit multicanal réel issu du corpus CHIME-3[4] est ensuite ajouté avec un rapport signal-sur-bruit dans la plage SNR [-10 ; 10] dB. Ces données de fichier audio sont fournies sous forme de fichiers WAV stéréo 16 bits échantillonnés à 16 kHz. Nous choisissons parmi les bruits de données CHiME-3, trois environnements de fichiers de bruits de fond: (rue, café, zone piétonne) car le CHiME-3 combine des niveaux élevés de bruit de fond. Cela nous permet d'utiliser le bruit réel du corpus CHiME-3, ce qui rend notre corpus simulé particulièrement réaliste mais difficile.

2.7.3 Bruits provenant du Aurora-2

L'AURORA-2 est une base de données conçue pour évaluer les performances des algorithmes de reconnaissance de la parole dans des conditions bruitées. Il était basé sur une version des TIDigits originaux. Le groupe de travail ETSI STQ-AURORA DSR a préparé ces données. Le bruit est ajouté artificiellement à la base de données TIDigits propre pour évaluer et comparer les performances d'algorithmes robustes au bruit. Tous les bruits utilisés dans ces données ont été enregistrés à différents endroits: (train de banlieue, bavardage de personnes, voiture, salle d'exposition, restaurant, rue, aéroport, gare)[95]. Dans nos travaux, nous sélectionnons le fichier de bruit Babble et de bruit Rue d'AURORA-2 car ils ont des niveaux de bruit élevés.

[4] *http : //spandh.dcs.shef.ac.uk/chime_challenge/chime*2015/

2.7.4 Autres types des bruits

Nous avons également utilisé dans les derniers travaux, le bruit enregistré localement en s'appuyant uniquement sur le microphone du smartphone, nous avons utilisé le micro pour enregistrer des types réels de bruits lors de la construction DARIJA_MO [96]. Cette base de données multi-locuteurs a été enregistrée dans les environnements réels de la vie courante (bruitées : City Bus, Café, salle de photocopie, cour de faculté). Nous avons utilisé différents appareils smartphone pour enregistrer les fichiers audio (audio mono à un taux d'échantillonnage 44 kHz, 16-bit). Plus de détails seront présenté dans le chapitre quatre, qui sera consacré aux expériences et simulations.
Nous avons utilisé aussi un autre type spécial du bruit, c'est le bruit additif blanc gaussien (AWGN: Additive White Gaussian Noise en anglais), obtenu sous forme numérique à partir d'un générateur aléatoire, il est caractérisé par sa densité spectrale de puissance uniforme *N0/2*.
Un bruit gaussien suit une distribution gaussienne, caractérisée par une moyenne μ et une variance ^{2}. Les interférences large bande sont aussi prises en compte par le modèle du bruit blanc.
Comme détaillé dans le chapitre 4, nous aborderons le problème de la séparation de sources de parole distantes dans des conditions de bruit. Pour ce faire, nous avons créé des nouveau corpus en ajoutant déférents types du bruit réel aux plusieurs corpus originaux que nous avons élaboré dans différentes expériences pour notre thèse.

2.8 Boite à outils de RAP

2.8.1 PocketSphinx

CMU Sphinx est l'un des systèmes, open source, les plus populaires de reconnaissance automatique de la parole. Il est actuellement utilisé par les chercheurs et les développeurs dans de nombreux endroits dans le monde entier, y compris les universités, les instituts de recherche et de l'industrie. Le terme licence libérale de CMU Sphinx a fait de lui un membre important de la communauté open source et a fourni un moyen à faible coût pour les compagnies pour bâtir des entreprises autour de la reconnaissance de la parole. Il permet aussi dans le cadre de la recherche principalement, d'obtenir la transcription écrite de données orales. Avec un langage de programmation assez simple, basé sur des phonèmes, il permet d'obtenir des résultats prometteurs pour le développement d'applications libres[97].
Pocketsphinx est une librairie permettant d'intégrer la reconnaissance vocale dans des projets, écrit en language C à l'aide des fonctionnalités du projet open source, elle est développée à l'Université Carnegie Mellon (CMU Sphinx USA). CMU pocketSphinx est une librairie gratuite disponible à télécharger ; elle vise principalement à faciliter la construction des systèmes de reconnaissance vocale. Elle est basée sur les Modèles de Markov Cachés (HMM). En plus d'avoir des possibilités de décodage multiplateforme, il est possible de développer avec plusieurs langages. La boîte à outils CMUSphinx[5] se compose de : SphinxTrain qu'est un outil qui permet de créer son propre modèle acoustique et son propre modèle de langage, il contient la base Sphinx2-3-4 qui est une bibliothèque de support requise

[5] *https : //cmusphinx.github.io*

à la fois par SphinxTrain , de Cmudict qui permet la mise au point de dictionnaire de prononciation et enfin de PocketSphinx qui fournit une interface python aux bibliothèques CMU Sphinxbase et Pocketsphinx [98].
Pour installer Pocketsphinx sur un ordinateur mono-carte, certaines conditions préalables doivent être installées dans le système. Par exemple: compilateur cc, python, libasound dev, alsa utils, bison. Après avoir téléchargé et compilé le code source de Pocketsphinx, un programme de test utilisant le modèle de langage, le modèle acoustique et le dictionnaire par défaut a été testé. Pocketsphinx utilise des HMM et des GMM comme modèle acoustique et le modèle N-Gram comme modèle de langage. CMUSphinx a plusieurs modèles acoustiques et linguistiques tels que l'anglais américain, l'allemand, l'espagnol, l'indien, etc. mais pour le modèle arabe n'est pas disponible [49]. Par Conséquent, nous avons construit un modèle de dialecte arabe du Maroc, l'accent de région de Beni Mellal, comme une nouvelle langue pour CMUSphinx. Nous avons créé le modèle acoustique et le modèle de langage pour le corpus DARIJA. Le modèle acoustique a été formé sur des données principalement propres et bruyantes.
La figure 1.2 du chapitre 1 décrit l'architecture générale d'un système RAP. Le modèle acoustique, le modèle de langage et le dictionnaire sont les entrées données au module de reconnaissance.
Nous avons choisi d'utiliser Pocketsphinx dans nos recherches travaux de pour construire un système de RAP adapté au dialecte marocain, car il a pour avantage d'être acoustiquement entraîné sur une quantité très petite de données, il a également été utilisé en tant que système état de l'art pour une comparaison avec les autres outils open source. Donc, nous sommes intéressés à cet outil et nous allons l'étudier plus en détail dans le chapitre 4.

2.8.2 Kaldi

Kaldi est une boîte à outils open-source pour la reconnaissance de la parole, elle est écrite en C ++, autorisée sous la licence Apache v2.0. Elle a été lancé initialement par Povey et ses collègues le groupe des recherche de Johns Hopkins University. La boîte à outils Kaldi comprend des fichiers exécutables en C ++ et divers scripts Shell. Les codes sont très flexibles, modernes et simples[99]. La boîte à outils complète de RAP Kaldi est disponible sur le site Web de sourceforge[6]. Pour comprendre l'ensemble de la boîte à outils Kaldi, les détails sont disponible sur le site Web de Github[7]. Il prend en charge les transformations linéaires, la formation discriminante dans l'espace des fonctionnalités et les réseaux de neurones profonds. La boîte à outils de reconnaissance vocale Kaldi est utilisée pour l'entrainement et le test du système de RAP basé sur le corpus d'apprentissage. L'outil Kaldi fourni les modules nécessaires pour implémenter les deux composants essentiels d'un système de reconnaissance de la parole ; le premier c'est le modèle acoustique qui sera implémenté soit par un modèle de markov caché (HMM) ou soit par un réseau de neurone (DNN), le deuxième c'est le modèle de langage n-gram qui peut être sauvegardé au format ARPA et RNNLM. Il devra naturellement être entraîné sur un ensemble d'apprentissage approprié, comme dans le cas d'un modèle acoustique. La boite à outils Kaldi permet de décoder un flux de parole en combinant les deux modèles (acoustique et langage) à l'aide d'automates d'états finis pondérés (WFST). Les automates peuvent être créés

[6] $https://sourceforge.net/projects/kaldi/$
[7] $https://github.com/kaldi-asr/kaldi$

par n'importe quel outil comme Open-FST toolkit (par exemple)[99].
Ci-dessous, nous présentons un schéma simplifié de la structure de Kaldi. Les modules de la bibliothèque de Kaldi dépendent de deux bibliothèques externes : les bibliothèques d'algèbre linéaire (i.e. : BLAS/LAPACK) et la bibliothèque qui permet d'intégrer des transducteurs à états finis (i.e. : OpenFST). La classe « décodable » fait le pont entre ces deux librairies externes. Enfin, les modules plus bas dépendent d'un ou plusieurs modules du dessus. kaldi contient actuellement deux implémen-

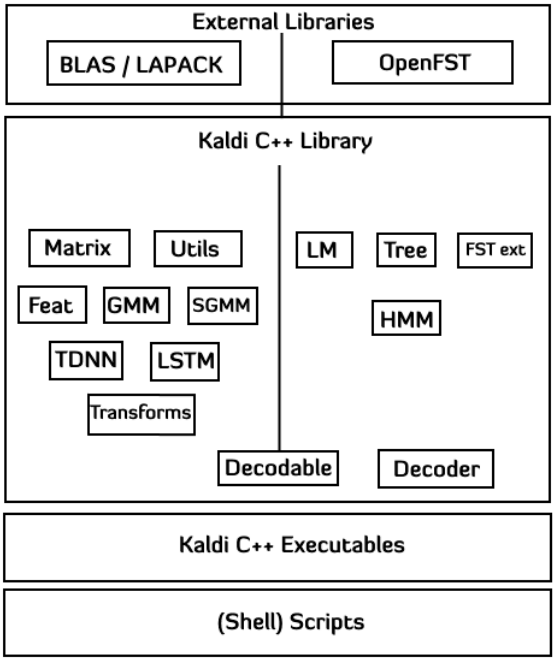

*Figure 2.8: Schéma simplifié de la structure des différents composants de **Kaldi**[99]*

tations parallèles pour la formation DNN. Le premier est que la configuration de Karel (nnet1) prend en charge la formation sur une seule carte GPU, ce qui permet la mise en œuvre d'être plus simple et relativement facile à modifier. La deuxième est que la configuration de Dan (nnet2) est plus flexible dans la façon dont vous pouvez vous entraîner: elle prend en charge l'utilisation de plusieurs GPU, ou de plusieurs processeurs, chacun avec plusieurs fils [99]. Dans notre cas, le système a été développé en utilisant la configuration DNN Kaldi de Karel. Les systèmes sont construits sur des coefficients cepstraux à l'échelle de Mel (MFCC), une analyse discriminante linéaire (LDA), une transformation linéaire à maximum de vraisemblance (MLLT), une régression linéaire à maximum de vraisemblance (fMLLR) avec des fonctionnalités de normalisation par la moyenne cepstrale (CMN). L'ensemble de DNN est exécuté dans un seul GPU à l'aide de CUDA (Compute Unified Device Architecture, l'architecture de calcul parallèle créée par NVidiaTM). Nous utilisons la boîte à outils KALDI pour construire le système hybride DNN-HMM, pour cela, nous nous sommes principalement concentrés sur le développement d'un système de RAP avec des données propres et bruitées artificiellement en créant un dictionnaire arabe.

2.8.3 HTK

HTK (boîte à outils pour modèles de Markov cachés ou Hidden Markov Model Toolkit en anglais) est un ensemble de librairies et de programmes en langage C et il s'exécute en ligne de commande, il est développé à l'origine à la machine intelligence du département d'ingénierie de l'Université de Cambridge sous la direction de S. Young à partir de 1989, c'est une boîte à outils portable permettant de créer et de manipuler des modèles de Markov cachés, elle est capable de mettre en œuvre un grand vocabulaire, indépendamment du locuteur et est applicable sur n'importe quelle langue. La documentation sur HTK est très riche avec des exemples pratiques (vers 300 pages). Depuis septembre 2000, HTK est distribué gratuitement sur leur site officiel[8] pour une utilisation non commerciale et maintenu par P. Woodland et ses collègues. HTK permet de faire des systèmes de reconnaissance de parole à petit ou moyen vocabulaire, avec une grammaire simple (par automate ou par bigramme). HTK est principalement utilisé pour la recherche sur la reconnaissance vocale. En effet, HTK lui-même est compilé pour construire l'ensemble d'un SRAP à partir de la base de données vocales, de l'extraction de caractéristiques, de la construction et de la formation de modèles acoustiques et de la réalisation de tests. On peut aussi l'utiliser pour autre chose que la parole (reconnaissance d'écriture manuscrite par HMM, par exemple). L'équipe de Cambridge a obtenu de très bons résultats en reconnaissance de parole sur les tâches « Wall Street Journal » et « Broadcast News », lors des évaluations américaines DARPA/NIST des années 90 et en transcription de parole spontanée [100]. Un décodeur plus performant permettant l'utilisation de trigrammes a ensuite été intégré [100].
L'utilisation de la boite à outils HTK pour mettre en oeuvre une application de RAP, induit trois phases principales. D'abord, la phase de préparation de données qui vise à enregistrer, étiqueter et segmenter des données d'apprentissage en utilisant l'outil HSLab. Suivi de l'extraction des vecteurs de caractéristiques après la configuration des paramètres avec l'outil HCopy.
La phase d'apprentissage sert à créer les modèles acoustiques qui représentent les vecteurs de caractéristiques. Durant l'apprentissage, les vecteurs sont d'abord utilisés pour l'initialisation des paramètres des HMMs en utilisant l'outil HInit et HCompV, car les paramètres des HMMs doivent être correctement initialisés. Ensuite, HRest est l'outil qui permet de réestimer les paramètres du modèle HMM (c'est l'implémentation de l'algorithme Baum-Welch).
Le processus de reconnaissance se fait avec l'outil HVite. Il existe d'autres outils dans HTK qui sont intéressants comme l'outil de calcul du taux d'erreur (WER) et le test des performances du système. La figure ci-dessous montre l'architecture de l'outil HTK (figure 2.9)[100]. HTK se distingue par l'utilisation exclusive des HMMs, il ne permet pas de faire des combinaisons ou d'hybridations des HMMs avec d'autres classifieurs. En dépit de la richesse de sa documentation, nous avons remarqué que quelques outils dans HTK ont moins d'information du point de vue pratique.
Dans le cadre de notre thèse, nous avons utilisé la boîte à outils HTK pour créer un modèle acoustique HMM qui contient la préparation des données nécessaires au traitement, le codage des données pour extraire la caractéristique des données vocales et la définition de divers paramètres HMM tels que la transition, la distribution et l'estimation. Ici, les résultats pour vérifier la modélisation HMM sont effectués en faisant varier la technique de codage par MFCC.

[8] *http : //htk.eng.cam.ac.uk*

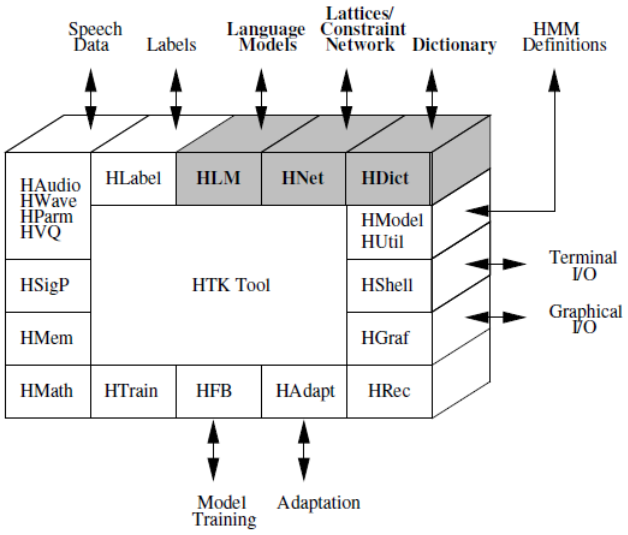

Figure 2.9: Architecture du HTK (d'après, Young et al [100]

2.8.4 MATLAB et la boîte à outils Deep Learning Toolbox

MATLAB ou est en fait l'abréviation de MATrix LABoratory, c'est un logiciel commercial de calcul interactif. Il est utilisé dans les calculs scientifiques basées sur des algorithmes d'analyse numérique et des problèmes d'ingénierie, en particulier ceux où les matrices interviennent, parce qu'il permet de résoudre des problèmes numériques complexes en moins de temps, grâce à une multitude de fonctions intégrées et à plusieurs programmes outils testés et regroupés selon usage dans des dossiers appelés boites à outils ou "Toolbox". MATLAB est contient une multitude de boites à outils spécifiques à des domaines variés. Elles regroupent un ensemble de fonctions spécifiques à un domaine d'application [9]. Matlab possède des boites à outils incluant des algorithmes d'apprentissage artificiel basés sur les modèles de Markov cachés ou réseaux de neurones, concernant le data mining: outils d'analyses factorielles outils de classification, arbres de décisions, régressions linéaires et non linéaires au sens large, des algorithmes de détection des séquences temporelles hors ligne et en ligne, il est disponible sur le site Mathworks [10].

MATLAB est un environnement dédié aux opérations de la manipulation vectorielles ou matricielles, permet d'accéder à des fonctionnalités de haut niveau de segmentation, de classification, des calculs statistiques, de prédiction et d'association, de la visualisation de données et images, de traitement du signal et vocale.

Parmi les boites à outils les plus utilisable, que nous avons utilisé dans les travaux pratiques de notre thèse:

Signal Processing Toolbox™: c'est une boite fournit des fonctions et des applications pour analyser, prétraiter et extraire des caractéristiques de signaux échantillonnés de manière uniforme et non uniforme, elle comprend des outils pour la conception et l'analyse de filtre, la réduction de la tendance et l'estimation du spectre de puissance, elle fournit également des fonctionnalités pour extraire des caractéristiques. Cette boite permet aussi de trouver des pics et des modèles de signaux, quantifier les similitudes de signaux et effectuer des mesures telles que le rapport signal

[9] *www.mathworks.com*
[10] *www.mathworks.com*

à bruit (SNR) et la distorsion, de prétraiter et analyser plusieurs signaux simultanément dans les domaines temporels, fréquentiels et temps-fréquence sans écrire de code, d'explorer de longs signaux et d'extraire des régions d'intérêt....

Audio Toolbox™ : propose des outils pour le traitement audio, l'analyse de la parole et les mesures acoustiques. Elle comprend des algorithmes pour le traitement du signal audio (comme l'égalisation et le contrôle de la plage dynamique) ainsi que pour les mesures acoustiques (comme l'estimation de la réponse impulsionnelle, le filtrage par bandes d'octave et la pondération perceptuelle). Des algorithmes sont également proposés pour l'extraction de caractéristiques audio et de parole (telles que les MFCC et le pitch) et pour la transformation du signal audio (banc de filtres gammatone et mel-spectrogramme, par exemple).

DSP System Toolbox™ : c'est un toolbox propose des algorithmes, des applications et des outils de visualisation pour concevoir, simuler et analyser des systèmes de traitement du signal dans MATLAB® et Simulink®. Grace à ces outils, nous pouvons modéliser des systèmes de traitement numérique du signal temps réel pour les systèmes de communications, radar, audio, les appareils médicaux, l'internet des objets et d'autres applications. Nous pouvons aussi concevoir et analyser des filtres FIR, IIR, multi-échantillonnés, multi-étages et adaptatifs. Nous pouvons récupérer des signaux à partir de variables, de fichiers de données et de matériel en réseau afin de développer et vérifier notre système. Les outils time scope, spectrum analyzer et logic analyzer permettent de visualiser et de mesurer les signaux en streaming de manière dynamique.

Deep Learning Toolbox™ : c'est un cadre dans les versions récentes de MATLAB telle que R2018a pour la conception et l'implémentation des réseaux de neurones profonds avec des algorithmes, des modèles pré-entraînés et des applications. Il facilite l'utilisation de MATLAB pour l'apprentissage en profondeur. Il prend en charge les réseaux de neurones à convolution (**MatConvNet**, CNN) et les réseaux LSTM pour effectuer la classification sur des données textuelles, images et de séries temporelles. Par conséquent, nous pouvons créer, analyser et entraîner des réseaux via une interface graphique. Aussi, il nous permet d'assurer le suivi des paramètres d'apprentissage, d'analyser les résultats et de comparer le code de plusieurs expériences. Vous pouvez visualiser les activations de couches et contrôler la progression de l'apprentissage sous forme graphique. Cette boîte à outils développée par Mathworks permet d'importer et d'exporter des modèles vers différents frameworks, y compris TensorFlow, Keras et PyTorch. Il prend également en charge l'optimisation GPU CUDA et permet de traiter des grands ensembles de données à l'aide de Parallel Computing Toolbox et en incluant NVIDIA GPU Cloud et des instances de GPU Amazon EC2 [11].

Il est conçu en mettant l'accent sur la simplicité et la flexibilité. Par exemple, pour les CNN, il expose les blocs de construction des CNN (ConvNets) en tant que fonctions MATLAB faciles à utiliser, fournissant des routines pour le calcul des convolutions avec des banques de filtres, la mise en commun des fonctionnalités, la normalisation, et bien plus encore. Pour cela, nous avons proposé d'utiliser les CNN pour former un modèle d'apprentissage en profondeur simple qui détecte la présence de commandes vocales arabes dans l'audio en se basant sur le projet [12].

[11] *www.mathworks.com*

[12] *https* : *//fr.mathworks.com/help/deeplearning/ug/deep − learning − speech − recognition.html*

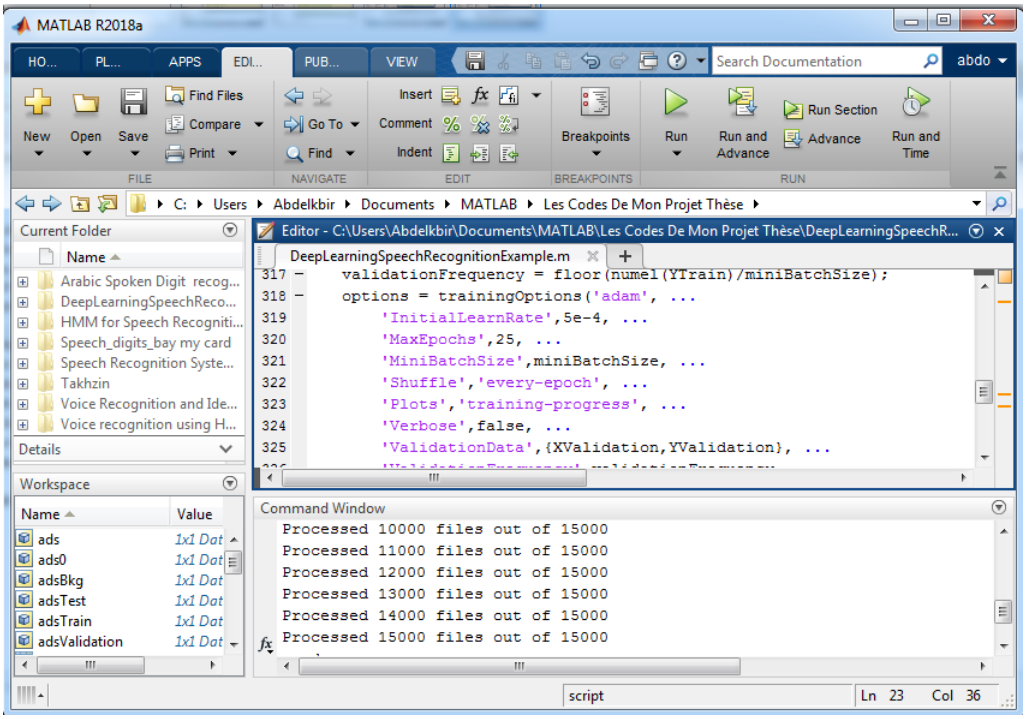

Figure 2.10: L'interface de l'environnement Matlab pendant l'entraînement du système (R2018a)

2.9 Conclusion

Dans ce chapitre, Nous avons présenté un panorama des techniques et des approches les plus courantes dans la littérature qui sont destinées à construire/améliorer des modèles acoustiques, sur lesquelles nous nous sommes appuyés dans notre thèse. De plus, nous avons présenté les différentes techniques d'apprentissage des modèles acoustiques probabilistes telles que : les modèles basés sur HMM, VQ, GMM, CNN ainsi que les modèles hybrides GMM-HMM et DNN-HMM. En outre, Nous avons présenté, aussi, les librairies et open source les plus utilisées dans les systèmes de reconnaissance automatique de la parole. Nous avons employé quatre boites à outils : PocketSphinx, HTK, Kaldi et Matlab. Enfin, nous avons décrit clairement les types de bruits issus des bases de données vocales universelles parmi eux : NOISEX-92, AURORA, CHiME-3 et les autres types qui nous appartient et que nous les avons enregistrés localement. Dans les chapitres suivants, nous expliquerons en détail comment ajouter ces bruits pour obtenir des bases de données modifiées que nous avons utilisées dans nos différentes expériences. De plus, nous testons ses nouveaux corpus pour les différentes techniques et les outils précédents. Dans le chapitre qui suit on va présenter aussi les méthodes que nous avons adopté pour notre système de reconnaissance automatique de la parole en milieu bruité pour augmenter la robustesse de notre système.

Chapter 3

Différents types de bruit et les techniques de débruitage adoptées dans cette thèse

3.1 Introduction

Notre environnement est souvent bruité, le signal de parole se trouve ainsi confronté au bruit ambiant. Pour le débruiter, on utilise des techniques de débruitage qui fonctionnent dans le domaine spectral (fréquentiel), à savoir, le filtrage adaptatif au sens de Wiener. La soustraction spectrale et d'autres types des algorithmes et techniques d'amélioration de signal de la parole. Dans ces familles de méthodes, le système de reconnaissance de parole est modifié afin de tenir compte de la présence d'un bruit lors de la reconnaissance. La réduction de bruit est devenue une nécessité dans les communications modernes ainsi que dans bien d'autres applications [101]. L'amélioration de la parole dégradée par le bruit, ou réduction du bruit, est un champ très important dans le traitement de la parole. Il est utilisé dans de nombreuses applications telles que les téléphones mobiles, les systèmes de téléconférence, la reconnaissance de la parole et les aides auditives [101].
D'autre part, l'utilisation de critères discriminants d'apprentissage permet de lutter contre la source de variabilité introduite par le bruit. L'apprentissage dans différents conditions prédéfinies de bruit reste une solution efficace, bien que difficilement réalisable en pratique, comme nous allons voir dans le chapitre 4. Avant l'étude, nous nous pencherons sur ces techniques pour identifier les différents types de bruit et leurs caractéristiques.
Donc, ce chapitre présente les différents types des bruits qui peuvent être mélangés avec des segments de la parole. Ainsi, dans ce chapitre nous essayons de décrire le signal de bruit et les traitements qui permettent sa suppression ou sa réduction. Après avoir parlé de la nécessité de l'opération de débruitage, la section 3.3 présente les différents algorithmes de débruitage du signal de parole, nous expliquons l'essor de ces techniques: La méthode de la soustraction spectrale, la méthode de filtrage de Wiener. Les expériences avec ces différents algorithmes et une étude comparative entre eux en termes du taux d'améliorations du SNR est effectuée dans le chapitre suivant. Nous présentons aussi quelques outils de mesures d'évaluation et les méthodes des mesures dans le domaine temporel

(rapport signal / bruit (RSB /SNR)). Enfin, nous terminerons notre chapitre par une conclusion qui résume les points abordés.

3.2 Le bruit sonore

La notion bruit est assez subjective on ne peut pas la définir simplement dans l'absolu. Mais on peut l'expliquer par des exemples dans la réalité. Car un bruit pour quelqu'un c'est une voix sans signification et pour un autre c'est un signal sonore utile. Comme par exemple ; deux personnes qui s'intéressent à écouter à l'un des deux autres personnes, chacun de ces derniers est un bruit pour l'un et un signal utile pour l'autre. Par conséquent, on peut définir un bruit comme tous les phénomènes qui empêchent la transmission d'un message d'une source à sa destination par la détérioration de la qualité et l'intelligibilité du message transmis. Un facteur important dans l'appréciation est la question de savoir si les bruits sont permanents ou entrecoupés de phases de silence. Des événements isolés périodiques (comme des trains qui passent toutes les heures) sont moins vite ressentis comme du bruit bien qu'ils soient vraisemblablement plus intenses que des nuisances permanentes (la circulation sur une route principale, par exemple) [101].
On ne peut pas mesurer le bruit comme on mesure le son, mais uniquement l'évaluer dans une fourchette déterminée. Physiquement, il n'y a pas de distinction entre le son et le bruit: le son est une perception sensorielle évoquée par des processus physiologiques dans le cerveau auditif. Le motif complexe des ondes sonores est perpétuellement étiqueté comme bruit, musique, parole, etc. Par conséquent, il n'est pas possible de définir le bruit exclusivement sur la base des paramètres physiques du son. Au lieu de cela, il est courant de définir le bruit simplement comme un son indésirable. Cependant, dans certaines situations, le bruit peut nuire à la santé sous forme d'énergie acoustique. Sur le plan scientifique, le son et le bruit sont techniquement identiques – il s'agit de vibrations dans l'air (ou dans l'eau) que nous captons avec nos oreilles. Plus les ondes sont grosses (l'amplitude), plus les vibrations sont fortes et plus le son est puissant. Cependant, le son désigne ce que nous entendons en général. Le bruit est aussi quelque chose que nous entendons, mais sans vouloir nécessairement l'entendre [101].
En résumé : Le son est la vibration mécanique d'un milieu gazeux, liquide ou élastique à travers lequel l'énergie est transférée loin de la source par des ondes sonores progressives. L'audition est simplement des fluctuations de pression d'air détectées par l'oreille. Le bruit a une qualité subjective et est souvent défini comme un son indésirable [102]. Le son se mesure en décibels (dB).

3.3 Caractérisation du bruit

Le bruit sonore est une oscillation de l'air qui, frappant le tympan, est interprétée par l'oreille et le cerveau. On parle en général de bruit pour les sons non nécessaires et qui déplaisent. Une onde sonore peut éprouver des difficultés à passer d'un milieu à un autre car chaque milieu impose une résistance plus ou moins importante appelée impédance. Ainsi une onde qui se propage dans l'air aura du mal à être perçue dans l'eau car celle-ci à une forte impédance.

Le son ou bruit est caractérisé par trois caractéristiques physiques importantes :sa fréquence, sa vitesse de propagation et son amplitude [103] [104].

- La fréquence correspond au nombre de vibrations par seconde (On parle de sons graves (de basses fréquences) ou de sons aigus (de hautes fréquences));
  - Exprimée en hertz (Hz);
  - Les fréquences entendues par l'humain s'étendent de 20 Hz à 20 000 Hz;
  - Le bruit est généralement composé de plusieurs fréquences;
- L'intensité dépend de l'amplitude de la vibration (son faible ou fort);
  - Exprimée en décibels (dB);
  - L'intensité du bruit (niveau sonore) correspond aux variations de pression plus ou moins importantes dans l'air ambiant;
- La vitesse du son/bruit ou célérité du son/bruit c'est la vitesse de propagation des ondes sonores qui dépend de la nature du milieu dans lequel l'onde se propage mais également de la température.
  - L'onde acoustique se propage dans l'air à 340 m/s, dans l'eau à 1500 m/s et à des vitesses encore supérieures dans les matériaux plus denses (3500 m/s dans l'os et jusqu'à 6000 m/s dans l'acier). Dans le vide, dépourvu de matière, aucun son ne se propage.

  On peut citer des autres propriétés liées à la propagation des sons:

  - **La durée** (son continu, intermittent ou impulsionnel (tel que les bruits d'impacts)). Lorsqu'une onde sonore est émise, elle tend à être modifiée par des paramètres tels que la distance ou d'éventuels obstacles.
  - **Atténuation**. En champ libre, c'est-à-dire dans un espace où aucun obstacle ne perturbe la propagation de l'onde sonore, son intensité acoustique diminue quand on s'éloigne de la source sonore.

  - **Réflexion**. Lorsqu'une onde sonore rencontre un obstacle tel que la paroi d'un local, une certaine quantité de l'énergie est réfléchie et revient dans la pièce : c'est la réflexion. Les réflexions successives constituent la réverbération. Quand il se réfléchit sur un obstacle (un mur, un rideau d'arbres…), on parle d'un "écho" : on entend le bruit atténué mais tel qu'il était.
  - **Absorption.** Une autre quantité de l'énergie est absorbée en partie par les matériaux constituant la paroi : les hautes fréquences étant plus facilement atténuées que les basses.
  - **Transmission.** Une partie de l'énergie est transmise dans la pièce voisine par le biais de la paroi, qui agit comme une source sonore secondaire.

Le son/bruit est constitué de vibrations. Ce sont les ondes sonores qui se diffusent comme des vagues à travers un milieu donné. Elles ne modifient pas le milieu dans lequel elles passent. L'onde comprime puis décomprime le milieu. Le haut-parleur utilise ce mécanisme. Comme pour les ronds dans l'eau qui se propagent, c'est la compression qui se déplace et non les molécules d'air. Plus ce milieu est dense, plus les ondes vont vite : à 20 °C la propagation est de 5 000 mètres à la seconde dans l'acier,

1 525 mètres à la seconde dans l'eau et 334 mètres à la seconde dans l'air. On pense souvent le contraire, mais la propagation des ondes sonores est bien cinq fois plus rapide dans l'eau que dans l'air.
Dans l'espace, où règne un vide sans fin, à l'exception des planètes qui ont une atmosphère, il n'y a pas de molécules pour porter ces ondes (l'absence de matière). C'est le silence absolu. Donc, dans le vide, les ondes sonores en générale ne peuvent pas se propager [103].

3.4 Le rapport signal sur bruit

Afin d'évaluer les performances des processus de robustesse, il faut disposer d'une mesure, d'un indice qui permet de rendre compte de la difficulté que présente un environnement du point de vue de la reconnaissance. L'un des critères d'évaluation de la qualité du signal de parole le plus populaire c'est le rapport signal sur bruit ou SNR (SNR, *Signal to Noise Ratio* en anglais) c'est le facteur de caractéristique physique le plus efficace pour évaluer le spectre de puissance dans les zones de non-parole et de parole, puis de prendre le log décimal du rapport. Le SNR se mesure donc en décibels (dB) et décroissent lorsque la composante de bruit est plus présente. Le rapport signal sur bruit S/N est une méthode pour mesurer la force du signal par rapport aux niveaux de bruit de fond, en utilisant l'équation (3.1) :

$$SNR(dB) = 10 \times log_{10}(\frac{P_{speech}}{P_{noise}}) \tag{3.1}$$

Avec, P_{speech} et P_{noise} désignent la puissance du signal de la parole et du bruit (de la non-parole), respectivement [105].
Un exemple de la corruption du signal propre avec un bruit gaussien blanc additif (AWGN) est donné à la figure 3.1. La figure 3.1 montre les spectrogrammes d'un mot prononcée dans milieu bruyant

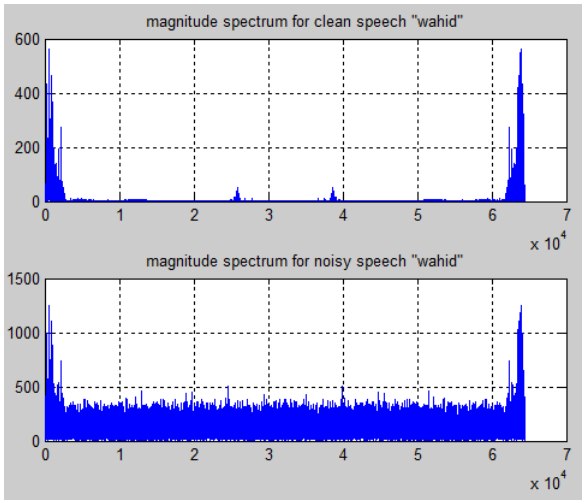

Figure 3.1: Spectrogrammes d'amplitude du signal original et le signal bruyant du mot ***"Wahid"*** *enregistré en SNR = 5 dB*

(bruit de la voiture) d'une voiture. L'axe des abscisses représente le temps, celui des ordonnées porte les fréquences et la nuance de gris code l'amplitude. Le premier spectrogramme (originale) a été obtenu grâce à l'enregistrement d'un homme par un microphone proche de la bouche. On peut donc

considérer que cet enregistrement est propre, on néglige le bruit produit par microphone.
Le deuxième spectrogramme (bruité), c'est le même signal de parole précèdent mais il est corrompu par un bruit additif (le bruit gaussien blanc additif (AWGN)). Le signal de bruit est visible en continu sur les basses fréquences. Ce sont donc les indices de parole situés dans le bas du spectre qui vont être recouverts par le spectre de bruit. On peut voir par exemple que la partie basse du spectre se confond avec un bruit de fond continu comme ici et que les passages entre unités acoustiques élémentaires sont généralement moins bien définis dans le bruit.
Les zones de parole et de non parole sont déterminées en comparant l'amplitude du signal par rapport à un seuil. Le SNR ne constitue qu'un moyen de quantifier l'influence du bruit sur la parole. Mais il ne permet pas de qualifier complètement la difficulté qu'un environnement bruité représente du point de vue de la reconnaissance. Par exemple, il ne permet pas de rendre compte de la dynamique du bruit, de la façon dont il varie sur une courte période. En effet, un bruit (additif ou convolutif) peut être stationnaire, évoluer lentement ou encore inopinément.
Il est cependant possible de rendre compte des plages de spectres influencées par le bruit en déterminant des RSB par bandes de spectres. Ainsi, il est possible de voir quels types d'indices acoustiques seront corrompus par un environnement particulier. Cela est particulièrement utile dans les méthodes de réduction du bruit, voir en section 9.1.

3.5 Le modèle d'environnement

Avant de discuter les techniques de robustesse, examinons d'abord nos connaissances sur l'effet du bruit sur la reconnaissance vocale. Plus précisément, nous décrirons l'effet du bruit dans les domaines du banc de filtres et des cepstraux sur la base d'un modèle d'environnement populaire qui caractérise la relation physique entre la parole et le bruit. Comme nous le montrerons dans les sections suivantes, les techniques robustes au bruit qui utilisent un modèle d'environnement sont appelées techniques basées sur un modèle, tandis que d'autres qui n'utilisent pas les modèles d'environnement sont appelées techniques basées sur les données. Dans cette section, nous utiliserons un modèle d'environnement couramment utilisé dans le domaine de la reconnaissance vocale robuste au bruit [106]. Il existe deux types courants de bruit qui peuvent affecter le signal vocal, ce sont les bruits additifs et les bruits convolutifs. Des exemples de bruits additifs sont le bruit de fond, le bruit de la circulation, etc., et le bruit de convolution peut être des distorsions du canal de transmission, le filtrage du microphone, la réverbération de la pièce, etc. Le modèle d'environnement que nous allons discuter est montré sur la figure 3.2, où le signal de parole propre est d'abord déformé par le canal, puis corrompu par le bruit additif plus loin [106].
Nous dérivons maintenant la représentation mathématique du modèle d'environnement dans les domaines temporel et fréquentiel. Soit $x(t)$, $n(t)$ et $y(t)$ la parole propre numérique, le bruit additif et la parole dégradée dans le domaine temporel respectivement où t est l'indice d'échantillonnage temporel, et soit $h(n)$ la réponse impulsionnelle du canal. Le modèle d'environnement dans le domaine temporel est:

$$y(t) = x(t) * h(t) + n(t) \tag{3.2}$$

Avec :

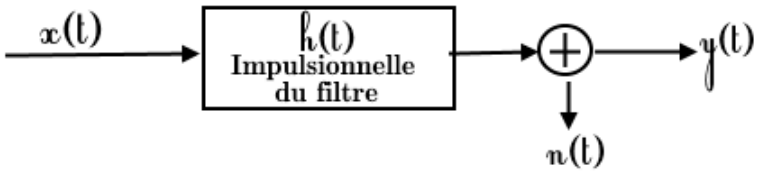

Figure 3.2: Le modèle d'environnement avec bruit additif et l'impulsionnelle du filtre.

- $x(t)$ le signal de parole propre
- $y(t)$ le signal bruité
- $n(t)$ le signal de bruit additif
- $h(n)$ la réponse impulsionnelle du filtre représentant les sources de bruit convolutif.

Et $*$ représente l'opérateur de convolution.
Dans le domaine cepstral le bruit de canal n'est plus convolué au signal propre mais il résulte que le signal propre et les sources de bruit sont liés de façon non-linéaire. On peut alors définir la fonction d'environnement f dans le domaine cepstral :

$$y = x + f(x, h, n) \tag{3.3}$$

où x, h, n et y sont les représentations respectivement de $x(t)$,$h(t)$, $n(t)$ et $y(t)$dans ce domaine.
Notez que nous ne considérons pas le cadrage et le fenêtrage dans le processus d'extraction d'entités par souci de simplicité. On peut considérer le signal dans (3.2) comme une trame. En appliquant la transformée de Fourier discrète, le modèle en domaine fréquentiel pour une seule trame devient :

$$Y(k) = X(k) H(k) + N(k) \tag{3.4}$$

Avec $k = 1, ..., K$ est l'indice de coefficient de Fourier et K est le nombre de coefficients de Fourier, $Y(k)$, $X(k)$, $N(k)$ et $H(k)$ sont les coefficients de transformée de Fourier de $y(t)$, $x(t)$, $n(t)$ et $h(t)$ dans la trame courante respectivement.
De nombreuses techniques de robustesse reposent sur une approximation de la fonction d'environnement. Cependant, la plupart des approches simplifieront la fonction d'environnement en limitant leur domaine d'application à des bruits additifs (par exemple la soustraction spectrale) ou convolutifs (par exemple la normalisation cepstrale). D'autres encore nécessiteront des informations a priori précises sur la nature du bruit, sous forme de code-book par exemple (comme le filtrage optimum probabiliste). Dans la section 3.9 nous présentons quelques techniques de robustesse qu'on a utilisé dans cette thèse.

3.6 Les types de bruits

Les différents bruits affectant un message peuvent être divisés:

- **Selon la nature du bruit** : Elle contient, elle-même, trois catégories :
  - **Bruit à bande large :** Qui touche une bande fréquentielle un peut important (large bande), à titre d'exemple on trouve le bruit blanc.

- **Bruit à bande étroite :** Qui touche une bande fréquentiel limité par un intervalle (qui a un spectre de raies bien défini), dans ce cas on trouve par exemple le bruit des outils de travail des ouvriers tel que le bruit du petit matériel électrique. Les bruits produits par les moyens de transports se caractérisent par une très forte stationnarité qui correspond à la vitesse de fonctionnement des organes moteurs.
- **Bruit intermittent dans le temps (bruit impulsionnel)** : C'est un signal qui s'annule dans certains intervalles de temps. Ce bruit nous aide à l'extraction du signal de parole pendant ces intervalles.

- **Selon les caractéristiques de stationnarité du bruit :** Elle contient aussi trois classes;
  - **Bruits stationnaires ou quasi-stationnaires :** On trouve ce type dans les organes moteurs fonctionnant moins vite. Ce type de bruit affecte beaucoup plus les voyelles dans un signal de parole.
  - **Bruits rythmiques :** Ce sont des bruits correspondants à la répétition d'une tâche de nature productive. Ils sont très souvent des bruits périodiques, produits par les systèmes industriels. Ce bruit est assez intense, mais l'énergie des raies n'est pas beaucoup plus importante. Il est, en outre, possible d'observer la présence de bruits aléatoires supplémentaires en basses fréquences, entre 0 et 3000 Hz. Ce bruit est cependant très étendu.
  - **Bruits aléatoires :** Ce sont des bruits associés aux ateliers de fabrication et aux usines. Ces bruits peuvent générer de nombreuses erreurs dans les systèmes de Reconnaissance Automatique de Parole (RAP). L'univers de la production est généralement très bruyant. Un exemple des bruits aléatoire est le bruit de soudures qui correspond à des raies verticales à des instants aléatoires dans le temps.

- **Selon le mode d'interférence :** C'est la classification la plus importante, qui nous intéresse. Il y a deux types principaux ; les bruits additifs et les bruits convolutionnels (multiplicatifs).
  - **Les bruits additifs:** Les bruits additifs sont dus à la multiplicité des systèmes de communication dans un même environnement. Plusieurs émetteurs et plusieurs récepteurs pouvant être confinés dans un même espace, les messages de tous les émetteurs peuvent donc se trouver en concurrence sur une même voie sans que les récepteurs possèdent un mécanisme efficace pour isoler le message qui leur est destiné. L'émetteur et le récepteur peuvent aussi se trouver en présence d'un ou de plusieurs équipements générant un bruit de fond de puissance variable. Les bruits additifs peuvent être subdivisés en trois groupes en fonction des lieux où ils peuvent être rencontrés:

    -Bruits des systèmes industriels : Ils peuvent être très intenses et sont, par nature, non stationnaires. Ils correspondent aux bruits émis par des machines possédant une faible isolation phonique.

    -Bruits des moyens de transport : Ils correspondent aux bruits qui peuvent être observés dans divers véhicules tels que les voitures, les trains et les avions…etc.

    -Bruits des milieux administratifs et urbains : Ce sont les bruits présents dans les bureaux, les domiciles ou dans les concentrations urbaines. Ces bruits peuvent être très variés (climatisation, bruit de parole) mais sont peu intenses.

Il est également possible de classer, dans ce type des bruits produits par les systèmes industriels. Ce sont les bruits qui peuvent être rencontré dans les systèmes de ventilation, les machines à écrire ou les ordinateurs. à cette liste peuvent être ajoutés les bruits de mobiles par rapports à l'auditeur tels que les voitures.

- **Les bruits convolutionnels** Les bruits convolutionnels (ou multiplicatifs) sont dus à la distorsion induite par la voie de communication. Ils résultent de la mauvaise qualité d'un ou de plusieurs éléments de support du message ou de son étroitesse en bande passante. Les moyens de communication à longue distance (la téléphonie, la radiophonie et la radiotéléphonie) sont élaborés à partir d'un compromis coût/efficacité. La parole lorsqu'elle est transmise est forcément dégradée en qualité et en intelligibilité. Un des champs possibles d'application de la RAP sont les serveurs vocaux accessibles par les lignes téléphoniques. Mais la parole transmise par téléphone souffre de déformations variables induites par la qualité de la connexion. Une transmission peut ainsi souffrir de l'étroitesse de la bande passante, de la mauvaise qualité des microphones de certains terminaux téléphoniques. La parole enregistrée dans tous les corpus utilisés pour la recherche est, en effet, toujours bruitée puisque le microphone utilisé effectue toujours un filtrage linéaire [107].
- **Les bruits physiologiques**

D'autres bruits peuvent également être considérés dans le domaine du traitement de la parole, mais ils n'ont pas la généralité des bruits de type additifs ou convolutionnels car ils sont spécifiques à l'être humain lors de sa phase de production de parole. La plupart des systèmes de RAP fonctionnent mal en milieu bruité car les contraintes posées par de tels environnements n'ont pas été prises en compte dès le départ. L'homme essaie, lui, de s'adapter aux conditions sonores rencontrées en modifiant sa méthode de production de parole. Un des phénomènes le plus remarquable de modification de production de la parole par l'homme est l'effet Lombard. Lorsqu'un locuteur est placé dans un environnement bruité, il modifie sa voix, et son effort vocal, en "haussant le ton" de manière à ce que la parole produite conserve un bon SNR par rapport à l'environnement. Cette accentuation de la voix pose cependant un problème majeur aux systèmes de RAP car les spectres de tous les phonèmes peuvent être modifiés ce qui a pour effet de nettement amoindrir les taux de mots isolés ou de la parole continue masqués par du bruit. Il faut enfin noter qu'il existe des situations où la parole est modifiée sans que l'homme ne modifie sa façon de parler de manière volontaire. Ceci peut arriver lorsque la parole est produite par une personne se trouvant en contact avec un appareil en phase vibratoire.

Le tableau 3.1 résume les différentes variations environnementales, nous passons une revue des méthodologies en vue d'améliorer l'exactitude et la résistance de RAP à l'égard des sources de variabilité. Nous avons abordé certaines des difficultés de la reconnaissance vocale, les questions les plus problématiques étant le grand espace de recherche et la forte variabilité, cela couvre l'accent, le taux de parole, dialectes régionaux et sociaux, la physiologie des orateurs, l'âge, les émotions, etc. Aussi, les différentes causes de la variabilité acoustique et environnementale. Certains attributs de l'environnement demeurent relativement constants au cours d'une déclaration comme l'équipement d'enregistrement, la quantité de réverbération de la pièce et les caractéristiques acoustiques du locuteur à l'aide du système. D'autres facteurs, comme les niveaux de bruit et de signal, seront supposés varier lentement par rapport à la vitesse à laquelle la parole change.

Les techniques conventionnelles qui compensent les effets du bruit additif et du filtrage linéaire des sons de la parole peuvent apporter une amélioration substantielle de la précision de la reconnaissance lorsque la cause de la dégradation acoustique est quasi stationnaire. La reconnaissance de la parole dans les SNR inférieurs, et en particulier la parole en présence de sources transitoires d'interférence, y compris en particulier la parole de fond et la musique de fond restent essentiellement des problèmes non résolus à l'heure actuelle [107]. Certaines techniques sont expliquées dans le tableau suivant.

Table 3.1: Différentes variations environnementales

Raison de la variation	**Effet de la variation**	**Techniques générales pour gérer la parole**
Anatomie des voies vocales	La densité spectrale de puissance de la parole varie dans le temps en fonction du signal global et de la configuration des articulateurs de la parole.	-Rémunération et invariance (normalisation) Normalisation de la longueur des voies vocales (VTLN) -Modèles Markov cachés (MMC), comme une séquence de régimes aléatoires stationnaires
Bruit	Les informations indésirables dans le signal de la parole comme des voix en arrière-plan qui corrompt la qualité du signal de la parole et dégrade les performances du système de RAP.	-Méthode de soustraction spectrale -L'algorithme SPLICE fonctionne sur la représentation spectrale -Un filtre FIR adaptatif de haute commande. Lorsqu'aucune référence à une source de bruit externe n'est disponible -Filtrage de la saucisse pour l'entrée stationnaire et le bruit, aucune source de référence de bruit n'est requise. -L'algorithme ALGONQUIN fonctionne sur les spectres de journal.
Effet d'écho	Le signal de la parole a rebondi sur un objet environnant, et qui arrive dans le microphone quelques millisecondes plus tard. Cet effet d'écho ajoute avec le signal de la parole d'origine, et difficile d'obtenir un discours original propre.	-Algorithmes d'annulation d'écho -Algorithme des moindres carré moyen (LMS) -Approche pour l'annulation d'Echo -Statistiques minimales (MS) -Décomposition en valeur propre -Transformation Fourier (DFT) -Modèle état-espace

Effet d'écho		-Développement en série de Taylor (VTS) -Méthode ALGONQUIN -Estimation de la variation temporelle -Changement de modèle dynamique linéaire (SLDM) -Cadre d'estimation bayésien -Modèle d'état de marche aléatoire
Réverbération	Si le lieu dans lequel le signal de parole a été produit fait fortement écho, cela peut donner lieu à un phénomène appelé réverbération, qui peut durer jusqu'à quelques secondes. Les signaux vocaux d'origine sont masqués par l'écho.	-Algorithmes de filtrage additif du bruit -Schémas adaptatifs -Schémas proportionnels -Filtres adaptatifs proportionnés -Combinaison basée sur des blocs -Schémas de combinaison -Filtrage adaptatif de sous-bande -Banques de filtres DFT sur-échantillonnés uniformes -Banques de filtres DFT sur-échantillonnés de sous-bande -Considérations dans le domaine temporel -Filtres de Volterra -Algorithmes de type proportionné -Algorithmes contrôlés par parcimonie
Variabilité des canaux	Le bruit qui change avec le temps, et différents types de microphones et tout ce qui affecte le contenu de l'onde acoustique du haut-parleur à la représentation discrète dans un ordinateur.	-Soustraction moyenne cépstrale -Le filtrage RASTA des trajectoires spectrales.
Bruit de convolution	Les dégradations de la qualité du signal de la parole dues au canal proviennent de ses propriétés spectrales lentement variables (ou réponse d'impulsion).	-Caractéristiques moyennes de la parole (Soustraction de la moyenne Cepstrale). -Évaluation de la réponse d'impulsion comme données manquantes et combinée avec la réduction du bruit additif. -Filtrage à faible passage, en supprimant la moyenne cepstrale de tous les vecteurs caractéristiques de l'expression.

Effet lombard	En raison des environnements bruyants, l'acoustique est corrélée dans le signal de la parole. Mais pour quantifier cet effet, aucune spécification n'est connue.	Algorithmes de filtrage du bruit additif Appliquer les filtres passe-bas et passe-haut
-Stress physique(l'environnement de la force, la distraction auditive, environnement thermique, équipement personnel.) -Stress émotionnel(charge de tâche,fatigue mentale,les angoisses de la mission et anxiétés de fond.)	Le bruit peut être considéré comme stationnaire lors d'une commande vocale, mais d'une commande vocale à l'autre, ses caractéristiques peuvent changer.	-Application d'une méthode d'extraction de fonctionnalités appropriée comme LPCC, MFCC -Algorithmes d'estimation et d'annulation du bruit. -Annulation du bruit à effectuer par le type Wiener Filtrage
Dialectes régionaux et sociaux	Les dialectes sont des variations liées au groupe au sein d'une langue. Le dialecte régional implique des caractéristiques de prononciation, de vocabulaire et de grammaire qui diffèrent selon la zone géographique. Les dialectes sociaux se distinguent par des caractéristiques de prononciation, de vocabulaire et de grammaire selon le groupe social de l'orateur.	Considérez les dialectes comme une autre langue en RAP, en raison des grandes différences entre deux dialectes. Comme dans notre cas des systèmes de RAP basé sur le dialecte marocain.
Émotions	Les émotions dans la reconnaissance vocale sont concentrées sur la tentative de classer un signal de discours « stressé » dans sa catégorie d'émotion correcte. Variabilités intrinsèques: fort, doux, Lombard, rapide, en colère, effrayé; et le bruit.	-Amélioration du traitement front-end, fonctionnalité des méthodes d'extraction pour la reconnaissance de la parole stressée et non stressée simultanément. -Amélioration du traitement back-end ou mesures de reconnaissance robustes. -Méthodes d'entraînement améliorées : formation multi-style et génération simulée de jetons de stress.

Le tableau 3.1 présente une revue des méthodologies en vue d'améliorer l'exactitude et la résistance de RAP à l'égard des sources de variabilité.

3.7 Méthodes et Algorithmes de débruitage du signal de parole

3.7.1 Introduction

Pendant plusieurs décennies, L'amélioration de la parole (speech enhancement (SE) en anglais) vise à améliorer la qualité des signaux de la parole bruitée qui sont corrompus par le bruit additif, le bruit multiplicatif ou le bruit convolutionnel en utilisant divers algorithmes. Le terme qualité de la parole peut être interprété comme la clarté, l'intelligibilité, la douceur ou la compatibilité avec une autre méthode dans le traitement de la parole comme la reconnaissance vocale et le codage de la parole. Les principaux objectifs de l'amélioration de la parole peuvent être classés dans l'élimination du bruit de fond, l'annulation des échos, la suppression de la réverbération et le processus d'apport artificiel de certaines fréquences dans un signal [108] ,[109] [110]. Après de nombreuses années d'étude, c'est toujours un problème très difficile, parce que la plupart des journaux s'appuient sur l'estimation du bruit pendant l'activité non vocale en supposant que le bruit de fond est non corrélé, non stationnaire et variant lentement [111], [112] [113] . Par conséquent, les caractéristiques sonores estimées en l'absence de parole peuvent être utilisées ultérieurement en présence de la parole, alors qu'en temps réel, les hypothèses ne tiennent pas tout le temps [113].
La source d'interférence peut être un bruit à large bande sous la forme d'un bruit blanc ou coloré, d'un signal périodique comme dans le bruit de bourdonnement, de réverbérations de pièce, ou il peut prendre la forme du bruit de profond. Les deux premiers exemples représentent des sources de bruit additif, tandis que les deux autres exemples représentent des sources de bruit convolutionnelles et multiplicatives, respectivement. Le signal de la parole peut être attaqué simultanément par plus d'une source de bruit [114] , [115], [116], [117] .
L'amélioration de la parole peut être séparée dans le filtrage spatial et le filtrage spectral. Dans le filtrage spatial, les sons interférents sont réduits en fonction de leurs propriétés spatiales, tandis que les signaux d'une direction cible sont maintenus. Les algorithmes de filtrage spectral, également connu sous le nom d'amélioration de la parole à canal unique, traiter les signaux de la parole bruité capturés par un microphone unique ou la sortie d'un faisceau ancien ou algorithme de filtrage spatial. De nombreuses approches ont été proposées dans la littérature pour la majoration de discours de canal simple. En général, ces méthodes peuvent être classées par catégories dans deux larges classes en incluant des approches supervisées et non supervisées [118], [119], [120]. Ainsi, ces deux types d'approches seront reconsidérés ici en les divisant dans quelques sous-classes fondamentales. Dans les sections suivantes, nous donnons un aperçu de certaines approches et nos contributions.

3.7.2 Méthodes non supervisées

Plusieurs algorithmes non supervisés ont été proposés pour améliorer les signaux monocanaux, la plupart d'eux sont basées sur la transformée de Fourier à court terme (STFT). Une révision détaillée peut être trouvée dans [108], [109], [113], [114], [116], [120] . Ces méthodes peuvent être divisées en deux principales approches en incluant des approches paramétriques et non-paramétriques. Dans l'approche paramétrique, la distribution de signal est connue. Donc, peut-être jusqu'à un certain paramètre vectoriel, qui rend possible de recourir à Bayésien standard et la théorie de probabilité. Dans l'approche non-paramétrique, la distribution de signal est inconnue.

3.7.2.1 Approches non-paramétriques

Dans le cadre des approches non-paramétriques, les méthodes d'amélioration de la parole les plus simples à mettre en œuvre sont les soustractions spectrales de puissance. Les méthodes peuvent être effectuées avec peu de calculs et sans beaucoup d'information préalable [114], [121]. Ils sont basés uniquement sur le modèle de signal fondamental où le bruit est additif. Une autre technique est l'algorithme de Wiener optimal, qui suppose un rapport linéaire entre les coefficients de signal bruités et les coefficients de signal propres [115] ,[117]. Autres estimateurs non paramétriques sont basés sur la décomposition sous-espaces. L'idée principale est que l'espace bruité peut être décomposé en un espace de signal propre et un espace uniquement de bruit [109], [119]. Récemment, certains des méthodes de masquage binaires ont été proposées afin d'améliorer l'intelligibilité de la parole [122]. Dans le domaine temps-fréquence, les techniques consistent à ne conserver que quelques séries fréquentielles des spectres bruyants en forçant à remettre à zéro les autres.
Pour notre thèse nous avons utilisé l'algorithme de la soustraction spectrale [114], le filtre de Wiener optimale [122], et l'algorithme d' OMLSA (The Optimal Modified Minimum Mean Square Error Log-Spectral Amplitude) [123] et dans quelques tests nous avons utilisé la fonction réduction de bruit du logiciel Audacity.

- **Débruitage par la méthode de Soustraction Spectrale** La soustraction spectrale est la méthode de débruitage la plus ancienne, elle est introduite par S.Boll [124]. Le principe de la méthode de soustraction spectrale suppose que le bruit est un procède stationnaire dans un temps court et qu'il est non-corrélé au signal de parole. On suppose que l'oreille humaine est insensible à la phase du signal et on considère que la phase du signal de parole est la même que celle du signal bruite comme montrer par l'équation (3.5).

$$y[n] = s[n] + d[n] \tag{3.5}$$

  Où n représente l'index temporel discret et $y[n]$, $s[n]$ et $d[n]$ sont respectivement le signal de la parole bruitée, le signal de la parole propre et le bruit additif. Cette technique utilise la transformation de Fourier DFT pour travailler dans le domaine fréquentiel et a pour principe de soustraire une estimée du bruit à partir du signal observé. Le bruit est supposé additif, stationnaire ou légèrement variant ce qui nous permet de l'estimer pendant les périodes de silence, l'estimation du bruit se fait sur plusieurs trames d'acquisition ($\simeq$ 300ms). Une régie de soustraction spectrale est réalisée afin de séparer le bruit du signal de parole.

Dans le domaine fréquentiel, 1'équation (3.5) implique :

$$Y[f] = S[f] + D[f] \tag{3.6}$$

Où $S[f]$et $D[f]$sont respectivement les transformées de Fourier du signal et du bruit. Il est nécessaire d'effectuer un estimé de l'amplitude fréquentielle du bruit. Une fois cet estimé connu, il ne reste qu'à soustraire la valeur obtenue du signal :

$$\hat{S}[f] = Y[f] - \hat{D}[f] \tag{3.7}$$

Pour la reconstruction du signal débruité dans le domaine temporel, il est nécessaire d'avoir la phase du signal original. Cette dernière est obtenue par extraction de la phase du signal bruité lui-même qui est utilisée avec la transformée de Fourier inverse sur l'estimé de l'amplitude fréquentielle pour récupérer le signal débruité.

- **Débruitage par filtre optimal (filtrage de Wiener)** Le filtre de Wiener est parmi les méthodes de débruitage classiques les plus utilisées dans la littérature [125]. Il a été introduit dans les années 1950 par Norbert Wiener pour trouver une estimation optimale d'un signal $s[n]$ au sens du minimum de l'erreur quadratique moyenne (MMSE) à partir d'un signal bruité$y[n]$. Le filtre peut s'écrire par la relation suivante [125] :

$$F_W(f) = \frac{P_s[f]}{P_s[f] + P_n[f]} \tag{3.8}$$

Où $P_s[f]$ et $P_n[f]$sont les densités spectrales de puissance, ou DSP, du signal de parole $s[n]$ et du bruit $d[n]$, respectivement. Cette formule peut être dérivée en considérant le signal s et le bruit d comme des signaux non corrélés et stationnaires. Il est possible d'exprimer l'équation 3.8 en fonction du rapport signal sur bruit (SNR), Donc, Le SNR est défini par :

$$SNR = \frac{P_s[f]}{\hat{P}_n[f]} \tag{3.9}$$

Cette définition peut être intégrée dans la relation (3.8), alors le filtre de Wiener peut s'exprimer sous cette forme comme suit [125]:

$$F_W(f) = \left[1 + \frac{1}{SNR}\right]^{-1} \tag{3.10}$$

L'inconvénient du filtre de Wiener est de fixer la fréquence à toutes les fréquences, ainsi la nécessité d'estimer la DSP du signal propre et du bruit avant le filtrage. En ce qui concerne $P_n[f]$, elle est estimée à l'aide d'une méthode de détecteur d'activité vocale (VAD), qui permet d'estimer le bruit pendant les temps d'inactivité vocale [125].

- **Débruitage par logiciel Audacity.**

Audacity est l'un des logiciels les plus utilisés dans le domaine de traitement et d'édition audio, car il est rapide, facile et gratuit. Il est capable de fonctionner avec les différents formats et encodages de fichiers audio[1] [126], il permet d'enregistrer, de jouer, d'importer et d'exporter des données en

[1] *https : //manual.audacityteam.org*

plusieurs formats. Cela peut être utilisé pour ajouter des effets spéciaux de différents types tels que l'amplification des basses, l'élimination du bruit, etc. Audacity intègre également un éditeur d'enveloppes de volume et permet l'analyse du son grâce à l'affichage paramétrable de spectrogrammes. Voici la procédure par étapes utilisée pour la suppression au d'ajouter du bruit du signal vocal à l'aide du logiciel Audacity.

étape 1 : Nous ouvrons le logiciel Audacity. Un écran apparaîtra sur le système.

étape-2 : Nous cliquons sur l'option « Fichier » et nous sélectionnons « Ouvrir » afin de sélectionner le signal vocal souhaité.

étape 3 : Nous sélectionnons le fichier « signal bruité » souhaité et on clique sur le bouton « OK ». Le fichier « son bruité » s'ouvrira sur l'écran.

étape 4 : Sélectionner l'option « Réduction du bruit » dans l'option « Effets ». Cliquer sur Effet puis élimination du bruit.

étape 5 : Sélectionner l'option « Prendre profil du bruit ».

étape 6 : Sélectionner l'option « Réduire le bruit » et cliquer sur « ok »

étape 7 : Le signal « réduite » apparaîtra sur l'écran.

étape 8 : Sélectionner l'option « Fichier » et cliquer sur l'option « Exporter l'audio » pour enregistrer le fichier sans bruit.

étape 9 : Nommer et sauvegarder sur le fichier « signal débruité ».

3.7.2.2 Approches paramétriques

En prenant en compte la distribution de la parole propre et du bruit, cette approche estime la parole propre en formulant le débruitage comme un problème d'estimation en utilisant soit le maximum de vraisemblance (ML), l'estimateur d'erreur quadratique moyenne minimale (MMSE) ou l'estimateur du maximum a posteriori (MAP). Afin de dériver des estimateurs MAP et MMSE, la fonction de densité de probabilité (PDF) de la parole peut être supposée gaussienne, super-gaussienne , laplacienne ou gamma généralisée [127]. Dans la plupart des techniques paramétriques mentionnées ci-dessus, le bruit est supposé être gaussien. En fait, le bruit est également supposé avoir une distribution laplacienne. Certaines techniques intègrent également la connaissance de la présence ou de l'absence de la parole pour améliorer encore la qualité de la parole [127].

3.7.3 Méthodes supervisées

Pour l'approche supervisée, les paramètres du modèle de parole et de bruit sont estimés en apprenant à partir des échantillons d'apprentissage correspondants. Sur la base de ces paramètres de modèle, une stratégie est proposée pour combiner le signal d'intérêt et les modèles de bruit. Ensuite, les problèmes de débruitage sont traités avec le signal bruité. Cette approche est très large, elle peut être divisée en trois classes principales: Approche de Wiener basée sur un dictionnaire (codebook), approche basée sur le modèle de Markov caché (HMM) et approche basée sur les réseaux de neurones profonds (DNN) [127].

3.7.3.1 Approche basée sur HMM

Ici, au lieu de la synthèse de prédiction linéaire, le signal de la parole propre et le signal bruité ou les autres paramètres sont modélisés par HMM. Dans certains travaux [128] [129], les paramètres de la parole et du bruit sont supposés être gaussiens. Plus récemment, autres chercheurs [130] travaillent directement avec les coefficients dans le domaine transformé où ces coefficients sont supposés avoir une distribution gaussienne ou super gaussienne.

3.7.3.2 Approche basée sur DNN

Au cours des dernières années, de nouvelles approches dans le domaine de la séparation de sources et du rehaussement de la parole sont apparues, basées sur les algorithmes d'apprentissage profond. Les réseaux de neurones ont permis des progrès considérables par rapport aux méthodes classiques de débruitage de la parole, comme les approches basées sur la soustraction spectrale ou le traitement d'antennes. En particulier, la réduction de bruit et la séparation de sources de paroles ont été beaucoup améliorées [131] et les réseaux de neurones ont amené des avancées en réduction d'écho [103], [132] et en déréverbération .Toutefois, ils n'ont jamais été utilisés pour résoudre la réduction conjointe de bruit, d'écho acoustique et de réverbération de bout-en-bout, c'est-à-dire, pour l'optimisation conjointe des modules de rehaussement, et non pas dans le sens usuel [131]. Les méthodes de débruitage de la parole utilisant différentes architectures de réseaux de neurones profonds, comme les auto-encodeurs (DAE) qu'il s'agit d'un réseau de neurone qui tente de mapper les entrées bruitées à leurs versions propres. Il s'est avéré plus robuste et fournit de meilleurs résultats en termes de évaluations subjectives et objectives. Les réseaux de neurones récurrents (RNN) sont également utilisés pour la tâche de débruitage. L'architecture basée sur les réseaux neuronaux récurrents appelée RDAE (Recurrent Denoising Auto-Encoders) et qui a montré des performances significatives en exploitant les informations de contexte temporel dans les signaux embarqués.
D'autres méthodes s'appuient sur les réseaux ayant sur des couches totalement connectées LSTM (long short-term memory) qui sont les meilleurs réseaux de neurones profonds appliqués pour la tâche de débruitage et LSTM bidirectionnelles (biLSTM) [116]. Dans d'autres cas le réseau peut être entraîné à reconstruire le signal source lui-même, parfois de façon générative, ou bien un masque qui permet d'estimer la proportion de signal cible présente dans le mélange en chaque point temps-fréquence [133] [134].

3.7.3.3 Approche de Wiener basée sur codebooks:

Cette approche est basée sur le filtre de Wiener, elle utilise des codebooks de paramètres autorégressifs (AR) pour la synthèse linéaire des prédictions des signaux de la parole et du bruit. En fait, le filtre de Wiener est le rapport du signal propre et du spectre d'alimentation bruité. En outre, le spectre d'alimentation bruité est raisonnablement supposé être la somme des spectres de signal propre et de puissance sonore. Ces spectres peuvent être déterminés à partir des paramètres AR. Par conséquent, cette approche construit d'abord des livres de code (codebooks) pour les spectres de la parole et du bruit via l'apprentissage de la base de données mixte.

3.8 Conclusion

Dans ce chapitre, nous avons présenté une revue des notions fondamentales du signal de parole, ses applications ainsi que les différents types de bruits peuvent affecter ce signal. Nous avons commencé par présenter les caractéristiques des bruits. Nous avons aussi présenté quelques méthodes de base d'analyse et de modélisation du signal de parole. Ces méthodes peuvent être utilisées pour mettre en évidence les caractéristiques fréquentielles de ce signal et pour mettre en oeuvre, en exploitant ces caractéristiques.
En outre, nous avons présenté l'ensemble des techniques de réduction de bruit les plus répandues dans la littérature, des approches basées sur l'apprentissage automatique (supervisé) et des techniques non supervisées. De plus, nous avant expliquée le principe des algorithmes de débruitage, chaque algorithme a un principe différent pour obtenir en sortie un signal contenant moins de bruit.

Chapter 4

Implémentation et Résultats

4.1 Introduction

Plusieurs facteurs influencent sur la performance des systèmes de RAP en termes de conditions environnementales. Les effets des bruits additifs sont l'un des problèmes majeurs dans un système de RAP. Par conséquent, nous s'intéressons dans ce chapitre à la mesure du degré de cet effet sur les performances des systèmes de RAP, en plus de contribuer à des solutions pour améliorer ces performances en se concentrant plus sur la méthode d'injection de bruit, c.-à-d. l'apprentissage des données mélangées avec plusieurs bruits à plusieurs rapports signal sur bruit.
Dans ce chapitre nous avons présenté tous nous expériences que nous avons réalisées, nous présentons aussi les outils, les bases de données que nous avons utilisé. Au cours de cette thèse, nous avons mené un certain nombre d'expériences ciblant spécifiquement l'effet des données bruitées sur les performances des systèmes de reconnaissance de la parole Arabe. Les premières étaient relatives à la construction des bases de données que nous avons utilisé dans nos expériences, nous combinons des corpus de parole propre et de bruit pour construire des corpus de parole bruitée, nous les avons exploité dans la section 4.3. Ensuite, les secondes étaient relatives au développement d'un système de RAP avec le modèle hybride de mélange de gaussien et la quantification vectorielle (GMM-VQ), nous avons présenté une partie des résultats dans la section 4.8, ainsi le choix de Matlab comme librairie de développement. Nous avons procédé à la construction de quatre (4) systèmes de RAP de sorte que:
Nous avons construit le premier système de reconnaissance automatique de la parole hybride GMM-HMM basé sur la boite à outils CMUSphinx, le deuxième système se construit via le modèle hybride HMM-DNN basant sur la boite à outils KALDI. Le troisième système se construit sous la base de la boite à outils HTK, il est fondé sur la modélisation statistique (HMM) des modèles acoustiques. Enfin, le quatrième système basé sur des modèles neuronaux profonds sera tester avec l'envirnmemnmet Matlab fondée sur un système se basant sur des réseaux de neurones convolutifs (CNN). Tous ces systèmes seront évalués relativement avec nos propres corpus bruités et avec des bases de données mélangé avec des bruits. Afin de valider l'efficacité et de choisir la méthode appropriée à notre approche, nous proposons une étude comparative de ces quatre systèmes.

4.2 Les bases de données utilisées

Nous avons d'abord utilisé des bases de données existant dans littérature. Nous l'avons modifié en ajoutant plusieurs bruits à différents pour évaluer notre approche d'apprentissage multi-styles. Ensuite, nous avons construit nos propres corpus plus adaptés à notre problématique. Dans cette section, nous donnons toutes les informations liées à cette étape.

4.2.1 La base de données vocales SDDN

La base de données SDDN[135] est une base des chiffres prononcé en anglais dans des conditions bruitées, elle est extraite en modifiant l'ensemble de données de commandes vocales v0.02 [136], c'est une base de donnée gratuite disponible sur le site GitHub[1], l'ensemble de données original se compose de plus de 105 000 fichiers audio .wav de personnes prononçant trente mots différents. Ces données ont été collectées par Google et publiées sous une licence CC BY. Nous avons pris seulement les 1500 premiers locuteurs pour chaque prononciation des 10 chiffres.

Le corpus SDDN[135] se compose d'environ 150 heures de fichiers audio .wav de personnes prononçant dix mots différents mélangées à des bruits déférents. Afin de préparer un grand corpus conçu pour les chercheurs pour l'utiliser dans des algorithmes d'enchaînement de la parole et de détection de bruit. Nous ajoutons cinq bruits de fond à tous ces fichiers comme les bruits de rue, de zone piétonne, de café, de babillage et le bruit rose que nous avons choisi parmi tous les bases de de données bruité populaires tels que AURORA-2 et CHiME3 car ils ont une puissance élevée. Nous avons choisi ces cinq types de bruit pour avoir différentes conditions de bruit de fond. Ces types de bruit ont été mélangés avec les chiffres prononcé en anglais que nous avons choisi parmi l'ensemble de données de commandes vocales v0.02 [136] avec une grande variété du rapport signal-bruit (SNR) à 10 dB, 5 dB, 0 dB, -5 dB et - 10 dB respectivement. Le SDDN se compose de dix fichiers de zéro à neuf, chaque fichier contient cinq sous-fichiers de bruit (Murmure : Babble, café, zone piétonne, rue, rose) chaque sous-fichier contient cinq sous-fichiers constitués de niveaux SNR (-10 dB, -5 dB, 0 dB, 5 dB, 10 dB). L'ensemble de données final SDDN se compose de 375000 audio de 10 mots répartis dans les fichiers de bruits et les niveaux SNR, indiqués dans le tableau 4.1 et la figure 4.1, (10 chiffres, 1500 enregistrements pour chaque chiffre, 5 type de bruit sur 5 niveaux de SNR : $10 \times 5 \times 5 \times 1500 = 375000$). La taille de l'ensemble de données est plus de 12 Go.
Dans la figure 4.1, les fichiers sont organisés en dossiers, chaque nom de répertoire marquant le chiffre prononcé dans tous le contenus des fichiers audio. Des identifiants aléatoires ont été attribués à chaque locuteur; aucun détail n'a été conservé sur l'âge, le sexe ou le lieu des participants.

[1] *https : //github.com/tensorflow/docs/blob/master/site/en/r1/tutorials/sequences /audio_recognition.md*

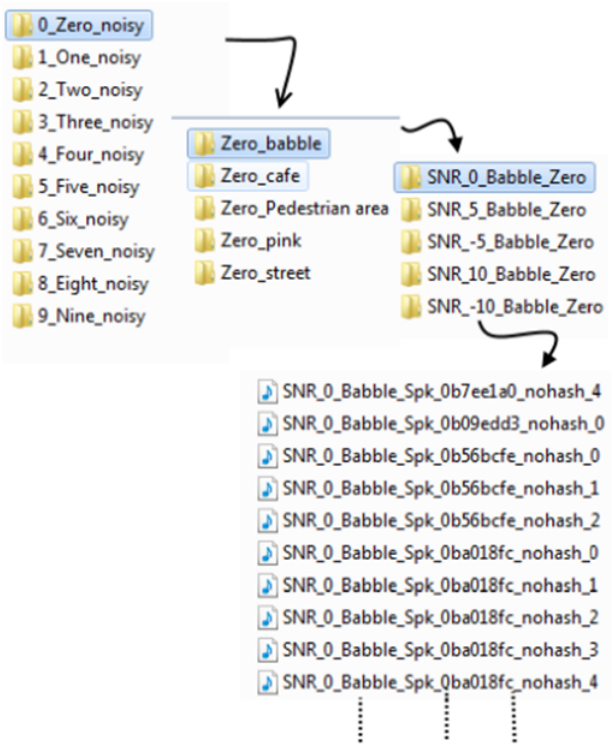

Figure 4.1: Structure générale des fichiers et sous-fichiers pour la base de données SDDN.

Table 4.1: Nombre d'énoncés de mot dans l'ensemble SDDN

Mots isolés	Nombre d'occurrences
Zero	1500
One	1500
Two	1500
Tree	1500
Four	1500
Five	1500
Six	1500
Seven	1500
Eight	1500
Nine	1500

Prétraitement des données : Nous utilisons l'environnement MATLAB R2018a pour ajouter le type de bruit préféré à tous les mots sélectionnés parmi l'ensemble de données SDDN. Nous avons utilisé l'algorithme «SpeechAddNoise.m» pour mélanger les signaux de chiffres prononcés dans des fichiers .wav avec le type du signal bruité et le rapport signal sur bruit souhaité, nous avons fixé la fréquence à 16000 HZ, les enregistrements sont coupés de manière à avoir un silence presque minimal au début et à la fin, la fonction «resample.m» a été utilisée pour rééchantillonner des données uniformes ou non uniformes pour un taux fixe, donc le signal de bruit prend la durée de la parole. La normalisation du vecteur est une dernière étape du traitement des sons, parfois, nous normalisons d'abord pour éviter l'écrêtage en utilisant la formule ci-dessous:

$$parole \;=\; parole \;/\; max\;(abs\;(parole)) \tag{4.1}$$

Après avoir normalisé la parole et déterminé le niveau du rapport SNR, nous obtenons le résultat

final de la parole mixte comme suite:

$$\text{Speech\_mix} = \text{parole} + \text{bruit} \tag{4.2}$$

Alors, nous résumons les étapes suivies pour mixer la parole avec le bruit comme suite:
étape 1 : Sélection de 1500 énoncés à partir de tous les chiffres prononcés à partir de l'ensemble de données des commandes vocales contenant différents locuteurs.
étape 2 : Appliquer les algorithmes de réduction du bruit de base spectrale dans MATLAB pour réduire le bruit de fond de toutes les pièces.
étape 3 : Choisir le type de bruit et le niveau de SNR que nous désirons.
étape 4: Sélectionner toutes les données audio puis mixer-les avec le type de bruit choisi. Tous les fichiers audio durent une seconde (1 s) et de format wav (16 bits, mono, 16 000 Hz).
étape 5: Enregistré les nouvelles données sous la format *.wav dans le répertoire approprié. Les fichiers sont nommés comme suit:
Nom général:
SNR_{10 ou 5 ...)_{Pink,Babble,...}_Spk_ {nom du locuteur à partir de la base des commandes vocales}. Le nom peut être divisé en trois segments, qui sont séparés par le signe de soulignement {_}. **Exemple :** SNR_10_Pink_Spk_0a196374_nohash_0.wav

Table 4.2: Informations sur l'ensemble des données SDDN

Processus	Description
Nombre de mots propres sélectionnés	$1500 \times 10 = 15000$
Nombre total de mot bruitées	$1500 \times 10 \times 5 \times 5 = 375000$
Nombre de types de bruits utilisé	5
Nombre de niveaux de SNR utilisés	5
Taille totale de la base de données 12 Go	12 GB
Fréquence d'échantillonnage, Fs	16000 HZ
Logiciel utilisé pour mixage	MATLAB R2018a v Essai

Nous présentons certains tests et les implémentations que nous avons fait par la base SDDN [135] dans la section 4.2.

4.2.2 Le corpus ARBDIGITS

Les ressources vocales dédiées à la langue arabe sont beaucoup moins nombreuses que celles disponibles pour les autres langues. Donc, Nous avons construit une petite base de données sonores, ARBDIGITS de 100 locuteurs arabes marocains, hommes et femmes. Ils prononcent les chiffres arabe de «Siffer» (zéro) à «Tisaa» (neuf). Chaque locuteur prononce chaque chiffre plusieurs fois de manière isolée. Les données vocales ont été enregistré à des fréquences d'échantillonnage de 8000 Hz.Notre corpus ARBDIGITS a été constitué en quatre étapes, qui représentent autant de moments de choix méthodologiques :

1. Le travail préparatoire à la phase d'enregistrement ;
2. L'enregistrement des données ;
3. Le traitement des données par l'outil « Audacity » ;
4. La mise en forme des données avec MATLAB dans la phase de test.

Nous utilisons l'outil de suppression de bruit disponible dans le logiciel "Audacity" pour supprimer le bruit de fond des enregistrement d'origine. Par conséquent nous obtenons les données propres que nous utilisons dans la phase d'apprentissage du système dans le travail [36]. Nous utilisons dans la phase de test le ARBDIGITS bruité, on ajoute le bruit gaussien blanc (AWGN) avec MATLAB à la base de données propre ARBDIGITS à différents niveaux de rapport signal sur bruit (SNR) variant de 5 dB à 20 dB. Les données de test se composent de 15 locuteurs (10 des hommes et 5 femmes) âgées entre 15 et 40 ans. Le corpus comprend trois répétitions par chaque locuteur du même chiffre. Pendant l'enregistrement, chaque répétition a été rejouée pour s'assurer que le chiffre entier a été inclus dans le signal enregistré. Le tableau 4.3 présente plus de détails techniques et paramétriques du corpus ARADIGITS utilisée dans l'évaluation expérimentale.

Table 4.3: Paramètres d'enregistrement utilisés pour la préparation du corpus ARBDIGITS

Processus	Description
Participant	100 Locuteurs (70 Hommes 30 femmes)
Environnement	Réverbération et deux canaux - mode stéréo.
Mots	10 premiers chiffres de l'arabe classique
Sous-corpus d'apprentissage	85 Locuteurs
Sous-corpus de Test	15 Locuteurs
Nombre de mots propres	$10 \times 3 \times 100 = 3000$
Nombre de mots bruités	$3000 \times 1 \times 4 = 12000$
Type du bruit	AWGN
Niveaux de SNR	5 dB, 10 dB, 15 dB, 20 dB
Taille du corpus	1 GB
Fréquence , fs	8000Hz
Logiciel	MATLAB R2018a v d'essai

Dans le but d'évaluer les performances de notre système de reconnaissance automatique de la parole Arabe avec la base de données acoustiques ARBDIGITS[36], nous l'avons testé pour diffèrent rapport de SNR et nous exposons certains résultats expérimentaux dans la section 4.4.

4.2.3 la base de données vocales NASCIW

Le corpus NASCIW (The new Noisy Arabic Speech Corpus for Isolated Words) c'est une version bruitée de la base de donnée arabe gratuit ASCIW[2] (Arabic Speech Corpus for Isolated Words) [137]. C'est un corpus en arabe pour les mots isolés qui a été développé par l'auteur du département des systèmes d'information de gestion de l'Université King Faisal. Il contient environ 10 000 énoncés de 20 mots prononcés mots environ 10 fois par 50 locuteurs natifs arabes (47 hommes et 3 femmes). Il comprend 20 mots arabes: 10 chiffres arabes (de 0 à 9) et 10 autres mots ("نعم", "لا", "التحويل", "انهاء", "التنشيط", "الحساب", "البيانات", "التمويل", "الحساب", "التسديد"). Il a été enregistré avec une fréquence d'échantillonnage de 44100 Hz et numérisé sur 16 bits, ainsi qu'un mode stéréo à deux canaux [137]. Ce système de codage permet aux chercheurs d'utiliser cet ensemble de données non seulement pour les systèmes de RAP, mais également pour différentes tâches de classification, par exemple, les systèmes d'identification des locuteurs, de biométrie vocale, etc. Dans d'autres formats pour permettre aux chercheurs d'appliquer différentes méthodes d'extraction de caractéristiques.
Dans le tableau 4.4, nous avons représenté tous les mots sélectionnés, qui ont été inclus dans le corpus

[2] *http : //www.cs.stir.ac.uk/ ∼ lss/arabic/*

de la parole arabe pour les mots isolés ASCIW, leurs prononciations, son approximation en anglais et leur traduction avec l'alphabet phonétique international (IPA).

Table 4.4: Tous les mots qui ont été inclus dans le corpus ASCIW avec le nombre des occurrences pour chaque mot et son approximation et traduction en anglais.

Arabe	Traduction	Anglais Approximation	IPA	Nombre des occurrences
صفر	Zero	Safer	/ sˤfr /	93 occurrences
وَاحد	One	Wahed	/ wa:hid /	100 occurrences
اثْنان	Two	Ethnan	/ ʔθna:n /	100 occurrences
ثلاثة	Three	Thaltha	/ θala:θh /	100 occurrences
أربَعة	Four	Arbah	/ ʔrbaʕh /	100 occurrences
خمسَة	Five	Khamsah	/ xamsat /	100 occurrences
سِتّة	Six	Setah	/ sitat /	100 occurrences
سبعة	Seven	sabah	/ sabʕah /	100 occurrences
ثمانية	Eight	Thamanah	/ θma:njh /	100 occurrences
تِسعة	Nine	Tesah	/ tisʕah /	100 occurrences
التنشيط	Activation	Al-tansheet	/ a:tanʃytˤ /	100 occurrences
التحويل	Transfer	Al-tahweel	/ a:tahwyl /	99 occurrences
الرصيد	Balance	Al-raseed	/ a:rasˤyd /	100 occurrences
التسديد	Payment	Al-tasdeed	/ a:tasdyd /	100 occurrences
نعَم	Yes	Naam	/ nʕm /	100 occurrences
لا	No	Laa	/ la: /	100 occurrences
التّمويل	Funding	Al-tamueel	/a:tamwyl/	100 occurrences
البيَانات	Data	Al-baynat	/a:lbayana:t/	100 occurrences
الحسَاب	Account	Al-hesab	/ a:lhisa:b /	100 occurrences
انهَاء	End	Enha	/ ʔinha:ʔ /	100 occurrences

Pour créer une version bruitée de la base de données ASCIW, nous avons divisé le corpus en deux sous-ensembles, l'un d'apprentissage et d'estimation des paramètres et l'autre de test. La base d'apprentissage contient 80% de la taille total du corpus (8000 échantillons) et la base de test contient les 20% restants (2000 échantillons). Par conséquent, le corpus de test ne se recoupaient pas avec celles d'apprentissage. Les fichiers sont organisés en dossiers, chaque nom de répertoire marquant le mot prononcé dans tous les fichiers audio contenus. Des identifiants aléatoires ont été attribués à chaque locuteur; aucun détail n'a été conservé sur l'âge, le sexe ou l'emplacement du participant. Nous avons organisé et modifié la base de données ASCIW en la renommant NASCIW, c'est une version modifiée du "Corpus arabe des mots isolés" avec l'ajout de bruit de fond, nous proposons de l'utiliser dans nos études, ce nouveau corpus (NASCIW) se compose de 50 fichiers de S01 à S50, chaque fichier contient 3 sous-fichiers de bruit (Babble, street, pink) chaque sous-fichier contient cinq sous-fichiers qui consistent en des valeurs SNR (10 dB, 15 dB , 20 dB et propre pour la phase d'entraînement, -5 dB, 0 dB et 5 dB pour la phase de test). Les bruits ont été ajoutés à tous les occurrences en commençant à des points de départ aléatoires dans les enregistrements de bruit, ce qui a entraîné une plus grande variété.

La version finale du corpus NASCIW se composait de 122000 occurrences de 20 mots répartis dans les sous dossiers des bruits et les niveaux de SNR (96000 occurrences les données d'apprentissage et 26000 les données de test). Nous disposons suivant le mixage suivant le corpus de test final composé de 5h pour 10 conditions différentes : 10 locuteurs × 20 mots × 10 répétition × 3 (types de bruit) × 3 (niveaux de SNR) + (la condition de parole propre). Les autres informations techniques sont indiquées dans le tableau 4.5, (50 locuteurs arabes, 20 mots, 10 répétitions, 3 nombres de bruit dans 3 niveaux de SNR. La taille de l'ensemble de données s'élève à environ 25 heures d'audio qui soit

près de 9 Go [96].

Table 4.5: Informations et état du corpus NASCIW utilisé [96].

Processus	Description
Participant	50 locuteurs (47 hommes 3, femmes)
Environnement	Mode réverbérant et stéréo à deux canaux.
Mots	20 mots prononcés en arabe
Ensemble d'entraînement	40 locuteurs (96000 mots)
Ensemble de test	10 locuteurs (26000 mots)
Nombre de mots propres sélectionnés	20×10×50=10000
Nombre total de mots bruités	10000×3×3=90000
Type de bruit utilisé	Street, Babble, Pink
Niveau SNR utilisé	Propre,5 dB, 0 dB, -5 dB
Taille totale de la base de données	9 GB
Fréquence d'échantillonnage, fs	44100 Hz et 16 kHz avec 16 bits
Logiciel utilisé pour le mixage	MATLAB R2018a version Trial

4.2.4 Le corpus DARIJA_MO

Dans le travail [96] nous préparons notre propre corpus de petite taille des dix salutations les plus célèbres au dialecte marocain dans les conversations téléphoniques nommé DARIJA_MO. Ce corpus de parole enregistré par 60 locuteurs (30 hommes et 30 femmes) la majorité sont des étudiants de l'Université Sultan Moulay Slimane viennent des régions de la ville de Beni Mellal, prennent 1800 expressions dans lesquels chaque locuteur prononce chaque expressions trois fois, cette base de données multi-locuteurs a été enregistrée dans les environnements réels de la vie courante (bruitées : Autobus, Café, salle de photocopie, Cour de faculté).
Notre base de données est riche de nombreux aspects, on résume les dimensions et les aspects de sa richesse comme suite:

-(a) Environnement
Calme (Cour de faculté), moyenne calme (salle de photocopie), moyenne bruité (Café), très bruité (autobus).

-(b) Microphones
Nous avons plusieurs microphones des déférents smart phone par rapport à un microphone dans certaines bases de données.
Microphone de qualité moyenne, de haute qualité et de très haute qualité.

-(c) Différentes combinaisons de microphones et d'appareils d'enregistrement
Par exemple enregistrement de faible qualité avec microphone de qualité moyenne.

-(d) Sessions
3 répétitions

-(e) Multi-accents : Ethnicité
Locuteurs d'accent dialecte marocaine.
Locuteurs d'accent non arabes (amazigh).

Les signaux du corpus sont pour la plupart dégradés par la présence d'un bruit additif en raison du bruit de fond ajouté dans les différentes conditions d'enregistrement. Pour les données propres on

utilise le logiciel Audacity pour le débruitage des fichiers audio dans le but d'améliorer le rapport signal sur bruit et de réduire l'effet du bruit de fond.
Le tableau 4.6 représente les dix expressions de salutation en Dialecte marocaine utilisé dans des conversation téléphoniques et quotidienne.

Table 4.6: Les expressions construis DARIJA_MO corpus

N°	Expression en Dialecte Marocain	Expression en Latin scriptes
1	السلام عليكم	Salam Alaykom
2	صباح الخير	Sbah lkhir
3	كيداير	kidayir
4	لباس عليك	Labas Aelik
5	الحمد لله	lhamdolilah
6	مالين الدار بخير	Malin dar bikhir
7	كلشي بخير	Kolchi bikhir
8	أش تتعاود	Ach tataawed
9	الوليدات	Lwlidat
10	بالسلامة	beslama

Nous avons utilisé différents appareils smartphone pour enregistrer les fichiers audio (audio mono à un taux d'échantillonnage 44 kHz, 16-bit). Nous avons donc obtenu un corpus de 1800 signaux classés par 2/3 données d'apprentissage et 1/3 pour de test. Dans le tableau 4.7 nous donnons certains détails techniques et paramètres de DARIJA_MO corpus.

Table 4.7: Paramètres d'enregistrement utilisés pour la préparation du corpus DARIJA_MO

Processus	Description
Nombres des expressions	10
Nombres des expressions propres	$60 \times 10 \times 3 = 1800$
Nombres des expressions bruitées	$20 \times 10 \times 4 = 800$
Base de Traitement	40 locuteurs (20 hommes,20 femmes)
Base de Test	20 locuteurs (10 hommes,10 femmes)
Base de Test bruité	20 locuteurs et 4 bruites $20 \times 10 \times 4 = 800$
Types de bruit courent	4 (autobus, café, salle de photocopie, cour de faculté)
Taille de base de données	1 GB
Fréquence d'échantillonnage, Fs	44000 HZ
Logiciels du mixage et du débruitage	MATLAB R2018a v Essai et Audacity
Appareils de capture	Smartphones Android

La structure des répertoires et des fichiers SphinxTrain pour DARIJA_MO corpus est décomposé de la façon indiquée sur la figure 3.

Le répertoire principal du notre projet basé sur PocketSphinx nommé "DARIJA_MO" a deux dossiers supérieurs, à savoir le etc et le wav. Le répertoire « wav » contient aussi des sous-dossiers, chacun d'eux contient tous les fichiers sonores de chaque locuteur (du locuteur sp1m jusqu'à locuteur sp60w). Le répertoire « etc » contient des fichiers des paramètres de configuration nécessaires pour former le modèle acoustique:

- DARIJA_MO.dic : Dictionnaire phonétique de l'arabe.
- DARIJA_MO.phone : liste de phonèmes
- DARIJA_MO.lm.DMP: modèle de langage

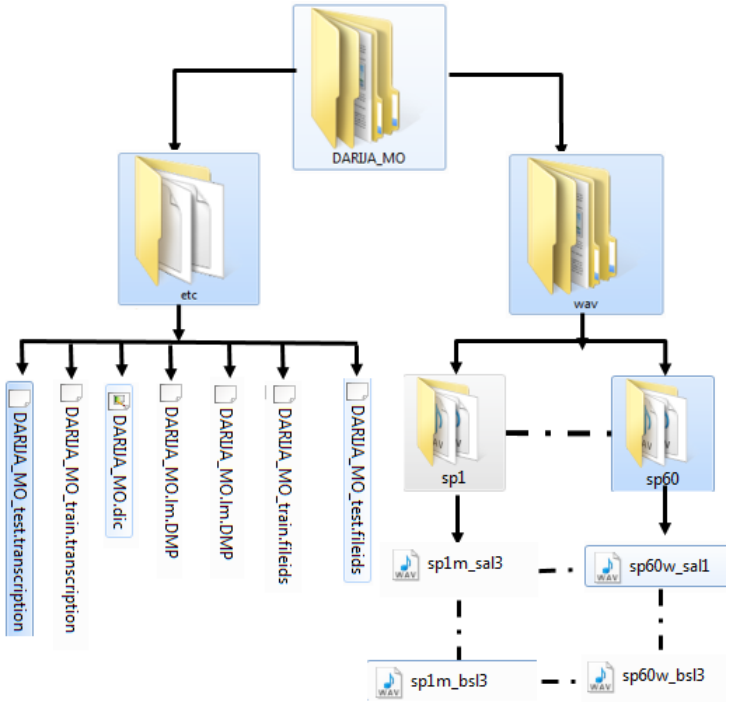

Figure 4.2: La structure des fichiers SphinxTrain pour la base de données DARIJA_MO.

-DARIJA_MO_train.fileids : Liste des fichiers d'apprentissage.
-DARIJA_MO_train.transcription : le fichier de transcription d'apprentissage.
-DARIJA_MO_test.fileids : Liste des fichiers à tester
-DARIJA_MO_test.transcription : le fichier de transcription pour le test.
Le fichier DARIJA_MO.filler doit avoir les symboles pour le silence. Le contenu du fichier etc/ DARIJA_MO.filler:
<s> SIL
<sil> SIL
</s> SIL

4.3 Évaluation des performances

Les performances des systèmes de RAP peuvent être évaluées selon différentes mesures, telles que l'erreur, la précision et la correction. La mesure la plus courante est appelée taux d'erreur sur les mots WER (Word Error Rate). Le WER est basé sur une mesure résultant de la programmation dynamique [138].
En général, les systèmes de RAP ont trois types d'erreur communs: substitution, insertion et suppression. La substitution se produit lorsque Le système reconnaît un mot différent du mot prononcé. L'insertion se produit lorsque les hypothèses de sortie contiennent un mot qui n'est pas prononcé par le locuteur. La suppression survient lorsque les résultats reconnus ont manqué des mots prononcés [139]. Le WER est obtenu par la division du nombre d'erreurs (D +I +S) par le nombre de mots dans la référence N, comme le décrit l'équation (4.3) :

$$WER(\%) = \frac{D + I + S}{N} \times 100\% \tag{4.3}$$

Où S correspond aux substitutions (mots incorrectement reconnus), D aux suppressions (mots omis), I aux insertions (mots ajoutés) et N est le nombre de mots dans da référence.
Une autre mesure d'évaluation que nous avons adopté aussi dans notre thèse est le taux de reconnaissance des mots **WRR** (Word recognition rate). Quelque expérience a été effectuée selon WRR (il s'appelle aussi la précision (Accuracy)) est symbolisé par **WAcc** ou **WRR** défini par l'équation 4.4 suivant :

$$WRR(\%) = \frac{H - I}{N} \times 100\% \tag{4.4}$$

Où N c'est le nombre total de mots de référence, I est le nombre d'insertions (mots ajoutés) et H est le nombre de mots correctement reconnus. Dans nos expériences, on considère I=0 puisque on utilise un dictionnaire des mots presque isolés.
La relation entre les deux métriques WRR et WER peut être représenté comme suit:

$$WRR = 1 - WER \tag{4.5}$$

4.4 Expérimentations

Dans les sections et les sous-sections suivantes, nous décrivons nos systèmes de RAP construit en basant sur les quatre corpus précédents, nous décrivons aussi les résultats des expériences menées pour évaluer la robustesse des méthodes et des outils proposés basé sur notre approche d'apprentissage par données bruitées. Enfin, nous présentons les évaluations relatives à tous les propositions.

4.4.1 Système basé sur le Modèle VQ-GMM et le corpus ARBDIGITS

4.4.1.1 L'architecture du système « SRAP1 »

Afin de contribuer au développement des systèmes de reconnaissance de la parole arabe que nous le l'appelons « SRAP1 », nous avons construit deux systèmes dans un comme la montre la figure 4.3. Le premier pour l'identification du locuteur et le second pour la reconnaissance des chiffres prononcé en arabe sous l'effet du bruit de fond AWGN (bruit additif, blanc et gaussien). C'est pourquoi nous proposons le modèle hybride GMM-VQ qui est basée sur les vecteurs caractéristiques MFCCs comme technique d'extraction de caractéristiques. Nous avons utilisé la méthode de Quantification Vectorielle (VQ) pour former les modèles et pour la phase d'identification du locuteur. Par suite, on utilise le modèle du GMM pour la modélisation des locuteurs et dans la phase de la reconnaissance. L'efficacité de la méthode proposée est observée lors de la réalisation de différentes expériences en comparant nos résultats avec des travaux précédents.

Dans le système « **SRAP1** », nous avons défini la partie de reconnaissance automatique du locuteur comme un processus d'identification de chaque locuteur en analysant la forme spectrale de son signal vocal. Ce processus d'identification destiné à vérifier que la voix analysée correspond bien au locuteur qui est sensée la produire. Le premier sous-système du système **SRAP1** représente, comme illustré sur la figure 4.3, le processus d'identification d'un locuteur à partir de sa voix enregistrée par un microphone, puis de le comparer avec les autres mémorisées dans une référence. Le deuxième sous-système destiné à reconnaître des chiffres arabe prononcé par ce locuteur. Nous distinguons

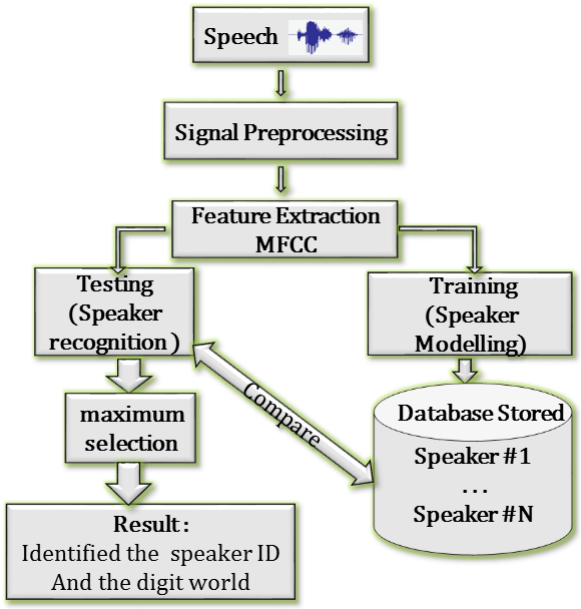

Figure 4.3: Architecture générale du système de reconnaissance de la parole et du locuteur proposé.

deux phases dans le processus d'identification du locuteur: l'apprentissage et le test. Lors de la phase d'apprentissage, un modèle associé à chaque locuteur de la base d'apprentissage. Cette phase se déroule en deux étapes :

- La paramétrisation qui consiste essentiellement à réduire l'information du signal vocal en quantité et en redondance tout en augmentant la discrimination. A la sortie, le signal est représenté par un ensemble de vecteurs de coefficients.
- La modélisation qui consiste à estimer les paramètres d'un modèle mathématique approprié pour les vecteurs de coefficients.

Lors de la phase de test (la tâche de classification), une mesure de similarité est établie entre les vecteurs testés et les données stockées comme références pour chaque locuteur de la base de données. En sortie de cette phase, le système émet la décision (reconnu / non reconnu). Par conséquent, le système peut identifier la personne qui parle et le chiffre qui dit dans notre cas.

4.4.1.2 Tests et résultats du système SRAP1

Dans nos expériences, nous avons utilisé VQ, GMM et le GMM+VQ comme modèles de classification. L'apprentissage du mélange de lois gaussiennes GMM se déroule en deux étapes. La première est une initialisation du modèle par Quantification Vectorielle (algorithme VQ) basée sur l'algorithme de LBG. La seconde étape est une optimisation des paramètres du mélange par l'algorithme classique Expectation Maximisation (algorithme EM). Les vecteurs formant le dictionnaire de quantification (codebook) sont développés en utilisant l'approche VQ-GMM proposée à partir de la voix d'un locuteur spécifique. Ensuite, ils seront comparés aux modèles de référence obtenus en phase d'apprentissage. Dans nos expériences nous avons évalué les trois algorithmes de classification. Au cours de la phase de test, nous utilisons le corpus de test artificiellement bruitée constitué de 450 d'enregistrements audio de 15 locuteurs (10 voix des hommes et 5 voix des femmes) choisies parmi notre base de données ARABDIGITS décrit précédemment (section 2.2). Ces données de test pro-

pres ont été bruitées artificiellement par addition d'un bruit gaussien blanc (AWGN) à différents rapports signal à bruit (5, 10, 15, 20 dB), les données vocales sont enregistrées avec une fréquence d'échantillonnage de 8 KHz en utilisant la fonction AUDIORECORD de l'environnement MATLAB 2018a sous un système Windows de 64 bits. La figure 4.4 montre un exemple d'une représentation temporelle et spectrale d'un signal du corpus. Nous avons effectué une série de 10 tests pour chaque

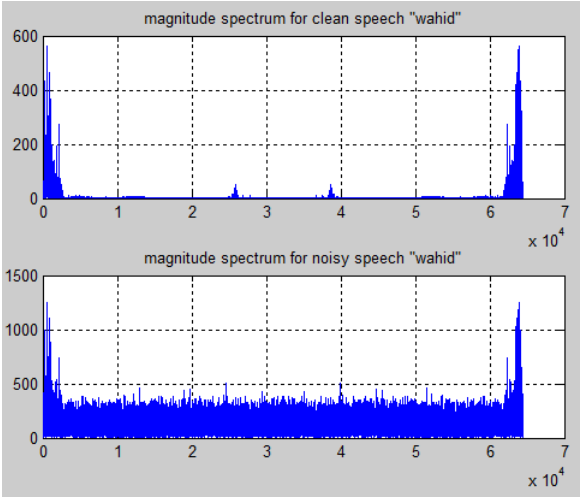

Figure 4.4: Représentation temporelle et spectral (Audiogramme) du mot « wahid » dans le cas normal et bruité enregistré en SNR = 5 dB.

mot dans un environnement calme (peu bruité), et pour les quatre niveaux du rapport signal sur bruit pour les sous-corpus de test construit par 15 locuteurs. Les modèles acoustiques sont entraînés sur la parole propre et le test s'effectue dans l'environnement bruité. Les résultats de ces tests sont récapitulés par les tableaux et les graphiques suivants.
Le pourcentage de reconnaissance est obtenu à partir de ces tests par :

$$Taux\_Rec = \frac{\text{Nombre Total De Mots Reconu}}{\text{Nombre Total des mots de Tests}} \times 100 \tag{4.6}$$

Nous présentons dans ce qui suite les résultats d'évaluation du SRAP1 sous la forme de taux de reconnaissance moyen dans un environnement propre et dans un environnement bruyant, ils sont représentés dans le tableau 4.8 et la figure 4.5 respectivement.

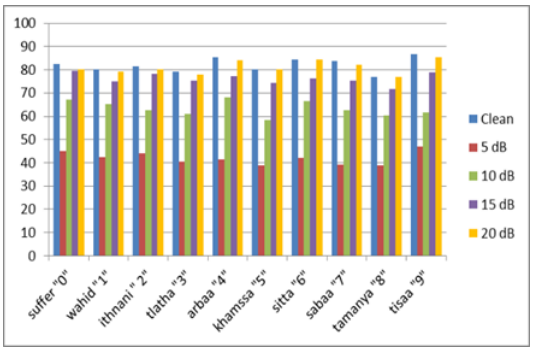

Figure 4.5: Représentation des Taux de reconnaissance pour des chiffres arabes (%) en cas propre et en présence de bruit AWGN

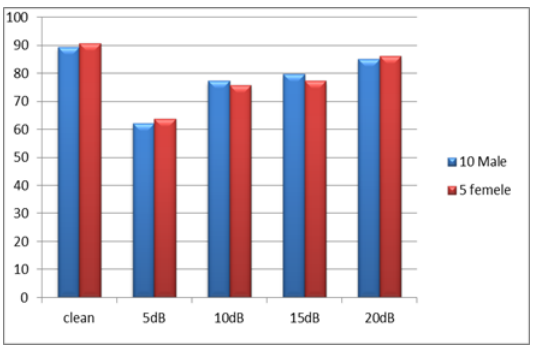

Figure 4.6: Représentation des Taux d'identification (%) obtenus en cas propre et pour des niveaux de bruit AWGN de 5à 20 dB

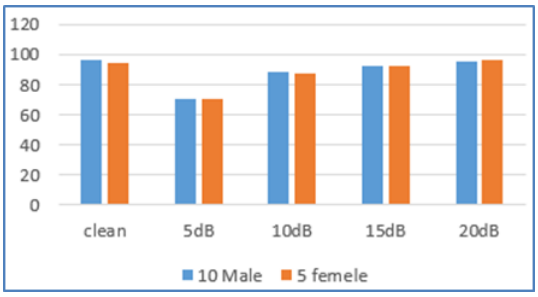

Figure 4.7: Représentation Taux d'identification (%) obtenus dans différents conditions en utilisant le modéle GMM

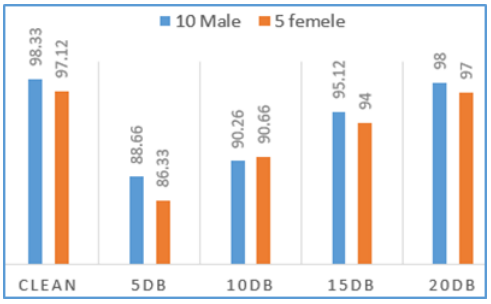

Figure 4.8: Représentation de Taux d'identification (%) obtenus dans les différents conditions en utilisant le modéle hybride GMM+VQ

Table 4.8: Taux de reconnaissance moyen des chiffres arabe (%) avec l'algorithme MFCC + VQ.

Chiffre prononcé	**Propre**	**5dB**	**10dB**	**15dB**	**20dB**
0	82.37	45.17s	67.27	79.66	80.33
1	80.11	42.34	65.17	75.11	79.28
2	81.54	44.06	62.66	78.09	80.05
3	79.08	40.47	60.87	75.33	78.00
4	85.52	41.52	68.12	77.18	84.11
5	80.17	38.78	58.34	74.23	80.17
6	84.42	42.10	66.52	76.12	84.42
7	83.67	39.27	62.71	75.25	82.05
8	77.02	38.96	60.33	71.78	77.02
9	86.63	47.03	61.51	79.00	85.28
Moyenne	**82,05**	**41,97**	**63,35**	**76,17**	**81,07**

Table 4.9: Taux d'identification (%) en utilisant la base de Test pour des niveaux de bruit AWGN de 5 à 20 dB en utilisant MFCC+VQ.

Méthodes	# Locuteurs	propre	5dB	10dB	15dB	20dB
MFCC+VQ	10 hommes	89.43	62.11	77.27	79.66	85.12
	5 femmes	90.66	63.72	75.17	77.26	86.06

Table 4.10: Taux d'identification (%) en utilisant la base de Test dans le cas où le signal parole est bruité avec AWGN de 5 à 20 dB on utilisant : MFCC+GMM.

Méthodes	#Locuteurs	Propre	5dB	10dB	15dB	20dB
MFCC+GMM	10 hommes	96.12	70.26	88.33	92.66	95
	5 femmes	94.66	70	87.67	92.33	96.33

Table 4.11: Taux d'identification (%) en utilisant la base de Test pour des niveaux de bruit AWGN de 5 à 20 dB en utilisant le modèle hybride GMM+VQ

	Propre	5dB	10dB	15dB	20dB
10 hommes	98.33	88.66	90.26	95.12	98
5 femmes	97.12	86.33	90.66	94	97

4.4.1.3 Analyse et discussion des résultats obtenus

À partir des résultats présentés dans le tableau 4.8 et en se basant sur les figures 4.5, nous pouvons conclure que l'effet du bruit n'est pas très important si le SNR est supérieur à 20 dB, dans ce cas nous obtenons approximativement presque la même valeur du taux moyen de reconnaissance obtenue avec les données propres. Nous avons noté que les taux de reconnaissance se dégradent si le niveau du bruit augmente, ceci est due à la dégradation des segments vocaux se dégradent lorsque le SNR <5 dB.
Les résultats des taux d'identification du locuteur sont donnés dans les tableaux 4.9, 4.10 et 4.11 fondé sur les méthodes MFCC+VQ, MFCC+GMM et MFCC+GMM+VQ respectivement, ainsi ces représentations graphiques sont montrées dans les figures 4.6, 4.7 et 4.8 respectivement.
En se basant sur ces résultats, nous constatons que les taux de reconnaissance obtenus avec notre modèle de la combinaison des algorithmes VQ et GMM en présence d'un bruit AWGN sont meilleurs dans la plupart des cas par rapport à ceux obtenus avec le VQ ou GMM seulement. Dans le tableau 4.11, nous avons noté presque les mêmes observations précédentes avec notre base de données de ARABDIGITS, c'est à dire que nous avons trouvé le meilleur taux moyen de reconnaissance égale à 97,12% tester avec la parole propre et 86,33% quand le test s'effectue dans l'environnement bruité à 5 dB en utilisant le modèle hybride MFCC+GMM+VQ. Plus précisément, on obtient une amélioration du taux de reconnaissance de 16,33% relativement à un système de reconnaissance basé sur le modèle MFCC+GMM et une amélioration de 22,61% relativement à un système de reconnaissance basé sur le modèle MFCC+VQ dans le cas de SNR=5 dB.
Les tests de simulation sont réalisés sous l'environnement MATLAB Simulink. Pour cela, nous avons construit une interface GUI comme illustré sur la figure 4.9 pour simplifier le processus de test où le locuteur peut être testé directement par un nouvel enregistrement vocal ou à partir de la base de test. Pendant l'enregistrement, l'utilisateur ajoute du bruit AWGN à sa voix et sélectionne le niveau SNR. Cette interface GUI permet l'enregistrement et la représentation graphique d'un son ainsi elle nous permet de collecter des nouvelles données de test et l'identification de l'ID du locuteur.

En résumé dans cette section, nous avons vu que notre approche fondée sur les coefficients MFCC et le modèle hybride GMM+VQ permet d'améliorer les performances de notre premier système (SRAP1) de reconnaissance automatique du locuteur dans un environnement bruité en présence du bruit artificiel AWGN pour différentes valeurs de SNR. Les résultats expérimentaux que nous avons obtenus montrent que cette méthode bien qu'elle est simple, elle a donné des résultats remarquables pour l'identification des locuteurs prononçant des chiffres arabe isolés dans le cas où le bruit est fort.

Figure 4.9: Interface graphique GUI Matlab pour le system RAP basé sur VQ-GMM.

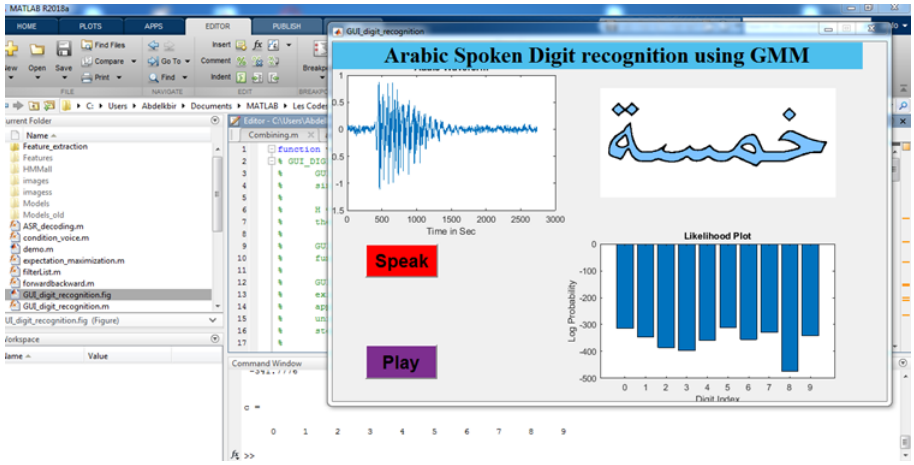

Figure 4.10: Interface graphique GUI Matlab pour le system de RAP basé sur GMM

4.4.2 Développement d'un système de RAP basé sur le Modèle CNN et le corpus SDDN

4.4.2.1 Description du système « SRAP2 »

Dans le papier [135] nous avons reconstruit une base de donnés vocales nommé SDDN très bruité d'après l'ensemble original de données de commandes vocales v0.02[136], que nous avons donné, précédemment, dans la section 2.1 les détails sur ce corpus. Le but général ensemble est de construire une grande base de données vocales dans des conditions bruités pour tous ceux qui travaillent sur la reconnaissance vocale. Afin, de tester les algorithmes de rehaussement de la parole, d'évaluer les performances des méthodes de débruitage du signal de parole et pour examiner la robustesse des systèmes de RAP.

4.4.2.2 Essais avec SDDN

Nous avons utilisé la base de données SDDN de parole pour évaluer des systèmes de RAP basé sur

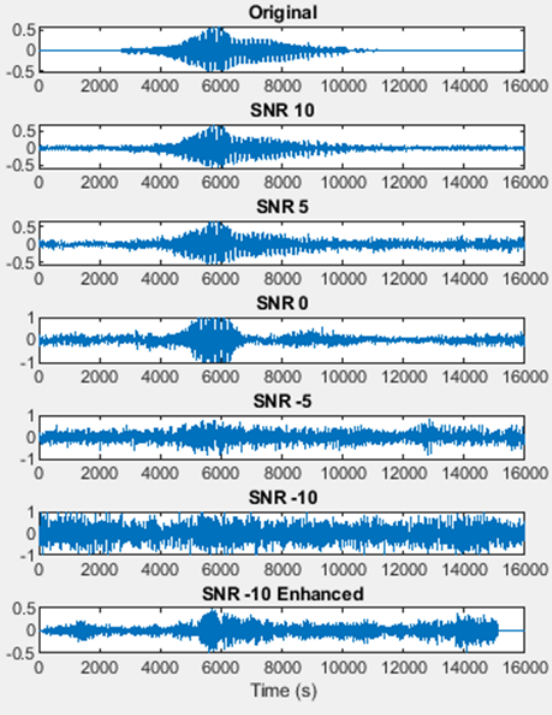

Figure 4.11: zero_SNR_10_Babble_Spk_0b7ee1a0__nohash_4

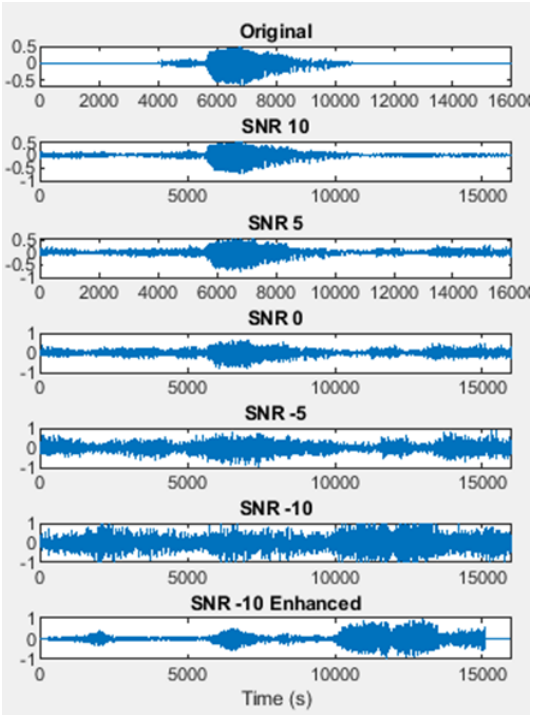

Figure 4.12: Two_SNR_0_Pedestrianarea_Spk_5ba724a7_nohash0

notre approche d'apprentissage avec plusieurs bruits à plusieurs rapports signal sur bruit que nous avons abordé dans cette thèse pour améliorer les performances d'un système de RAP. Cette technique d'injection de bruit, bien que déjà appliquée au calcul neuronal, n'a pas été incluse de manière significative dans l'apprentissage des systèmes de reconnaissance de la parole. Comme cette méthode nécessite la disponibilité des bases de données de grande taille et d'unités du calcul très puissants et plus sophistiqué, nous nous sommes limités provisoirement au quelques expériences simples.

Par exemple, l'un de nos objectifs est d'évaluer les performances des algorithmes d'amélioration de la parole en utilisant la base de données SDDN, nous avons expérimenté l'algorithme d'amélioration de la FFT pour la suppression des bruits de fond afin d'améliorer la qualité audio. Donc, nous avons testé dans différentes conditions de bruit non stationnaires comme : murmure (babble), café, zone piétonne, rue et le bruit rose comme un type stationnaire à différents niveaux de SNR: -10 dB, -5 dB, 0 dB, 5 dB et 10 dB. Les résultats ont montré que ces données acoustiques étaient significatives en termes de taux de reconnaissance pour différents SNR dans tous les différents types des bruits de fond utilisés, ceci implique une meilleure intelligibilité de la parole, même à un niveau de bruit élevé. Les Figures 4.11, 4.12, 4.13 et 4.14 présentent quelques exemples montrant les audiogrammes : le signal d'origine, le signal bruité par bruit réel pour chaque SNR et le signal est amélioré.

Par exemple, la figure 4.11 montre le spectre du mot « zéro» Prononcé par le locuteur nommé «0b7ee1a0-nohash-4», avec l'ajout d'un bruit de type « babble » pour un SNR de 10 dB, 5 dB, 0 dB, -5 dB et -10 dB respectivement. Ce signal lui-même est amélioré ensuite par un algorithme soustractif pour SNR=-10 dB. Les autres exemples sont indiqués par les noms donné pour chaque figure.

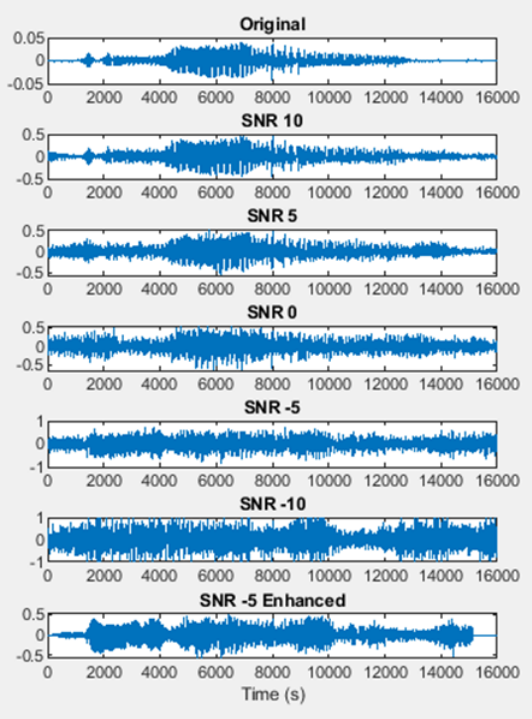

Figure 4.13: ONE_SNR_0_Cafe_Spk_00f0204f_nohash_0

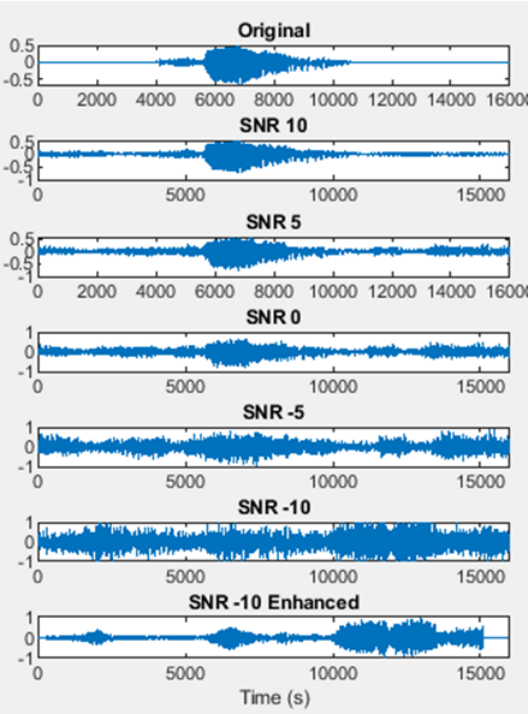

Figure 4.14: three_SNR_10_Pink_Spk_00b01445_nohash_0

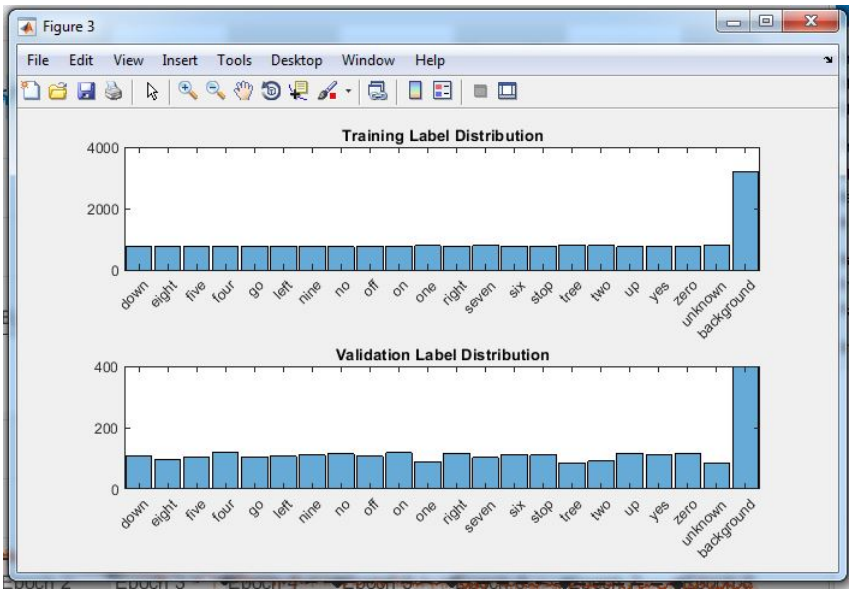

Figure 4.15: Distribution des différentes classes des mots des corpus d'apprentissage et de test

Les figures 4.15 et 4.16 montrent les premières étapes pour réaliser un simple système de reconnaissance de mots isolés SRAP2 fondée sur des réseaux de neurones convolutifs (CNN) en utilisant le corpus SDDN, les deux sous ensemble train et test, ce qui nous permettra d'entraîner et d'évaluer ce système.

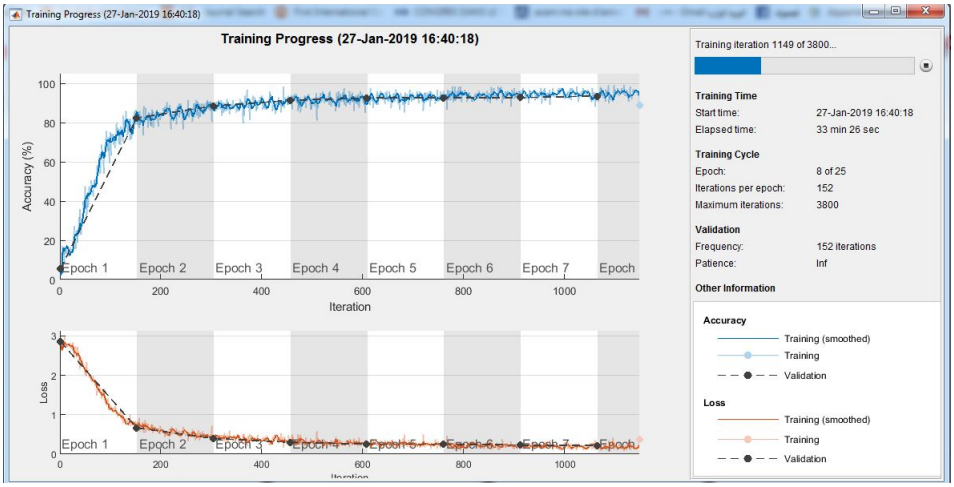

Figure 4.16: Évolution de la précision et de la perte au cours de l'entrainement des CNN avec le corpus SDDN.

4.4.3 Développement d'un système de RAP sous PocketSphinx basé sur le Modèle GMM-HMM et le corpus "NASCIW"

4.4.3.1 L'Architecture du système « SRAP3 »

Dans cette section, nous décrivons comment créer et développer un système de reconnaissance de la parole arabe dans des conditions bruitées en utilisant le Framework open source Pocket Sphinx. Nous l'avons nommé "**SRAP3**". Ce système est construit à partir du système CMUSphinx qui est basé sur le modèle hybride (GMM-HMM) des modèles de Markov cachés et des mixtures de gaussiennes (GMM).

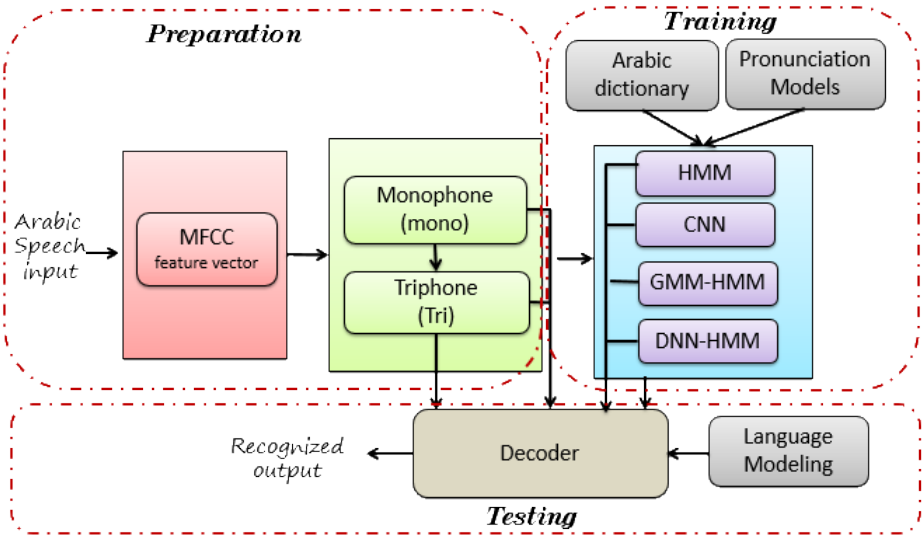

Figure 4.17: Architecture du système de reconnaissance automatique de la parole incluant la procédure d'apprentissage et le décodage.

Nous avons défini précédemment les composantes de ce système dans le chapitre 1. Il se compose par diverses données que l'on peut classer en 3 catégories : les modèles acoustiques décrivant les

entités (les unités de base) à reconnaître, le lexique codant les mots du vocabulaire, et un modèle de langage décrivant la structure des phrases du langage, comme illustré dans la figure 4.17 pour le système SRAP3. Au cours d'une phase préalable d'apprentissage, les différentes entités sont mémorisées par le système. Ces entités peuvent être de différentes formes selon le type de système : mots, phonèmes, syllabes, etc.
La boite à outils Pocket Sphinx est l'une des plus connues et les plus robustes dans le domaine de la reconnaissance vocale, elle contient des outils permettant l'entrainement d'un modèle acoustique, la compilation d'un modèle de langage et différents traitements permettant l'analyse et l'exploitation des signaux de parole.
Donc, pour créer l'environnement de travail et pour la mise en œuvre de notre système SRAP3 sur PocketSphinx, nous suivons ces étapes :
1-Installer les logiciels supplémentaires et les dépendances nécessaires pour l'exécution de pocket-Sphinx qui sont (Python, Les différentes librairies qui composent pocketSphinx et Microsoft Visual Studio : Pour compiler les sources en C afin de produire les exécutables) ;
2-Créer un nouveau modèle acoustique à partir du corpus NASCIW contient (hmm, lm, dictionnaire);
3-Insérer les composantes du modèle (les paramètres acoustiques MFCC obtenus de la phase de paramétrisation, les transcriptions, la liste des phonèmes et le dictionnaire) dans le projet ;
4-Lancer l'entrainement à l'aide de l'outil SphinxTrain ;
5-Sortir des fichiers constituent notre modèle acoustique (means, mixture_weight, transition_matrices, variances, nasciw.mdef) ;
Notons que PocketSphinx a plusieurs modèles acoustiques et linguistiques tels que l'anglais, l'allemand, l'espagnol, l'indien, etc., mais il a la possibilité de le créer, alors nous avons créé le modèle acoustique et le modèle de langage pour le corpus NASCIW. Le modèle acoustique a été formé sur la base d'un corpus d'apprentissage NASCIW comme nous avons montré dans la section 2.3 de ce chapitre. Il y a deux cas de tester du modèle acoustique. La première phase est basée sur les données d'entraînement propres, tandis que la deuxième cas est basée sur des données bruitées artificiellement, respectivement.

4.4.3.2 Création du modèle acoustique

Le modèle acoustique définit les représentations des phonèmes constituant les mots d'un modèle de langage. Il est construit souvent par les techniques d'apprentissage via l'algorithme Baum Welch avec les modèles GMM-HMM. Les performances des modèles obtenus dépendent des caractéristiques des corpus d'apprentissage utilisée comme: la fréquence d'échantillonnage, le style de parole, topologie du HMM utilisé, fonctions de distribution, des observations, nombre d'itérations, ... etc.
Pour pouvoir entraîner notre modèle acoustique nous avons 20h d'enregistrement à partir des données audio du train set NASCIW et les transcriptions correspondantes, nous avons aussi le dictionnaire phonétique qui contient la transcription en phonèmes de tous les mots du corpus (tableau 4.4).
Dans un premier temps, Nous créons un modèle monophone qui est l'élément initial par suite pour créer le modèle triphone. Nous avons utilisé comme unité de base des phones en contexte. Chaque phone en contexte est modélisé par une chaine de Markov cachée à un nombre d'états variés avec

des densités d'observation multi-gaussiennes. Le nombre d'états liés et le nombre de mélanges par état sont réglés à l'aide du corpus d'entraînement. Nous avons évalué différents mélanges par état (8,16,64 et 256). Ces paramètres ont été réglé en utilisant la boite à outils PocketSphinx.
Pour des étapes du modèle acoustique on utilise le modèle de Markov caché avec le modèle de mélange gaussien (GMM) pour extraire les vecteurs de caractéristiques acoustiques, alors, nous avons choisi d'utiliser les caractéristiques utilisées dans la construction du modèle sont les coefficients MFCC, de dimension égale à 39, est constitué des 13 premiers coefficients MFCC augmentés de leurs dérivées première et seconde pour obtenir un vecteur de caractéristiques de 39 dimensions par trame. Ces vecteurs sont normalisés par rapport à la moyenne et la variance sur une phrase. La normalisation par rapport à la variance permet de diminuer la variabilité par rapport au locuteur. L'extraction des paramètres est réalisée avec l'outil Wave2feat de SPHINX.
Une fois la phase de paramétrisation terminée nous pouvons alors construire notre modèle acoustique et lancer l'entrainement à l'aide de l'outil SphinxTrain. La structure des fichiers de notre modèle extrait de la base de données NASCIW est illustrée sur la figure 4.18:

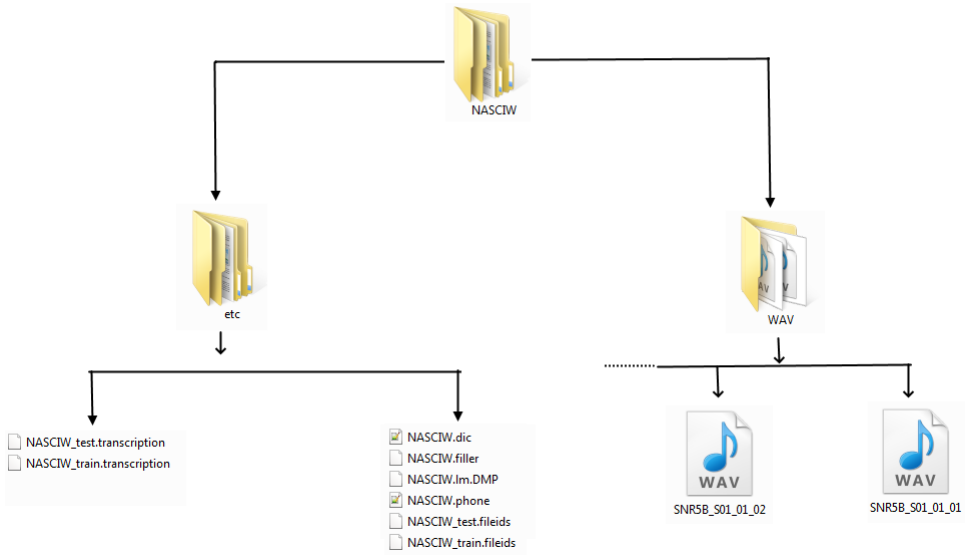

Figure 4.18: La structure des fichiers SphinxTrain de la base de données (NASCIW).

Pour les fichiers du dossier wav :

- NASCIW.dic : Dictionnaire phonétique qui définie la décomposition phonétique pour chaque mot ;
- NASCIW.filler : liste de caractères à remplacer par le caractère « SIL » ;
- NASCIW.lm.DMP : Modèle de langage utilisé ;
- NASCIW.phone : Liste des phonèmes utilisés ;
- NASCIW_train.fileids : Liste des fichiers d'entraînement ;
- NASCIW_test.fileids : Liste des fichiers de test ;
- NASCIW_train.transcription : Transcription textuelle des fichiers d'entraînement ;
- NASCIW_test.transcription : Transcription textuelle des fichiers de test ;

Cette étape de développement est résumée par le schéma présenté dans la figure 4.19 suivante, qui résume les différents fichiers d'entrée et de sortie en utilisant l'outil SphinxTrain pour la construction du modèle acoustique et notre propre modèle de langage. Il est composé d'un ensemble de script PERL en outre des fichiers de configuration.

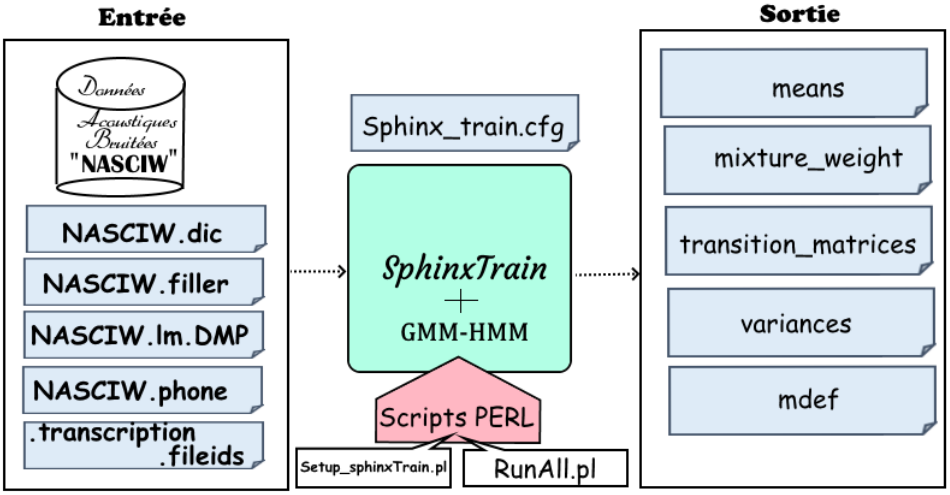

Figure 4.19: Schéma représentant la création du modèle acoustique avec SphinxTrain.

Par conséquent, les fichiers : (means, mixture_weight, transition_matrices, variances, NASCIW.mdef) obtenus constituent notre modèle acoustique.

4.4.3.3 Création du modèle de langage

Modèle de langage (Language model ou grammar model) c'est un modèle qui définit l'usage des mots dans une application, il contient des probabilités des mots et des combinaisons de mots des données d'apprentissage (NASCIW). Ces probabilités sont estimées à partir de ces données sous différents modèles:1-gram, bi-gram, 3-gram, ...etc. Chaque mot dans le modèle de langue doit être dans le dictionnaire de prononciation. Il existe plusieurs outils et façons de création de ces modèles comme SRILM « SRI Language Modeling Toolkit», LMtools « Sphinx Knowledge Base Tool» ... etc. Ici, nous utilisons le service Web en ligne rapide proposé par CMUSphinx nommé LMTool Base[3], nous allons l'utilisé pour générer les fichiers nécessaires. Tout d'abord, nous créons un fichier texte nommé NASCIW.txt. Dans ce fichier, nous mettrons les mots de notre corpus l'un après l'autre. Ensuite, en accédant au lien de l'outil LMTool, puis en cliquant simplement sur le bouton «Parcourir…», puis on sélectionne le fichier NASCIW.txt que nous avons créé, puis on compile. Un numéro à quatre chiffres est associé à cette nouvelle base de données (0552 dans notre cas) et des fichiers sont produits. Le tout est contenu dans le fichier compressé TAR0552.tgz que nous téléchargeons (voir figure 4.20).
On décompresse ce fichier, puis on récupère les deux fichiers 0552.dic et 0552.lm que nous plaçons convenablement ici: *"C:/ProjectSphinx/NASCIW/etc/ "*
Le fichier du notre modèle de langage 0552.lm présenté dans la figure 4.20, qu'on le renomme par NASCIW.lm.DMP, c'est un fichier de format DMP binaire qui nous permet de générer des en-

[3] *http : //www.speech.cs.cmu.edu/tools/lmtool − new.html*

trées de dictionnaire avec des prononciations phonétiques pour des mots. Pour le fichier 0552.dic (NASCIW.dic) représente le dictionnaire (modèle phonétique), il réalise la correspondance entre les mots et les phonèmes.

Notons qu'on a utilisé aussi un autre type du modèle de langage de type *.gram (grammaire JSGF) au lieu du modèle de langage DMP, c'est un fichier texte simple qui va indiquer les mots attendus, leur hiérarchie. C'est un modèle de langage 3-gramme et des modèles acoustiques adaptés au genre du locuteur (homme/femme) et aux conditions acoustiques (les conditions d'enregistrement des donnés). Donc, on enregistre la grammaire dans un fichier texte appelé "NASCIW.gram", puis exécutez sphinx par le code :

pocketsphinx_continuous -inmic "yes" -hmm "C:\ProjectSphinx\NASCIW\output\ Nasciw.ci_cont" -dict "C:\ProjectSphinx\NASCIW\output\NASCIW.dic" -jsgf "C:\ProjectSphinx\NASCIW\output\ NASCIW.gram"

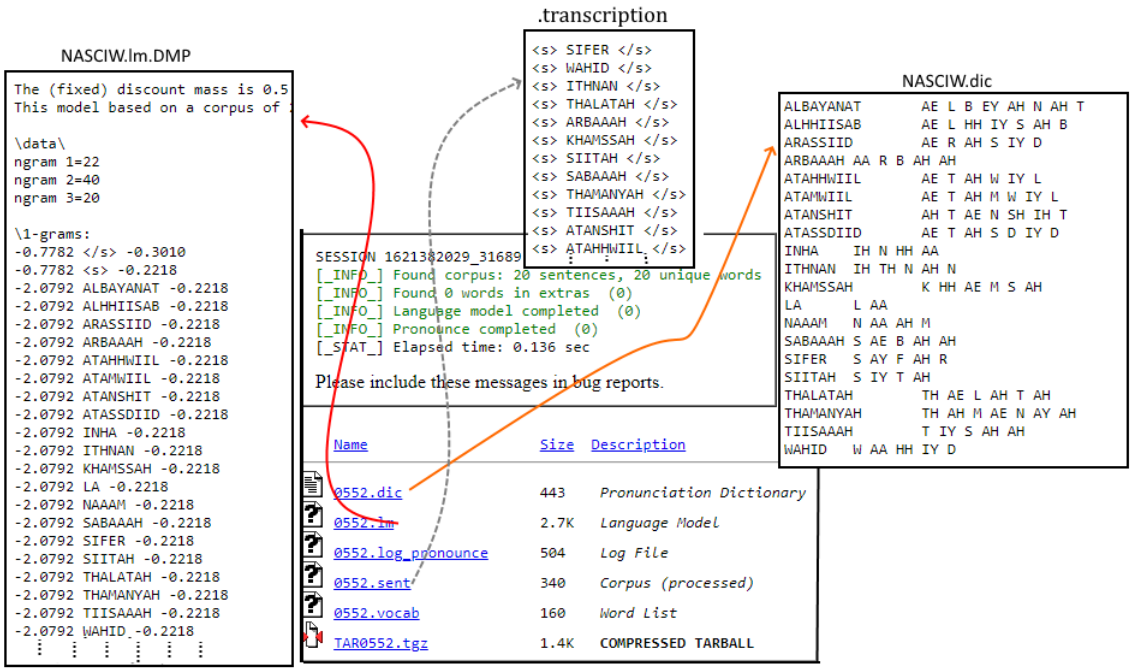

Figure 4.20: Extrait du modèle de langage créé par l'outil LMtool.

Nasciw.gram

```
#JSGF V1.0;
/**
 * JSGF  Grammar for Arabic speech corpus for isolated words
 */
grammar NASCIW;
public <B> = (oh | SIFER  | WAHID | ITHNAN | THALATAH | ARBAAAH | KHAMSSAH | SIITAH | SABAAAH | THAMANYAH | TIISAAAH |
              ATANSHIT | ATAHHWIIL | ARASSIID | ATASSDIID | NAAAM | LA | ATAMWIIL | ALBAYANAT | ALHHIISAB |INHA) * ;
```

Figure 4.21: Extrait du fichier de grammaire du corpus NASCIW.

4.4.3.4 Le dictionnaire phonétique

Nous avons utilisé une notation latine pour représenter les mots et ces phonèmes du corpus arabe NASCIW. Notre fichier des phonèmes contient 24 phonèmes comme indique dans le tableau 4.12.

Table 4.12: Symboles des phonèmes utilisés pour NASCIW.

Symbole	Alphabet Arabe	Translittération
AA	أٌ	Alef
AH	أه	Ah
B	ب	Ba’
T	ت	Ta’
TH	ث	Tha’
HH	ح	Ha’
KH	خ	Kha’
D	د	Da’
R	ر	Ra’
S	س	Sin
SH	ش	Chin
SS	ص	Sad
AI	ع	Ain
F	ف	Fa’
L	ل	Lam
M	م	Mim
N	ن	Non
H	ه	Ha’
W	و	Waw
Y	ي	Ya’
AA	عَ	Fatha
OU	عُ	Damma
IY	عِ	Kasra
AE	عْ	Sokon

Dans le fichier de dictionnaire NASCIW.dic, on définis chaque mot du corpus avec l’unité phonétique comme indiqué dans le tableau 4.13.

4.4.3.5 Tests, résultats et discutions du système SRAP3

Nos expériences sur le système SRAP3 consistent à tester plusieurs facteurs et paramètres en utilisant la plateforme Pocketsphinx. Parmi eux, l’effet de la quantité de bruit injecté dans les données d’entraînement, l’effet du nombre gaussien également et l’effet d’états HMM. Tout d’abord, nous avons évalué un système multi-conditions, entraîné avec les 3 types de bruit en 3 niveaux de SNR et avec les conditions propres, afin de voir quelles sont les types des bruits additifs qui affectent plus particulièrement les performances du système SRAP3, ensuite nous allons voir l’effet du nombre de composantes gaussiennes du modèle GMM sur les performances globales de la classification des 24 phonèmes. Enfin, nous testons l’influence de la variation du nombre des états HMM (3 et 5) sur la robustesse du système dans le modèle acoustique.

Table 4.13: Structure du fichier NASCIW.dic

Mot arabe	Mot en lettres latin	Décomposition phonétique
صِفر	SIFER	S IY F AH R
وَاحد	WAHID	W AA HH IY D
اثْنَان	ITHNAN	IY TH N AE N
ثَلَاثة	THALATAH	TH AA L AE T AH
أربَعة	ARBAAAH	AE R B AI AH
خمسَة	KHAMSAH	KH AE M S AH
ستَّة	SITAH	S IY T AH
سبعة	SABAAAH	S AE B AI AH
ثمَانية	THAMANEYAH	TH AE M AE N Y AH
تِسعة	TISAAAH	T IY S AI AH
التنشيط	ATANSHIT	AE T AA N SH IY T
التحويل	ATAHHWIIL	AE T AA HH W IY L
الرصيد	ARASSIID	AE R AA SS IY D
التسْدِيد	ATASSDIID	AE T AA S D IY D
نعَم	NAAAM	N AI AA M
لَا	LA	L AA
التَّمْويل	ATAMWIIL	AE T AA M W IY L
البيَانَات	ALBAYANAT	AE L B AA Y AE N AE T
الحِسَاب	ALHHIISAB	AE L HH IY S AE B
انهَاء	INHA	IY N H AE

a)-Test 1: Effet du bruit additif et le niveau de SNR sur les performances du système proposé

Les tableaux 4.14 ci-dessous montrent les taux de reconnaissance pour les séries d'expériences du système SRAP3 qui est basé sur la méthode de classification GMM-HMM avec l'outil open source PocketSphinx nous rappelons que toutes nos expériences, l'apprentissage du système de reconnaissance est fait dans des conditions bruités et le test se fait dans conditions bruitées et non bruitées. Les meilleurs résultats en termes de taux de reconnaissance de mots (WRR) sont indiqués en gras.

En générale, les résultats présentés dans le tableau 4.14 montrent que les performances du système SRAP3 se dégradent en présence de bruit dans les données de test à des degrés divers selon les types de bruit. Nous avons effectué des tests pour déférents conditions. A travers ces tests, nous avons vu que pour les données de test mélangés avec le bruit de type « babble » n'affecte pas beaucoup plus par rapport à l'effet des deux autres types. Dans ce cas, nous avons trouvé le meilleur taux moyen de reconnaissance égale à 80,4% pour SNR=5 dB en utilisant le modèle hybride GMM-HMM. Mais pour des niveaux de bruit élevés, pour des SNR $\leq$ 0 dB les meilleurs taux de reconnaissance

Table 4.14: Taux de reconnaissance de mots (WRR%)obtenus pour différentes conditions de test de SRAP3.

Types de bruit utilisés et le niveaux de SNR		WRR dans toutes les conditions
Dans les conditions propres		96.2
Le bruit bavardage (Babble noise)	-5 dB	24.6
	0 dB	42.6
	5 dB	80.4
Bruit de la rue (Street noise)	-5 dB	24.8
	0 dB	44.2
	5 dB	75.1
Bruit rose (Pink noise)	-5 dB	17.9
	0 dB	35.1
	5 dB	70.5
Moyenne (-5 at 5 dB)		46.13

sont obtenus en présence du bruit « Street » dans les données de test à SNR=-5 dB, on a trouvé 24,8%, l'apport est de plus de 0.2% par rapport au bruit « babble », et plus de 6,9% sur le taux de reconnaissance par rapport au bruit « rose ». Nous remarquons aussi également que les performances du système en utilisant le bruit rose sont nettement moins bonnes que pour les autres types de bruit (babble et street).

Donc, on peut déduire d'après les résultats présentés ci-dessus (tableau 4.14) que le système hybride GMM-HMM basé sur Pocketsphnix est plus performant avec un taux de: 96,2% en conditions propres et 46,13 % comme taux moyenne en conditions bruités. C'est un taux important, autant plus que l'entraînement a été réalisé sur des données bruités. Cela est dû au fait que nous avons construit des modèles acoustiques robustes au bruit avec l'entraînement multi-conditionnelle du corpus NASCIW. Ainsi, que Pocketsphinx utilise des de fonctionnalités robustes au bruit: soustraction spectrale et masquage temporel.

b)-Test 2: Effet de la taille du nombre de mélanges de gaussiennes

Dans ce test, nous avons testé les performances de notre système SRAP3 en changeons les valeurs de mélange gaussien (8,16 ,64 et 256) dans les conditions mentionnées précédemment.

Les résultats obtenus par le système SRAP3 du WRR en fonction et du type et niveau de bruit ainsi le nombre de gaussien sont présenté dans le tableaux 4.15 et la figure 4.21 respectivement.

Table 4.15: Effet du nombre de mélange Gaussien à différents niveaux de SNR du bruit additif "babble" sur le taux de reconnaissance (WRR%).

Nombre de mélange Gaussien	Base de test Normal SNR > 20	**Bruit " Babble"** -5 dB	0 dB	5 dB	**Moyenne (SNRs)**
8	90.1	**25.6**	43.6	75	48.06
16	92.2	24.8	44.2	75	48.03
64	94.4	**25.6**	**45.2**	71.5	47.43
256	**96.2**	24.6	42.6	**80.4**	**49.2**

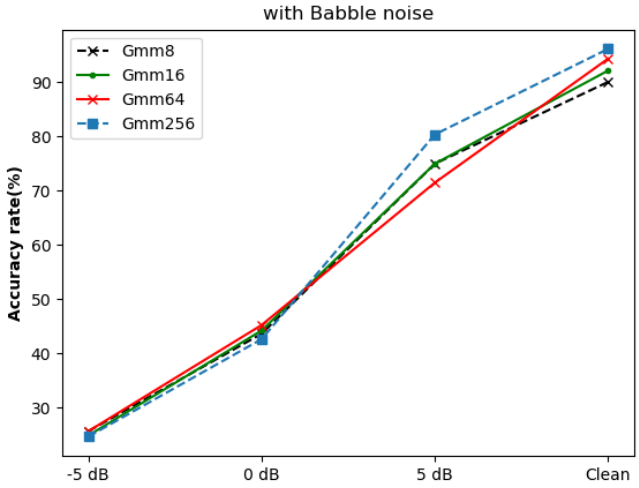

Figure 4.22: Evolution des taux de reconnaissance obtenus en fonction des nombres de gaussiennes à différents niveaux de SNR du bruit additif "babble".

Dans cette expérience, nous avons changé le nombre de mélanges de gaussiennes dans chaque condition de test (entre 8 et 256). Nous remarquons, d'après les résultats obtenus que le meilleur WRR pour des données propres est (96,2%) obtenu avec 256 mélanges de GMM. Pour le test avec les données bruitées en présence du bruit bavardage à (SNR=-5 ,0 et 5 dB), le meilleure taux moyenne WRR est 49,2% obtenue avec 256 mélanges de gaussiennes. Nous constatons, d'après la figure 4.22, que la tendance des performances (WRR) des conditions des systèmes va dans le sens croissant quand nous augmentons le nombre de mélanges de gaussiennes. Par conséquent, le WRR a augmenté de manière significative lorsque le nombre de mélanges a augmenté dans le cas que les données moins bruitées. Cependant, cette remarque n'est pas valable quand les données sont plus bruitées (0 dB et -5 dB). En effet, nous avons eu une dégradation de performance pour un mélange de 256 gaussiennes, dans ce cas (0 dB et -5 dB) le meilleur taux moyenne de reconnaissance est de 35,4%, et il est obtenu avec des modèles gaussiens comportant 64 composantes.
En général, plus le nombre de mélanges de gaussiennes augmente, plus il est possible de modéliser beaucoup les accents dans le modèle acoustique qui a surentraîné même dans toutes les conditions, c'est-à-dire représenté des données qui n'existe pas dans l'espace de vecteurs acoustiques. Mais, généralement l'utilisation d'un nombre trop important de gaussiennes ne conduira donc pas nécessairement à une meilleure reconnaissance.
Dans le cas pratique, le nombre de gaussiennes du modèle GMM est choisi en fonction des données dont nous disposons. Donc, la taille du nombre de mélanges de gaussiennes nécessitent une plus grande quantité de données d'apprentissage.

c)-Test 3: Effet du Nombre d'états par HMM

Afin d'évaluer l'impact du nombre d'états par HMM sur les performance de notre système SRAP3 au niveau des modèles acoustiques, nous avons changé dans le fichier des configurations "sphinx_train.cfg"

concernant la line
$CFG_STATESPERHMM= {2,3,5} qui représente le paramètre spécifiant le nombre d'états qui représente un HMM. Nous avons entraîné et testé le système dans Pocketsphinx en utilisant trois cas de nombre des états par HMM {2 état, 3 état et 5 état} à 64 mélanges de gaussiennes. Les résultats des taux de reconnaissance des mots des trois tests sont présentés dans le tableau 4.16.

Table 4.16: Effet du nombre d'états par HMM en fonction du SNR du bruit additif "babble" et pour GMM=64 sur le taux de reconnaissance (WRR%).

Nombre des états HMM	Base de test Normal SNR>20	**Base de test bruité par le bruit 'Babble'** -5 dB	0 dB	5 dB	Moyenne (SNRs)
2	87.2	16.4	18.2	65.4	33.33
3	**96.2**	**24.6**	**42.6**	**80.4**	**49.2**
5	90.4	25.6	45.2	71.5	47.43

Comme le montre les résultats affichés dans tableau 4.16, les meilleurs taux ont été obtenus dans le cas de 3 état par HMM à toutes les conditions bruités ou propres. On peut déduire que, le changement de nombre d'états par HMM n'est pas lié à la nature des données. Nous remarquons aussi sur la figure 4.22 que le niveau du bruit « babble » n'influe pas sur les performances : le changement du nombre d'états par HMM semble indépendant aux niveaux de bruit. Mais, en général il affecte les performances des systèmes.

4.4.4 Développement d'un système de RAP sous KALDI basé sur le Modèle DNN-HMM et le corpus "NASCIW"

4.4.4.1 Présentation du système « SRAP4 »

L'objectif de cette section est de présenter et d'expliquer de manière générale les différentes étapes nécessaires d'un système de reconnaissance de la parole arabe dans un milieu bruité en utilisant la boîte à outils open source Kaldi[99] , en le nommant SRAP4. Elles sont presque les mêmes étapes décrites dans la section 4.3 pour le système SRAP3, comme illustré sur la figure 4.17. Nous avons déjà présenté l'architecture simplifié des fichiers librairies de cet outil dans le chapitre 2 section 8.2. Les étapes nécessaires à la création de notre système SRAP4 sur Kaldi sont décrite dans la figure 4.17.
Ils se composent de quatre sous-modules important (figure 4.17), à savoir :
- Paramétrisation ou bien Extraction des caractéristiques MFCC avec le modèle acoustique, cette étape permet d'extraire des caractéristiques paramétriques du signal et de représenter digitalement des signaux d'entrée et de faciliter le traitement des étapes d'apprentissage et de décodage.
-Modèle acoustique provient d'un entraînement réalisé à partir des enregistrements du corpus,

représenté par des chaînes de Markov cachées. Il consiste à identifier le phonème prononcé en utilisant les paramètres acoustiques MFCC extrait de chaque segment (20ms) du signal vocal.
-Modèle phonétiques de mots (aussi appelé modèle de prononciation ou dictionnaire phonétique ou lexique) qui contient la liste de toutes les prononciations des mots que le système sera en mesure de reconnaître, il permet de faire le lien entre le représentation acoustique d'un mot et sa représentation phonémique.
-Modèle de langage (ou la grammaire) qui définit comment les mots peuvent être connectés à chacun.
Autrement dit, à partir de données d'entraînement NASCIW, nous pouvons entraîner un modèle statistique de la parole. Le modèle donne la probabilité qu'un mot donné produit un signal acoustique. à partir de ce signal acoustique, nous pouvons déterminer la phrase la plus probable.

4.4.4.2 Préparation des donnés et réalisation

Nous avons commencé dans un premier temps par l'installation de l'environnement Kaldi et préparant ces composantes et ces ressources nécessaires. Nous nous sommes appuyés sur le site officiel (https://kaldi-asr.org) en particulier le tutoriel de Kaldi (Kaldi for Dummies tutorial[4]) pour construire notre système en utilisant le corpus NASCIW pour l'entraînement. Toutes les expériences ont été effectuées sur un seul ordinateur avec les configurations suivantes: processeur Intel® Core (TM) i5-4310U @ 2 ,20 GHZ 2,20 GHz × 8Go de RAM, de carte graphique NVIDIA GeForce GT 720M prend en charge CUDA (Compute Unified Device Architecture, l'architecture de calcul parallèle créée par NVidia TM), du système Ubuntu 18.04 avec 64 bits comprenant Python 3.6.
Nous résumons ci-dessous tous les étapes séquentiellement que nous avons fait durant la préparation de l'environnement de travail de la boite à outil Kaldi :

- Dans la première étape on a téléchargé et installé la machine virtuelle '**Oracle VM VirtualBox 6.1.2'** dans notre ordinateur **«** i5-4310U » ordinateur sous Windows 7 64bits ;
- Installation du système d'exploitation Ubuntu 18.04 64bits sur la machine virtuelle, pendant cette étape, on doit choisir les paramètres matériels suffisante pour l'environnement Kaldi, on a réservé 100G de la taille du disque, 4G du la mémoire de mon ordinateur et 2 processeurs ;
- Ensuite, on a installé Kaldi sous Ubuntu 18.04 en suivant le tutoriel ;
- Compilation Kaldi en assurant que tous les composants sont installés avant la compilation ;
- Enfin, on configure la configuration de kaldi, et on l'exécute et on prépare nos propres fichiers pour le système SRAP3.
  Dans les notes suivantes on présente les étapes de préparation des donnés NASCIW.
- La tâche initiale consiste à organiser correctement les données selon le format kaldi qui comprend les fichiers généraux wav.scp, utt2spk, spk2utt, texte, nous créons donc un dossier nommé NASCIW dans le répertoire kaldi (la figure 4.23).
- Dans le dossier générale du projet **NASCIW**, nous créons dans le dossier **data,** deux sous-dossier **test** et **train**. Ces dossiers contiennent les fichiers audio du format wav comme nous

[4] $https://kaldi-asr.org/doc/kaldi\_for\_dummies.html$

l'avons déjà organisé (voir section 2.3 du présent chapitre).

- Nous Créons un fichier wav.scp dans le répertoire **train**.

Dans Kaldi ASR, nous avons divisé nos données en données acoustiques et données linguistiques [140].

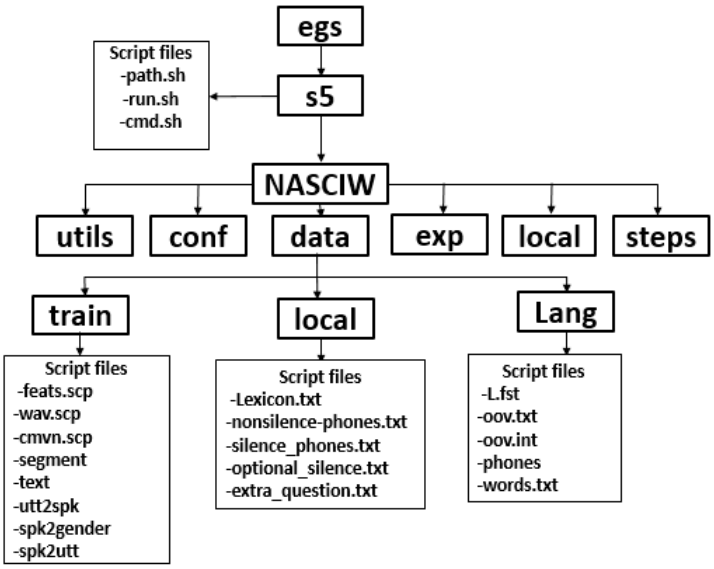

Figure 4.23: Structure des répertoires Kaldi pour les données NASCIW.

- Les métadonnées utilisées pour les données acoustiques sont données ci-dessous:
- **spk2utt:** ce fichier contient le mappage entre le locuteur et tous ses prononciations des mots.
- **utt2spk :** Il donne pour chaque identifiant d'utterance l'identifiant du locuteur qui l'a prononcée;
- **texte :** le fichier de transcription. Il regroupe l'identifiant de l'utterance et sa transcription ;
- **corpus.text:** il contient toute la transcription d'énoncé qui est utilisée pour construire le modèle ;
- **spk2gender:** correspondance entre le locuteur et le sexe ;
- **utteranceID :** il contient le chemin du fichier enregistré.wav ainsi que l'ID des locuteurs ;
- **wav.scp :** il donne, pour chaque identifiant de son, le chemin absolu ;
- Les métadonnées pour la préparation des données linguistiques obligatoires pour Kaldi sont:
- **lexicon.text:** il contient les transcriptions phonétiques de chaque mot (NASCIW : de 20 mots).
- **nonsilence_phone.text:** contient tous les phonétiques utilisés pour préparer la base de données.
- **silence_phone.text:** contient la liste des phonétiques silencieux.

La structure des répertoires Kaldi pour la préparation des données NASCIW est représentée dans la figure 4.21. La structure complète du système SRAP4 basé sur le modèle hybrides DNN-HMM sous Kaldi a été présenté dans la figure 4.17. Le corpus NASCIW est créé dans le répertoire principal Kaldi comme le montre la figure 4.24 qui a été entraîné en utilisant 96000 occurrences de 20 mots. Nous allons utiliser ce corpus pour entraîner les modèles acoustiques du système.
Après avoir préparé les données, nous avons utilisé les scripts suivants:

— les fichiers « mfcc.conf » et « decode.config » sont des fichiers de configuration ;
— le fichier « cmd.sh » permet au système d'utiliser le fichier run.sh plutôt que le CPU en ligne;
— le fichier « path.sh » définit la racine du dossier Kaldi et le chemin vers les outils utilisés ;
— le fichier « run.sh » représente le pipeline et lance les différentes étapes vers le WER.

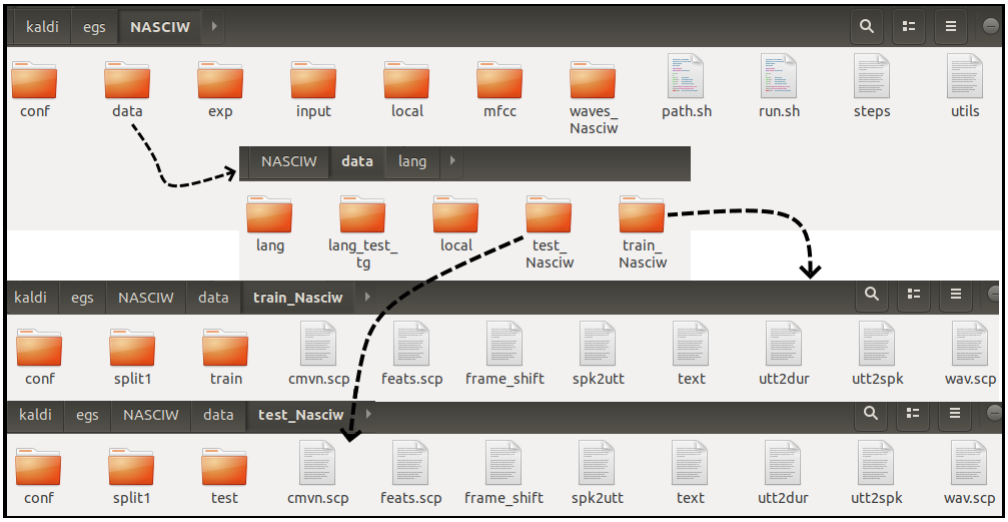

Figure 4.24: Structure des répertoires Kaldi pour le corpus NASCIW.

4.4.4.3 Création du modèle acoustique et de langage

Tout d'abord, nous entraînons un modèle acoustique HMM-GMM sur des paramètres MFCC. Le modèle est entraîné à partir des données audio du train_NASCIW. Nous avons commencé le processus d'apprentissage par l'entrainement d'un modèle acoustique monophone (nommé mono) qui a été utilisé pour effectuer un alignement forcé entre les signaux et les états HMM en exploitant les transcriptions références de notre corpus NASCIW, ce modèle entraîné sur 20H d'enregistrement du corpus NASCIW est entraîné pour reconnaître un seul phonème à la fois, le modèle de langage et le dictionnaire de mots utilisent ces phonèmes avec les HMM pour décoder le mot prononcé. Ensuite, par l'entrainement d'un modèle acoustique un triphone (nommé tri) de 13 vecteurs de caractéristiques MFCC, ce modèle acoustique est entraînée pour reconnaître un groupe de 3 phonèmes, ce qui permet au modèle de langage de disposer d'un contexte phonétique plus large pour retrouver les mots prononcés. Enfin, nous entraînons un modèle HMM-DNN sur les mêmes paramètres acoustiques MFCC en utilisant l'alignement produit par le modèle HMM-GMM.
Notre modèle acoustique neuronal est composé par les fonctionnalités normalisées dans l'apprentissage du modèle DNN-HMM sur les couches cachées non linéaires. Nous utilisons notre nouveau système DNN-HMM pour générer un nouvel alignement forcé unique pour l'ensemble d'apprentissage NASCIW, car il s'agit de l'approche la plus standard lors de la construction de modèles acoustiques basé sur DNN, en se basant sur les travaux de Senior et al[140]. Nous avons entraîné plusieurs réseaux DNN avec trois couches cachées avec 256-512-1024 unités dans chaque couche cachée respectivement et avec la fonction *softmax* dans la couche de sortie. La formation DNN a été effectuée en utilisant un GPU sur une seule machine en utilisant CUDA (Compute Unified Device Architecture, l'architecture de calcul parallèle créée par NVidia TM).

On résume les quatre étapes pour la modélisation acoustique basée sur DNN comme suit:

- Extraction de fonctionnalités (13 MFCC)
- la construction du système monophone (mono)
- la construction du système triphone (tri)
- la formation du modèle DNN-HMM final.

Pour la création du notre modèle de langage pour le système SRAP4, nous avons utilisé un modèle de langage initial bi-gramme (n_gram = 2), qui est estimé à partir de la transcription des données d'entraînement afin de ne pas introduire d'information supplémentaire au niveau linguistique. Nous nous avons utilisé ensuite, l'outil SRILM [5](The SRI Language Modeling Toolkit) développé par SRI, le laboratoire international de technologie et de recherche de la parole en Californie depuis 1995, qui nous permet de créer et de générer les modèles de langage statistiques.
Pour le faire, dans l'outil SRILM nous avons fourni le fichier des transcriptions des données d'apprentissage (train.txt) ainsi que l'ordre du n-gram désiré (3-gramme dans notre cas). Le résultat est donné sous la forme d'un fichier ARPA « **lm.arpa** », qui sera transformé par la suite au format FST nommé « **G.fst** » par l'intermédiaire d'un script Kaldi. Ce fichier décrit la grammaire de la langue, sous forme de transition à états finis. Il est généré à partir du modèle de langage « **lm.arpa** » précédemment créé. Finalement, les fichiers du modèle de langage sont créés dans le répertoire *data* dans les deux sous dossiers *local* et *lang* comme le montré la figure 4.22 de structure des dossiers. Enfin, Pour tester le modèle, on décode les données du corpus de test. Le script ***align_si.sh*** génère les alignements monophone utilisés pour l'entraînement du second modèle acoustique, le modèle triphone.
Alors, deux nouveaux modèles acoustiques sont entraînés à partir de cette nouvelle base de données bruitées(NASCIW). Pour tester les performances des modèles, on génère plusieurs sets de test correspondant respectivement à des SNR de -5; 0; 5; clean (SNR>20). Les résultats obtenus sont présentés et discutés dans la section suivante.

4.4.4.4 Résultats expérimentaux du système SRAP4 et discutions

La partie expérimentale de cette recherche s'est concentrée sur l'analyse de la performance de système SRAP4 à partir des trois aspects suivants: le choix optimal du modèle acoustique approprié entre le premier le modèle acoustique monophone GMM-HMM vs un deuxième modèle acoustique DNN-HMM triphoniques, l'effet de type et le niveau du bruit choisi dans la base d'apprentissage et de test, et enfin, l'impact du nombre des couchés cachés dans l'apprentissage du modèle DNN. En outre, une comparaison entre le système SRAP3 implémenté avec la plate-forme open source PocketSphinx et le système SRAP4 réalisé par la boîte à outils KALDI sera présentée dans la dernier section des expériences. Les résultats obtenus sont présentés dans les tableaux et les figures sous-dessus, nous rappelons que toutes nos expériences, l'apprentissage du système de reconnaissance est fait dans des conditions bruitées.
a)-Test 1: Effet du bruit additif sur les performances du système proposé

[5] *http ://www.speech.sri.com/projects/srilm/*

Table 4.17: Taux de reconnaissance de mots (WRR%)obtenus pour différentes conditions de test pour SRAP4.

Types de bruit utilisés et le niveaux de SNR		Modèle Acoustique	
		GMM-HMM monophone	DNN-HMM triphone
Clean		92.4	**97.1**
Le bruit bavardage (Babble noise)	-5 dB	20.8	**33.5**
	0 dB	44.2	46.7
	5 dB	45	**81.4**
Bruit de la rue (Street noise)	-5 dB	22.2	32
	0 dB	40.6	**49.8**
	5 dB	65.4	76.1
Bruit rose (Pink noise)	-5 dB	10.2	21.4
	0 dB	20.6	39.2
	5 dB	65.5	70.8
Moyenne (-5 at 5 dB)		37,17	**50.1**

Les résultats présentés dans le tableau 4.17 montrent que les performances du système de reconnaissance automatique de la parole arabe (SRAP4) réalisée par la boîte à outils KALDI se dégradent généralement en présence de bruit. Les meilleurs résultats ont été obtenus avec le système basé sur le modèle hybride DNN-HMM triphone dans tous les conditions de test. Ainsi, nous avons obtenu plus de 97.1% comme taux de mots reconnu dans les conditions de test propres et 50.1% taux moyenne des mots avec les données bruitées. Nous notons, aussi, pour un bruit de bavardage dans les trois intensités de SNR a un apport meilleur sur le taux de reconnaissance, 1,23% par rapport au bruit de la rue et de 10,06% par rapport au bruit rose. Ce faible écart s'explique entre le bruit résultant de bavardage et celui de la rue, de faite que les deux se ressemblent, ils contiennent la parole utilisé pour ces tests. Tandis que, le taux de mots du système basé sur le modèle GMM-HMM monophone présente 92,2% en état propre et 37,17% taux moyenne en état bruité. Dans les environnements très bruité (SNR $\leq$0), nous observons que le système triphone basé sur DNN-HMM fonctionne mieux que l'autre système basés sur le modèle GMM-HMM. Par exemple, dans le cas du bruit rose pour un SNR de 0 dB, le taux moyenne de mots reconnu pour le modèle triphone vaut 39,2% vs 20,6% pour le modèle mono. C.-à-d., soit une augmentation relative du taux WRR de 3,6%. Cet écart s'explique par l'apprentissage triphone d'ajouter des dépendances temporelles aux signaux et augmente ainsi leur complexité temporelle, ainsi que la stratégie itérative d'entraînement du modèle triphone est utilisé les alignements issus du modèle monophone. De plus, comme prévu, on voit qu'il y a une baisse de la performance lorsque le SNR décrois aussi, nous remarquons également que les performances dans le cas du bruit rose sont nettement moins bonnes que pour les autres types de bruit (bavardage et rue).

Donc, les performances des modèles bruité et non-bruité sont semblables pour des niveaux de bruits bas (SNR haut plus de 5). En revanche, lorsque le bruit commence à être perceptible (SNR $\leq$ 0), les performances du modèle de référence se dégrade considérablement alors que le modèle bruité conserve des performances honorables. De manière générale, le modèle bruité possède des performances nettement supérieures au modèle non-bruité.

b)-Test 2: Effet du modèle acoustique

D'après les résultats présentés dans le tableau 4.17, le modèle acoustique basé sur le système hybride DNN-HMM triphone est meilleur que celui qu'est basé sur le système hybride GMM-HMM monophone. La principale raison de la mauvaise performance de ces modèles peut être attribuée au manque de données d'entraînement suffisantes à cet effet. Cela peut être expliqué de fait que le modèle monophone fais sur un ensemble initial de donnés, on l'utilise pour aligner les données et entraîner un modèle triphone. Ainsi que, le modèle basé sur DNN intègre des méthodes d'extraction de phonèmes plus complexes que le modèle de monophone HMM singulier qui se compare sur une base de phonème unique. L'apprentissage triphone d'ajouter des dépendances temporelles aux signaux et augmente ainsi leur complexité temporelle, ainsi que la stratégie itérative d'entraînement du modèle triphone qui est utilisé les alignements issus du modèle monophone. Alors, les meilleurs taux de reconnaissance sont obtenus pour la modélisation par DNN-HMM triphone dans des conditions bruitées, en raison de sa dépendance au contexte. Par contre, les WRRs obtenus en utilisant le modèle HMM-GMM monophone sont relativement faibles en raison de sa variation insuffisante des phonèmes par rapport au contexte phonétique gauche et droit [99].
Dans la plupart des systèmes de reconnaissance de la parole actuels, les HMM basiques utilisés sont associés à des phonèmes (phone). Aussi, pour tenir compte de la variabilité de prononciation d'un phonème, un HMM est construit pour un phonème donné, associé à des contextes particuliers gauche et droit. Un contexte gauche d'un phonème est le phonème qui précède ce phonème et un contexte droit est le phonème qui succède à ce phonème. Ce triplet contexte gauche, phonème et contexte droit est appelé triphone ou phonème en contexte. Pour affiner la modélisation d'un phonème en contexte, la position de ce phonème dans un mot (début, milieu, fin ou phonème isolé) est parfois prise en compte[99]. En effet, le nombre de modèles augmente exponentiellement avec la taille du contexte. Par exemple, avec 24 unités phonétiques de base dans notre cas, il faut théoriquement estimer $24^3 = 13824$ modèles acoustiques si l'on veut modéliser des triphones. Donc, l'utilisation des modèles phonétiques dépendants du contexte (triphones) peut nécessiter des quantités excessives de données, par conséquent, prend plus de temps de calcul dans les deux étapes d'apprentissage et de la reconnaissance.
Dans notre cas, la plus grande amélioration peut être observée lors du passage de modèle monophone aux modèle triphone comme unité utilisée dans l'apprentissage avec les données bruitées. Le meilleur résultat pour le modèle acoustique formé à l'aide de GMM-HMM est obtenu pour le bruit bavardage. Donc, on peut conclure dans ce cas, que les deux modèles monophone et de triphone sont suffisants pour explorer les données (généralement, les modèles acoustiques monophonie n'ont pas besoin autant de données pour former leurs paramètres), cela est dû au fait que nous avons construit des modèles robustes au bruit avec l'apprentissage conditionnelle multiple à l'aide de la boite à outil Kaldi. Cependant, le système proposé est également moins robuste au bruit additif de type rose. De plus, d'autre méthodes et solutions peut être appliqué, en se basant sur les papiers [141], [142], [143] et [144] qui peut également améliorer les WRR considérablement dans nos travaux futurs.

c)-Test 3: L'effet du nombre des couches cachées

La boite à outil Kaldi fournit plusieurs variables paramétriques lors de l'entraînement du réseaux de neurones profond DNN, y compris d'architecture de réseaux, le nombre de couches cachés, la

taille ou nombre des unités cachés par couche, le nombre d'époques (epochs), la fonction d'activation utilisée, etc. Pour notre cas, afin d'évaluer l'effet du nombre de couches cachés sur la performance de notre système SRAP4 surtout dans des conditions bruitées, dans ce test sous différents niveaux du bruit additif de type rose, nous avons implémenté le modèle acoustique DNN en variant le nombre de couches cachées. En se basant sur les travaux des fondateurs de Kaldi : Karel Vesely et Daniel Povey, principalement le modèle **Dan's DNN (nnet2)**[6] de Daniel[145] car il est plus flexible pour l'entraînement, il prend en charge l'utilisation de plusieurs GPU, ou de plusieurs processeurs CPU, nous formons le modèle triphone DNN-HMM. Ce modèle exploite les réseaux de neurones de type DNN composé de trois couches cachées qui varie entre {1,2 et 3} à 512 unités dans chaque couche cachée respectivement, avec une fonction d'activité de type tangente hyperbolique tanh car elle présente une performance supérieure dans les travaux cité dans la littérature, et on applique la fonction *softmax* comme une couche de sortie. Le modèle a été appris sur 20 époques avec un taux d'apprentissage qui varie entre 0,04 et 0,004. Le tableau 4.18 résume les paramètres d'entraînement du DNN utilisé. Les résultats des trois tests sont présentés consécutivement dans le tableau 4.19 et ils sont exposés dans la figure 4.25.

Table 4.18: Paramétrage de d'entraînement de notre système DNN basé sur Kaldi.

Paramètre	**Valeur**
Code de base de DNN	***"nnet2" (Dan's DNN)***
Script	`train_tanh_fast.sh`
Fonction d'activité	`tanh:` dans les couches cachées `Softmax:` dans couche de sortie
Nombre de couches cachées	`num_hidden_layers =1,2 et 3`
Nombre d'epochs	`num_epochs=20`
Nombre d'epochs après l'arrêt de la réduction	`num_epochs_extra=5`
Taille des couches cachées	`hidden_layer_dim= 512`
Paramètres acoustiques	`MFCC à 13 dimensions`
Taux d'apprentissage (varie entre 0,04 et 0,004)	`initial_learning_rate=0.04` `final_learning_rate=0.004`

Table 4.19: L'effet du nombre des couches DNN sur le taux de reconnaissance WRR.

Nombre des couches cachées	base propre	Bruit rose			
		-5 dB	0 dB	5 dB	**Average (SNRs)**
1	94.4	18.1	35.7	64.7	39.5
2	95.6	20.6	41	66.6	42.73
3	97.1	21.4	39.2	70.8	43.8

Dans cette expérience, nous avons testé l'effet de faire varier le nombre de couches cachées du modèle hybride DNN-HMM entre (1, 2 et 3). D'après les résultats présenté dans le tableau 4.19 et figure 4.25 respectivement, qui présente le nombre de couches cachées de DNN en fonction du taux

[6] *https* : *//kaldi* − *asr.org/doc/dnn*2.*html*

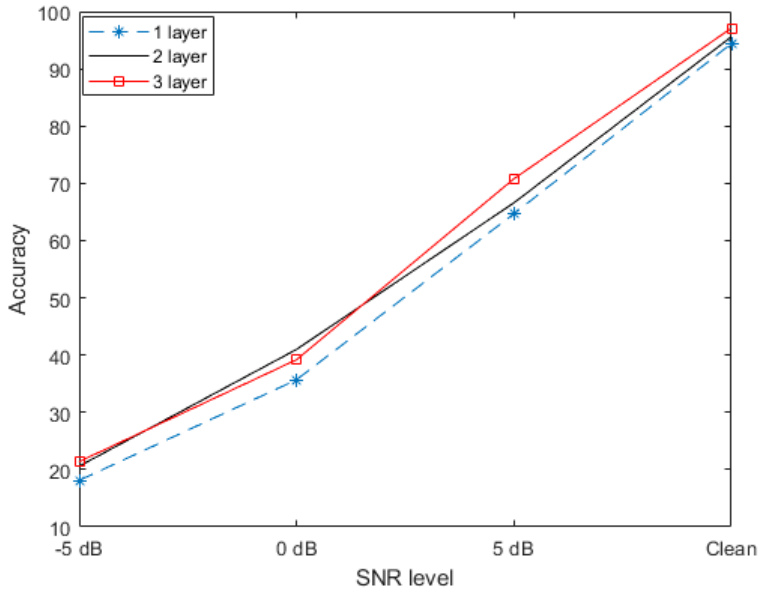

Figure 4.25: L'évaluation de l'effet du nombre des couches DNN sur le taux de reconnaissance sous le bruit additif rose dans les conditions de test.

de reconnaissance de mots dans différents niveaux du bruit rose. La meilleure précision dans des conditions de test propres est de 97,1%, il a été obtenu pour 3 couches cachées avec 512 unités, alors que dans les mêmes configurations DNN on a obtenu 43,8% de taux moyenne de reconnaissance dans les conditions de test de différents niveaux du bruit rose.
En basant de ces résultats, on peut conclure comme observations préliminaires que l'augmentation du nombre de couches cachées affecte proportionnellement les performances du système, de sorte que plus le nombre de couches cachées augmente, plus que le taux de reconnaissance de mots augmente aussi, mais il n'est pas significativement lié au type et le niveau du bruit. Cependant, ce résultat nécessite une étude plus approfondie en augmentant le nombre de tests.

4.4.5 Développement d'un système de RAP sous HTK basé sur HMM et le corpus "NASCIW"

4.4.5.1 Description du système « SRAP5 »

Dans cette section, nous décrivons comment créer et développer un système de reconnaissance de la parole arabe dans des conditions bruitées fondés sur les modèles de Markov cachés à partir de la plateforme HTK (Hidden Markov Model Toolkit) et évaluer sur la base de données NASCIW, nous l'avons nommé « SRAP5 ». Comme nous avons déjà décrit dans la section 2.3, la base de données NASCIW conçue pour évaluer les performances des systèmes SRAP5 dans différentes conditions de bruit, est utilisée pour sélectionner des paramètres acoustiques pertinents dans deux environnements: bruité et non bruité. Aussi, Nous avons déjà décrit la boite HTK dans la section 8.3 de chapitre 2, nous avons aussi présenté les différentes étapes du système de reconnaissance de la parole fondé sur les HMM sur la figure 4.16. Le système de référence basé sur les modèles HMM est implémenté à partir de la plateforme HTK. Toutes les étapes du système comme l'analyse acoustique, l'apprentissage des modèles HMM, la reconnaissance des séquences des mots inconnus ainsi que le résultat d'évaluation

sur un ensemble de séquences de test sont effectuées en utilisant la version 3.4.1 de la librairie HTK. Ainsi, un certain nombre de choix sont faits sur ce système, comme le nombre d'états des modèles, le type de densités de probabilité d'émission associées aux états et l'espace de représentation du signal par des coefficients acoustiques.
Dans la phase d'apprentissage, tout signal d'une séquence de mots de l'ensemble d'apprentissage est transformé en une séquence de vecteurs acoustiques pour qu'elle soit utilisée en tant que séquence d'observations d'entrée dans la modélisation markovienne des mots. L'ensemble des paramètres constituant chaque vecteur acoustique est constitué de 13 coefficients cepstraux en échelle de fréquence Mel MFCC (excepté le coefficient 0), le logarithme de l'énergie, leurs coefficients delta et delta-delta. Cet ensemble de 39 coefficients MFCC est calculé sur chaque trame d'analyse de 25ms avec un chevauchement de 10ms (outil ***HCopy*** de la librairie HTK).
Afin de concevoir notre système, on se base sur des 20 mots du corpus NASCIW. On commence par définir les ressources nécessaires dont on a besoin par la suite. Pour développer le système SRAP5 (presque la même façon comme le deux systèmes précédents SRAP3 et SRAP4) les principales étapes de construction sont les suivantes:

- Création de la base de données d'apprentissage (fichier ****.wav*** du corpus NASCIW): chaque élément du vocabulaire est enregistré plusieurs fois, et étiqueté avec le mot correspondant.
- Création de fichiers de transcription pour les données d'entraînement.
- Construire un dictionnaire : Le modèle de langage, appelé aussi lexique ou dictionnaire, qui décrit l'enchainement des mots.
- Construire la grammaire des tâches : la grammaire de la base de données NASCIW, composée d'un vocabulaire de 20 mots arabes qui définit le dictionnaire ou la grammaire (même que dans la figure 4.20).
- Extraction de fonctionnalités à partir des données d'entraînement (Analyse acoustique) : Extraction des coefficients MFCC sous HTK ; Après qu'on a défini le dictionnaire, la grammaire, on passe à l'extraction des coefficients MFCC exploités par les modèles de Markov cachés. Le fichier de configuration, appelé dans notre cas config (Figure 4.26), permet de définir les paramètres indispensables pour la phase de l'analyse acoustique. Ces coefficients sont extraits des fichiers *.**wav** et sur des fenêtres de 25ms grâce à l'outil ***hcopy*** en se servant du fichier de configuration comme paramètre d'entrée. En utilisant la commande DOS suivante: ***hcopy -A -D -C hcopy.conf -S hcopy.scp.*** Les fichiers des vecteurs acoustiques (****.mfcc***) obtenus sont utilisés à la fois dans la phase d'apprentissage et de reconnaissance du système.
- Définition des modèles : Définir le prototype de HMM pour chaque élément du vocabulaire.
- Entraînement des modèles : initialiser chaque HMM et l'entraîner avec les données d'entraînement.
- évaluation des identificateurs par rapport aux données de test ;
- Analyse des résultats de la reconnaissance

Après l'analyse acoustique, chaque mot est modélisé par un HMM représentant entièrement un mot. Nous avons créé un fichier de définition HMM (prototype) pour chaque phonème arabe. Initialement, nous avons utilisé un HMM à 3 états d'émission avec un mélange gaussien pour tous les phonèmes. Le HMM à 3 états d'émission est représenté par 5 états dans l'outil HTK (un état d'entrée, 3 états d'émission et un état final). Le vecteur de caractéristiques utilisé était MFCC_0 avec 13

config.hcopy

```
#  Example of an acoustical analysis configuration file NASCIW DATA
# Configuration file for hcopy command
# 12 mel frequency cep coeffs + cep(0); channel normalization done.
# Cepstral liftering is used to achieve quefrency weighted cep coeffs.
# Delta cep coeffs: window range is [-40msec ... 40msec].
# frame shift = 10ms;  window duration = 25ms.
# Sampling frequency = 16000Hz.
#SOURCEFORMAT= NASCIW   # Gives the format of the speech files
SOURCEFORMAT =WAV       # Gives the format of the speech files
SOURCERATE = 625        # 1 / sample-rate(16000)
TARGETKIND = MFCC_0_D_A # 0: power, D: delta, A: accel;
    # Unit = 0.1 micro-second :
WINDOWSIZE = 250000.0   # window-width [25ms]
TARGETRATE = 100000.0   # frame-rate [10ms]
USEHAMMING = T          # use hamming-window
ZMEANSOURCE = T
PREEMCOEF = 0.97       # Pre-emphasis coefficient
NUMCHANS = 26          # size of filterbank channels
CEPLIFTER = 22         # Length of cepstral liftering
NUMCEPS = 12           # Number of MFCC (here MFCC:12d (+ power:1d)*3)
ENORMALIZE = T
NATURALWRITEORDER = T
DELTAWINDOW = 4
SAVECOMPRESSED = T
SAVEWITHCRC = T
    # The End
```

Figure 4.26: Fichier de configuration pour la phase de l'analyse acoustique.

valeurs caractéristiques pour chaque trame. La figure 4.27 montre un fichier prototype HMM simple de 3 états d'émission. La fonction du prototype est de décrire la forme et la topologie du HMM. Les nombres initiaux représentant les probabilités d'état de transition utilisés dans la définition ne sont pas importants car ils seront mis à jour par le processus d'apprentissage. Par conséquent, la taille du vecteur (***VecSize***) et le type de paramètre (MFCC) devraient être spécifiés et le nombre d'états doit être choisi (***NumStates***). Les transitions permises entre les états devraient être indiquées en mettant des valeurs différentes de zéro dans les éléments correspondants à la matrice de transition (***TransP***) et zéros ailleurs. La somme de chaque ligne de la matrice de transition doit être égale à 1, sauf la dernière qui devrait être égale 0. Toutes les valeurs moyennes peuvent être zéro mais les variances diagonales devraient être positifs et les matrices de covariance devraient avoir des éléments diagonaux positifs.

Les modèles HMM des phonèmes sont d'abord initialisés à l'aide de l'algorithme de Baum Welch. L'outil HTK prend en charge deux types d'initialisation. Le premier type est appliqué lorsque les limites des phonèmes sont connues. La seconde est appliquée lorsque les frontières des phonèmes ne sont pas connues, ce qui est appelé initialisation plate. Comme les limites des phonèmes de nos données sont connues, le premier type d'initialisation est appliqué à l'aide de la commande ***HInit*** de HTK. Ensuite, ***HRest*** est utilisé pour réestimer les paramètres du modèle HMM sur les autres enregistrements de corpus d'apprentissage, c.-à-d., Pour chaque mot, des HMM seront générés. Alors, les données d'apprentissage sont prises et segmentées uniformément, et pour chaque modèle avec les données correspondantes, les moyennes et la variance ont été calculées.

Durant la reconnaissance, tout signal d'une séquence de mots de l'ensemble de test est transformé en une séquence de vecteurs acoustiques (MFCC pour notre cas) lequel sera transcrit en une séquence de mots. L'algorithme de décodage de Viterbi (commande HVite dans HTK) a été utilisé pour trouver le meilleur mot correspondant dans le dictionnaire. La grammaire accepte toute séquence constituée de n'importe quelle combinaison de mots, débutée et terminée par un silence avec la possibilité de pause entre les mots. Enfin, une analyse des résultats a été effectuée (à l'aide de la commande ***HResults***) en observant le taux de reconnaissance, le rapport entre le nombre de mots reconnus et

le nombre total de mots dans le test. Nous présentons dans le section suivante les résultats obtenus d'exécution du HTK avec NASCIW.

initial_0_D_A

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "initial_0_D_A"
<BeginHMM>
<NumStates> 5
<State> 2
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
1.0 1.0 1.0 1.0
<State> 3
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
1.0 1.0 1.0 1.0
<State> 4
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
1.0 1.0 1.0 1.0
<TransP> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Figure 4.27: Fichier de Prototype d'un HMM.

4.4.5.2 Evaluations et Analyse des résultats

Nous avons effectué une série d'expériences. Chaque expérience a ses propre configurations et conditions. Nous testons, dans un premier temps, les performances du système pour différents ensembles de fichiers d'entraînement et de test dans différents types de bruits. Puis nous évaluions la variation du nombre d'états HMM sur la performance.

a)-Expérience 1: performance du système SRAP5 dans différents conditions
Reconnaissance de la parole arabe à l'aide de HMM sur une base de données bruité : Comme nous avons détaillé dans la section 2.3 du présent chapitre, la base de données bruité NASCIW a été utilisé. Dans ce cas, ce corpus de travail a été divisé en deux parties. L'un est pour l'apprentissage se compose de 40 locuteurs et le second est pour le test du système qui est composé de 10 locuteurs. Les bruits artificiels ont été ajoutés à la base de données propre des échantillons de 20 mots pour obtenir une base de données bruité à 10 dB, 15 dB, 20 dB pour la phase d'apprentissage et à -5 dB, 0 dB, 5 dB pour la phase de test, pour les trois types de bruits.
La figure 4.28 montre le résultat d'un exemple de test en utilisant l'outil ***HResults*** qui compare le module de reconnaissance "Test_Babble5.mlf" avec la transcription de mot correcte "ASCIW_data.mlf" sous bruit de babillage pour un SNR = 5 dB. La ligne appelée (SENT) donne le taux de reconnaissance de la phrase (de % Corr =81.20), la ligne surnommée (WORD) donne le taux de reconnaissance des mots (% Corr = 81.20). Dans ce travail le taux de reconnaissance dans les

deux cas (phrase et mot) est le même, parce que notre système de reconnaissance est orienté vers une tâche de reconnaissance de la parole isolée, ainsi la grammaire ne permet pas la reconnaissance « phrases » qu'avec un seul mot (en dehors des silences), donc seulement la première ligne (SENT) doit être considéré ici. H = 1624 donne le nombre de données de test correctement reconnu, S = 376 est le nombre d'erreurs de substitution et N = 2000 est le nombre total de données de test comme suit:
10 locuteurs × 20 mots × 10 répétition × 1 type de bruit (***babble***) × 1 niveaux de SNR (5 dB) =2000

```
====================== HTK Results Analysis =======================
  Date: Thu Oct 10 22:34:06 2019
  Ref : ASCIW_data.mlf
  Rec : Test_Babble5.mlf
------------------------ Overall Results ----------------------
SENT: %Correct=81.20 [H=1624, S=376, N=2000]
WORD: %Corr=81.20, Acc=81.20 [H=1624, D=0, S=376, I=0, N=2000]
====================================================================
```

Figure 4.28: Performance du système en utilisant l'outil HResults de HTK sous bruit ***Babble à 5 dB***

Le tableau 4.20 sous-dessous, représente un résumé des résultats obtenus des performances du système SRAP5 dans les différentes conditions de test. Toutes les tests que nous avons réalisés sont effectuées en mode indépendant du texte. On a fait 10 tests au total, cela a été fait manuellement en changent la base de test de chaque expérience.

Table 4.20: Taux de reconnaissance de mots (WRR%) obtenus pour différentes conditions de test pour SRAP5.

Types de bruit utilisés et le niveaux de SNR		Modèle Acoustique HMM Sous HTK
	Clean	**94.6**
Le bruit bavardage (Babble noise)	-5 dB	25.2
	0 dB	44
	5 dB	**81.2**
Bruit de la rue (Street noise)	-5 dB	22
	0 dB	39.6
	5 dB	74.2
Bruit rose (Pink noise)	-5 dB	10
	0 dB	36
	5 dB	74.5
Moyenne (-5 at 5 dB)		**45.19**

Nous avons exécuté l'expérience du HTK pour différents ensembles de fichiers NASCIW de test. Pour explorer les résultats, l'indépendance de la prédiction en fonction du type du bruit et du SNR, le WRR a aussi été calculé pour chaque type de bruit et de différents SNR. Les résultats obtenus d'après le tableau 4.20 montre le changement les performances de SRAP5 avec le changement du type de bruit dans les données de test. Les performances sont meilleures dans le cas de test avec les données sans bruit de 94.6% et dans le cas des donnés bruités le taux moyen de reconnaissance est de

45.19%. Le meilleur taux dans le cas des données bruitées à trois niveaux de SNR a été marqué dans le cas du bruit ***babble*** avec un taux de 50.13%. Par contre, Les WRRs obtenus sont généralement très faible pour les systèmes mono-condition ***street*** et ***pink***. On a obtenu pour le bruit ***pink*** un taux moyen de 40,16 %. Donc, nous concluons que le type de bruit et la valeur du SNR influe sur la qualité de la performance dans le système HTK.

b)-Expérience 2: Effet de la variation du nombre d'états cachés dans le système SRAP5

Afin de tester l'effet du nombre d'états cachés dans cette expérience, nous avons modifié ce nombre dans les configurations du système basé sur HTK afin que nous puissions obtenir le nombre optimal du modèle. Le tableau 4.21 et la figure 4.29 représentent les résultats obtenus en utilisant un HMM sous l'effet du bruit ***babble*** à différents valeurs du SNR. Nous observons que le HMM à 7 états donne les meilleurs résultats.

Table 4.21: Taux de reconnaissance des mots (WRR %) en arabe en fonction du nombre d'états HMM dans le système HMM-HTK pour la base de données NASCIW dans le bruit **Babble**

Nombre d'états HMM	Base propre	Bruit « Babble »			
		-5 dB	0 dB	5 dB	moyenne (SNR)
3	89.7	16.4	24.5	65.6	43.53
5	92.1	20	36.4	70	49.5
7	**94.6**	**25.2**	**44**	**74.2**	**54.6**

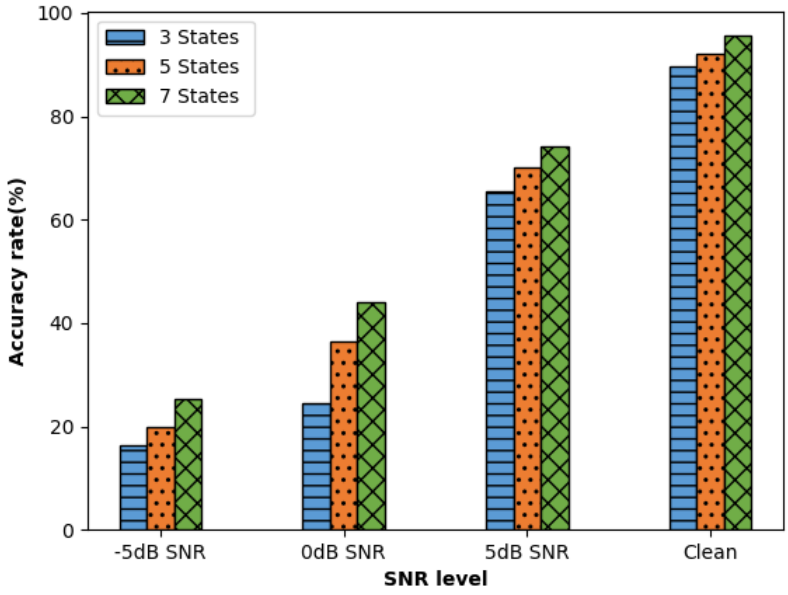

Figure 4.29: Evaluation du taux de reconnaissance de mots (WRR %) en fonction du nombre des états HMM et des différents niveaux de SNR en utilisant HTK sous le bruit ***Babble*** *.*

D'après la figure, il est observé que la performance du système augmente progressivement, au fur et à mesure que le nombre d'état de HMM augmente. Cela est principalement dû à un ajustement excessif des paramètres du modèle de mélange gaussien. Le meilleur paramètre sélectionnés pour la base de données NASCIW est de 7 états HMM avec une combinaison de 2 mélanges gaussiens. La performance du système dans ce cas est de 94,6 % dans les conditions propres et de 54,6 % dans les conditions bruitées.

4.4.6 Développement d'un système de RAP basé sur le modèle CNN et le corpus "NASCIW"

4.4.6.1 Développement du système « SRAP6 »

Dans cette expérience, nous avons utilisé l'ensemble de données ASCIW pour construire un modèle convolutif standard, une approche fondée sur des caractéristiques entraînées implicitement à l'aide des réseaux de neurones convolutifs (CNN). Nous l'appelons modèle SRAP6. Nous avons utilisé le corpus NASCIW avec différents types de bruit et niveaux de SNR pour créer un ensemble de données bruités. Nous divisons le corpus NASCIW en trois sous-ensembles: l'ensemble d'apprentissage, de validation et de test. Nous avons utilisé le script python "store_name_list.py" pour générer tous les fichiers wave dans deux fichiers texte, le premier appelé "validation_list.txt", qui contient une liste de fichiers qui devraient être utilisés pour valider les résultats pendant la phase d'entraînement (c'est la moitié des fichiers utilisée dans l'apprentissage). Le deuxième fichier nommé « testing_list.txt » contient les noms des fichiers utilisés que pour mesurer les résultats des modèles entraînés. Pour préparer les données pour l'apprentissage efficace du réseau de neurones convolutifs, nous convertissons les formes des ondes vocales en spectrogrammes auditifs log-bank. Il permet au modèle d'extraire et d'apprendre les caractéristiques nécessaires de l'image (spectrogramme). Les caractéristiques convoluées résultantes sont transmises à la couche de regroupement pour un traitement ultérieur. Nous le faisons en utilisant la fonction support "SpeechSpectrograms.m". Nous avons utilisé un réseau de neurones de trois couches de convolution avec 32 ***filtres*** et une ***fenêtre*** de taille 3×3 dans chaque couche avec un ***stride*** et même remplissage (***padding***). Les couches de regroupement sous-échantillonnent l'entrée à l'aide de l'opération ***MaxPooling***(). Le modèle est entraîné pour 10 époques (***epoche***), avec 90 itérations par époque. L'architecture du modèle final est fortement inspirée du projet MathWorks nommé « ***DeepLearningSpeechRecognitionExample*** » publié dans[7] [136]. Nous utilisons ces algorithmes pour détecter la présence de commandes vocales (20 mots arabes) grâce à des indices verbaux arabes. La figure 4.30 montre le graphique de la précision d'apprentissage et de la validation qui indique aussi une forte sur-optimisation, la fonction de perte d'apprentissage et de validation qui indique une sur-optimisation pour la modélisation CNN. La figure 4.30 représente également la perte générée à chaque époque.

Après l'apprentissage, nous conservons le modèle ayant les meilleures performances sur l'ensemble de validation et nous l'évaluons sur le corpus de test. Les sorties du classifieur sont évaluées en termes

[7] *https : //fr.mathworks.com/help/deeplearning/ug/deep − learning − speech − recognition.html*

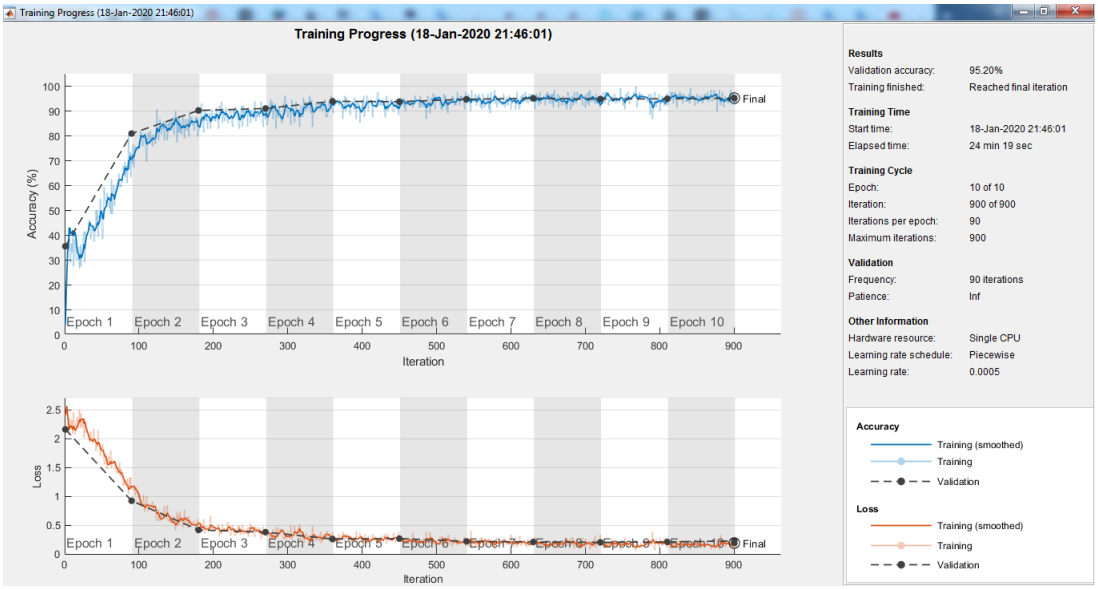

Figure 4.30: La courbe d'apprentissage pour les modèles CNN : évolution de la précision au cours de l'entrainement, de validation / évolution de la perte au cours de l'entrainement / de validation sur le corpus Test NASCIW dans les conditions propres.

de taux de bonne classification (***accuracy***). Les résultats obtenus sont présentés dans le tableau 4.22.

4.4.6.2 Evaluations et résultats

Dans notre travail, nous avons utilisé un réseau de CNN à trois couches qui donne une précision remarquable de 95,2%. Le choix des trois couches réside dans la petite taille de l'ensemble de données. Les caractéristiques ont été extraites des données à l'aide de l'algorithme MFCC. Les données du fichier « apprentissage » ont été utilisées pour entraîner le réseau CNN, de même que le modèle testé avec les données du fichier « tests » qui contiens les données de test. Nous avons testé le SRAP6 avec plusieurs conditions de test, les performances des tests du système sont résumées dans le tableau 4.22 suivant.

Table 4.22: évaluation de la prédiction des performances du SRAP6 obtenus pour différentes conditions de test.

Types de bruit utilisés et le niveaux de SNR		CNN Matlab Deep Learning Toolbox™
Clean		**95.20**
Le bruit bavardage (Babble noise)	-5 dB	**30.17**
	0 dB	**50.2**
	5 dB	78.23
Bruit de la rue (Street noise)	-5 dB	26.82
	0 dB	46.74
	5 dB	**79.25**
Bruit rose (Pink noise)	-5 dB	8.33
	0 dB	30.43
	5 dB	65.32
Moyenne (-5 at 5 dB)		**46.17**

D'après la figure 4.28, les résultats expérimentaux montrent une amélioration de la précision avec l'augmentation des époques ainsi l'entraînement avec les données bruitées augmente les performances. En outre, l'entraînement efficace est important car il fournit la solution pour une sortie de prédiction précise. La robustesse d'un réseau à de nouvelles données est importante pour un bon fonctionnement en conditions réelles. Nous avons donc effectué ces expériences pour déterminer la robustesse de notre réseau face à peu de données d'apprentissage. D'après le tableau 4.22, nous remarquons que le meilleur résultat dans le cas bruité est obtenu pour le bruit de la rue avec une précision de 79.25 % et de 95.20 % dans le cas où le test avec les données propres. La précision moyenne de l'approche proposée diminue également dans le cas de bruits ***pink***. Le taux moyen de reconnaissance dans les conditions bruitées à 5 dB est de 74.27%, c'est un taux élevé, étant donné que les données d'entraînement sont bruitées et par rapport à certaines approches publiées antérieurement présentés dans le tableau 4.23.

Table 4.23: Comparaison de notre modèle proposé avec certaines approches publiées antérieurement dans des conditions similaires.

Travail	Corpus	Techniques	Taux de Recogntion (%)
Modèle de Alalshekmubarak et al [137]	ASCIW (9992 unités)	MFCCs-ESNEKM RASTA - ESNEKM	99.69 % (Sans bruit) 65.48% (Présence de bruit ***babble*** à 10 dB)
Modèle de Abdelmaksoud et al [146]	ASCIW (29972 unités)	GFCC - CNN	99.77 % (Présence un peu de bruit)
Modèle de J Anwar Qadir et al [147]	Kurdish digits (11928 unités)	CNN-Multilayer perceptron	98.52% (condition propre : Dépendant du locuteur et indépendant)
Modèle de Kh-Koummich Fatima et al [148]	10 commandes arabe/français (300 unités)	TDNN	-Test mots arabe 92.67% en bruit gaussien AWGN à 35 dB et 97,6% en milieu neutre. -Test mots français 77.67% en bruit gaussien AWGN à 35 dB et 87% en milieu neutre.
Modèle de Bilal Dendani et al [149]	ASCIW (9992 unités)	Deep AutoEncoder (DAE)- HMMs	64.44% (Effets du codage)
Modèle de Naima Zerari et al [150]	-10 Spoken Arabic digit dataset (8800 tokens) -10 Spoken Arabic command dataset TV (10000 tokens)	MFCC-BiLSTM Multi-Layer Perceptron	Précision supérieure à 96% Pour les deux corpus (condition normal)
Modèle de Abdulaziz S Ba Wazir et al [151]	Arabic digits 0 through 9 (1040 utterances)	MFCC - LSTM	précision de reconnaissance globale de 69 % (codage audio Opus débruité par Audaciy)
Modèle de RA Rajagede et al [152]	10 lettres arabes (400 utterances)	MFCC - CNN	précision allant jusqu'à 83,00% (environnement assez calme)
Notre Modèle proposé	NASCIW (ASCIW bruité) (96000 utterances)	MFCC - CNN	-95.20% (donnés propres) **-74.27%** (moyenne **Acc** dans les bruits à 5 dB)

Dans l'implémentation du système SRAP6, nous avons mis en œuvre un modèle robuste de reconnaissance automatique de la parole arabe indépendant du locuteur basé sur un réseau de neurones convolutifs pour reconnaître les mots arabes. Nous essayons d'améliorer les performances de notre système par le choix de paramètres de réseau CNN pour la préparation d'un bon entraînement et aussi pour réduire le temps de classification. Nous avons augmenté les données du corpus ASCIW en injectons les types de bruits à différents SNR dans les données d'apprentissage afin de rende le

système plus robuste. Le modèle proposé a été évalué sur le corpus NASCIW. Il a été démontré que le MFCC avec l'entraînement du modèle avec le réseau CNN offrait la meilleure précision de reconnaissance au milieu bruité par rapport aux autres techniques utilisé dans les travaux similaires. Le modèle proposé a montré des résultats efficaces et des performances améliorées en termes de précision (***Accuracy***) par rapport aux approches existantes telles que présentées dans le tableau 4.23. Pour améliorer les performances de notre système, On peut penser d'essayer d'autres techniques de prétraitement des données, d'entraînement du modèle, d'initialisation du poids ou généralement de changer les paramètres pour chacun de réseau CNN (nombre de couches, de neurones, dimension des filtres de convolution, etc.) ou d'appliquer d'autres types de réseaux de neurones et de créer un ensemble de données très large plus important pour améliorer la précision est également nécessaire.

4.4.7 Développement d'un système de RAP basé sur le Modèle GMM-HMM et le corpus "DARIJA_MO"

4.4.7.1 Description du système « SRAP7 »

Dans cette expérience, nous avons réalisé un système de la reconnaissance automatique de dialecte marocain arabe sous différents bruits additifs en utilisant l'outil open source PocketSphinx basé sur le modèle GMM-HMM. Nous l'appelons dans ce rapport du thèse SRAP7. Nous avons suivi les mêmes étapes que nous avons suivi lors de la réalisation du système SRAP3.Ce système que nous avons mis en place satisfait certaines conditions qui sont:

- Parole de dialecte arabe marocain : Corpus DARIJA_Mo
- Conditions de plusieurs types de bruit
- Indépendance des locuteurs (multi-locuteurs)
- Vocabulaire de petit taille

Chacun de ces phases constitue, lui-même, un challenge à relever. Par conséquent, pour réaliser le système SRAP7, dans la boite PocketSphinx, nécessite plusieurs d'opérations complexes : la collection et l'organisation de la base de données en corpus d'apprentissage et test, la création du modèle acoustique et du dictionnaire, l'écriture de plusieurs scripts, la manipulation et la gestion d'un nombre important de lignes de commandes ainsi qu'une multitude de fichiers sonores, de données textuelles et binaires.
Comme nous l'avons détaillé dans la section 4.3 du présent chapitre, un système de reconnaissance de la parole nécessite généralement 3 phases de traitements à savoir : la préparation du système ou sa paramétrisation, l'apprentissage puis le décodage et l'analyse des résultats. Chacune de ces phases est-elle même partagée en plusieurs étapes de traitements. La figure 4.31 résume les différentes étapes suivies dans l'élaboration des modèles monophones pour les deux phases d'apprentissage et de reconnaissance.

D'abord, nous préparons un corpus de petite taille des dix salutations les plus célèbres au dialecte marocain dans les conversations téléphoniques. Ce corpus de parole enregistré par 60 locuteurs (30

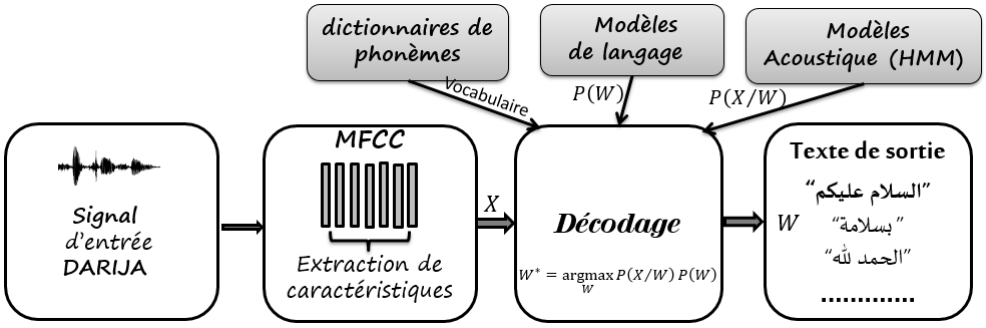

Figure 4.31: Architecture du système de reconnaissance automatique de la dialectale arabe Marocaine « SRAP7 ».

hommes et 30 femmes), prennent 1800 expressions dans lesquels chaque locuteur prononce chaque expressions trois fois, les expériences se faites dans des conditions réelles (bruitées). Puis, nous avons préparé, de la même façons, les fichiers du modèle acoustique, de grammaire, dictionnaire, fichier de transcription d'apprentissage et de test. La structure des fichiers répertoire principal du notre projet basé sur PocketSphinx nommé "DARIJA_MO" est présenté dans la section 4.2 (figure 4.2).
L'extraction des caractéristiques se fait avec les coefficients cepstraux dans l'échelle des Mels (MFCC), de dimension égale à 39, est constitué des 13 premiers coefficients MFCC augmentés de leurs dérivées première et seconde pour obtenir un vecteur de caractéristiques de 39 dimensions par trame. Nous avons utilisé pour cela des modèles acoustiques de phones indépendants du contexte (39 monophones). Pour la modélisation acoustique basée sur le monophone est effectuée par les modèles de Markov cachés (HMM) avec le modèle de mélange gaussien (GMM) pour extraire les vecteurs de caractéristiques acoustiques. Ces modèles sont des HMM gauche-droite à trois états avec des mélanges de Gaussiennes à 8 composantes, entraînés sur le corpus DARIJA_MO.
Enfin, nous avons évaluées notre système les données DARIJA_MO dans plusieurs conditions, nous allons donner les résultats dans la section suivante.

4.4.7.2 Evaluations et discussion des résultats

Dans l'étape d'évaluation du notre système. Nous avons testé le système en faisant varier les conditions réelles dans la base de données DARIJA_MO qu'on a présenté précédemment en se basant sur l'architecture du système présentée dans la section précédente. Toutes les expériences ont été effectuées sur le même ordinateur avec les configurations suivantes: processeur Intel® Core (^{TM}) i5-4310U @ 2 ,20 GHZ 2,20 GHz × 8Go de RAM, Windows 7 édition Intégral 64 bits. Pour la création des modèles acoustiques nécessite un ensemble de vecteurs caractéristiques calculés à partir des données audio d'apprentissage, un pour chaque enregistrement de ce corpus. Nous avons utilisé la technique de mélange du modèle de Markov cachés avec densités multigaussiennes (GMM - HMM) pour générer des modèles acoustiques de chaque mot. Nous avons appliqué l'outil ***sphinx_fe*** pour convertir des fichiers audio en fichiers acoustiques, après, nous les avons transformés en une séquence de vecteurs de caractéristiques comprenant les coefficients cepstraux Mel-Frequency (MFCC). L'apprentissage a été effectuée à l'aide de 1800 expressions de données de la base DARIJA_MO. Le premier ensemble d'expériences a été effectué par des données propres, puis a été testé en ajoutant certains types de bruit. De plus, nous avons évalué les performances de notre système lorsque nous varions le nombre

d'états HMM (3 et 5) et le nombre des composantes gaussiennes (allant de 4 à 64) dans les conditions les 4 types de bruit réel mentionnées précédemment (section 4.2). Toutes les modules acoustiques ont été créés avec des donnés propres et testées après avec des données dans différents conditions.
Nous avons classé les tests en quatre types comme suit :

- **Le premier test** a été fait avec des données propres.
- **Le deuxième test** a été réalisé avec des données enregistré dans des conditions réelles.
- **Le troisième test** c'est un test en direct c.-à-d. en utilisant un microphone dans un environnement spécifique.
- **Le quatrième test** : Était réalisé pour savoir l'effet des nombres d'états HMM et le nombre de gaussiennes sur la performance de notre système.
  Rappelons que l'évaluation de chaque expérience a été effectuée selon le taux de reconnaissance des mots (il s'appelle aussi la précision (Accuracy)) est symbolisé par ***(WRR)*** en anglais qu'on a défini par l'équation (4.4).

a)-Résultats du système de base

Nous avons aussi évalué les performances du modèle de reconnaissance sur quatre types de bruit réel pour différents SNR pour les tests dans les conditions du bruit réel et les autres dans les données presque propres ou dans des tests directs. Le tableau ci-dessous illustre les résultats qui sont obtenus à partir de la mise en œuvre du système SARAP7 proposé. Nous avons pris par défaut 8 densités gaussiennes et 3 états par HMM dans les trois premiers tests. Le tableau 4.24 montre les résultats obtenu dans ces cas.
Pour tester notre système avec un test spécifique on a utilisé cette commande dans le décodeur pocketsphinx :
pocketsphinx_continuous -infile "C:\ProjectSphinx\Darija\test\test_Bus.wav" -hmm "C:\ProjectSphinx\Darija\output\Darija_Mo.ci_cont" -dict "C:\ProjectSphinx\Darija\output\Darija_Mo.dic" -lm "C:\ProjectSphinx\Darija\output\Darija_Mo.lm.DMP"
Pour tester notre système directement par le microphone de notre machine on a utilisé cette commande dans le décodeur pocketsphinx :
pocketsphinx_continuous -inmic "yes" -hmm "C:\ProjectSphinx\ Darija \output\Darija_Mo -dict "C:\ProjectSphinx\ Darija\output\ Darija_Mo.dic" -lm "C:\ProjectSphinx\ Darija\output\ Darija_Mo.lm.DMP"

D'après le tableau 4.24 des résultats obtenus pour la reconnaissance automatique de la parole donnent des performances acceptables dans des conditions sans bruit, mais les performances se dé-

Table 4.24: Résultats des performances totales du système SRAP7 dans les trois tests.

Testes		**Résultats**			
		Expressions total	Correct	Erreur	WRR (%)
Test 1 : Dans les conditions Normal (SNR>15 dB)		800	786	14	98.25
Test 2: Dans les conditions bruyantes	Salle de photocopie (SNR 15dB)	800	548	252	68.5
	Cour de faculté (SNR 10dB)	800	397	403	49.62
	Café SNR 6 dB	800	301	499	37.62
	Autobus (SNR 2 dB)	800	145	655	18.12
Test 3 : tests en direct	Locuteur 1 : dans labo étude	50	27	23	54
	Locuteur 2 : Moi-même dans ma maison	50	39	11	78

gradent considérablement en présence de bruit.

b)-Test 4: l'effet du Nombre d'états par HMM et du nombre de gaussiennes

Pour tester l'influence de changement du nombre de distributions de probabilité gaussiennes sur les performances du système, ce dernier a été formé et testé pour différentes valeurs gaussiennes allant de 2 à 64 par un HMM de 3 états. Le tableau 4.25 et la figure 4.32 successivement présentent les résultats des expériences décrites ci-dessus sur les taux de reconnaissance pour la partie de test de notre base de données DARIJA_MO dans les conditions propres et bruités en variant le nombre de GMM.

Table 4.25: Taux de reconnaissance comparatifs de l'effet des conditions réels en fonction du nombre de gaussiennes pour 3 états par HMM.

Nombre de gaussiennes GMM	**Base de test Normal SNR>15**	Base de test bruité				
		Salle photoC 15 dB	Faculté 10 dB	Café 6 dB	Autobus 2 dB	**Moyenne (SNRs)**
2	95.5	62.38	44	32.5	15	**49.88**
4	96.62	65.63	46	34.75	17.13	**52.03**
8	**98.25**	**68.5**	**49.62**	**37.62**	**18.12**	**54.42**
16	97.12	67.5	47.5	39.25	14.66	**53.21**
32	97.38	67.38	48.63	36.38	18.25	**53.60**
64	97.5	67.12	46.63	39	19.75	**54**

Pour savoir l'effet du nombre d'états HMM en fonction des conditions des tests sur la performance de notre système au niveau des modèles acoustiques, nous varions ce nombre comme suite: le système a été testé avec 8 nombre de composantes GMM dans la première étape en utilisant un état par HMM,

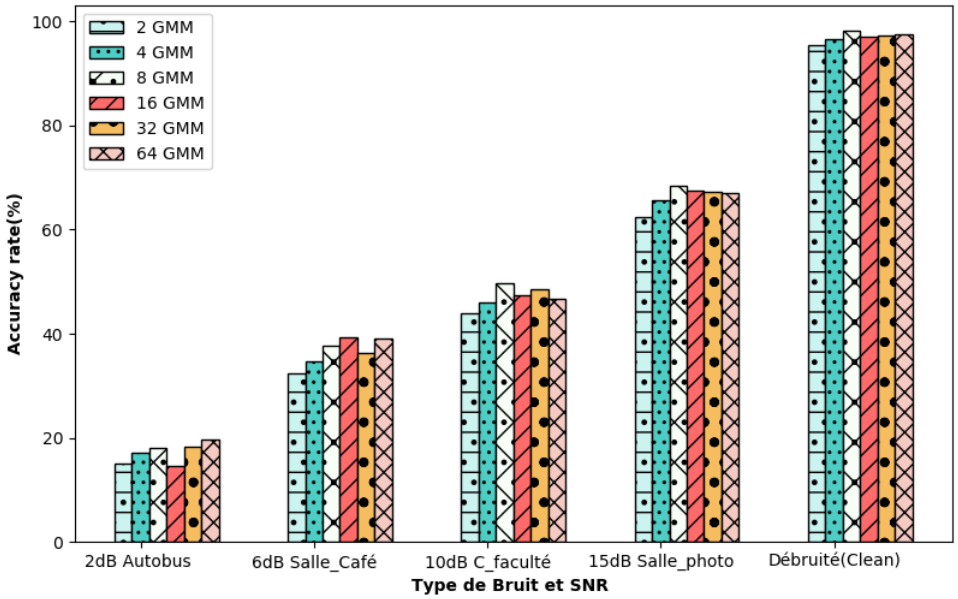

Figure 4.32: *Evolution du WRR en fonction du nombre de Gaussiennes par un HMM à 3 états.*

puis avec trois états par HMM et enfin avec cinq états par HMM à 8 GMM. Les résultats des trois tests sont présentés consécutivement dans le tableau 4.26 et ils sont illustrés dans le graphique de la figure 4.33.

Table 4.26: Taux de reconnaissance comparatifs de l'effet du nombre d 'états par HMM en fonction des conditions de test réels pour 8 GMM.

Nombre des états HMM	**Base de test Normal SNR>15**	**Base de test bruité**				
		Salle photoC 15 dB	Faculté 10 dB	Café 6 dB	Autobus 2 dB	Moyenne (SNR)
1	73.5	46.13	34.25	24.88	14.5	38.65
3	**98.25**	**68.5**	**49.62**	**37.62**	**18.12**	**54.62**
5	87.62	58.88	47.38	30.5	16.75	48.23

c)-Interprétation et Discussion

A note préliminaire, les résultats obtenus en général d'après les tableaux 4.19, 4.20 et 4.21 et la figure 4.30 montrent une relation presque linéaire entre le taux de reconnaissance de mots (WRR) en fonction du niveau du bruit. Le taux diminue proportionnellement avec l'intensité du type du bruit, ce qui indique que le modèle acoustique change presque avec la même intensité que le niveau de bruit de l'ensemble de test, sans aucune dépendance du type de bruit. Cela signifie que n'importe quel modèle est capable d'apprendre les modèles de bruit indépendamment les uns des autres.
Le taux le plus bas pour le système est obtenu dans le bruit autobus et le bruit de salle de café par un taux très faible de 18,12% et 37,63% respectivement (tableau 4.19). Pendant que, pour le bruit produit dans une salle de photocopie n'affecte pas beaucoup sur la précision de reconnaissance

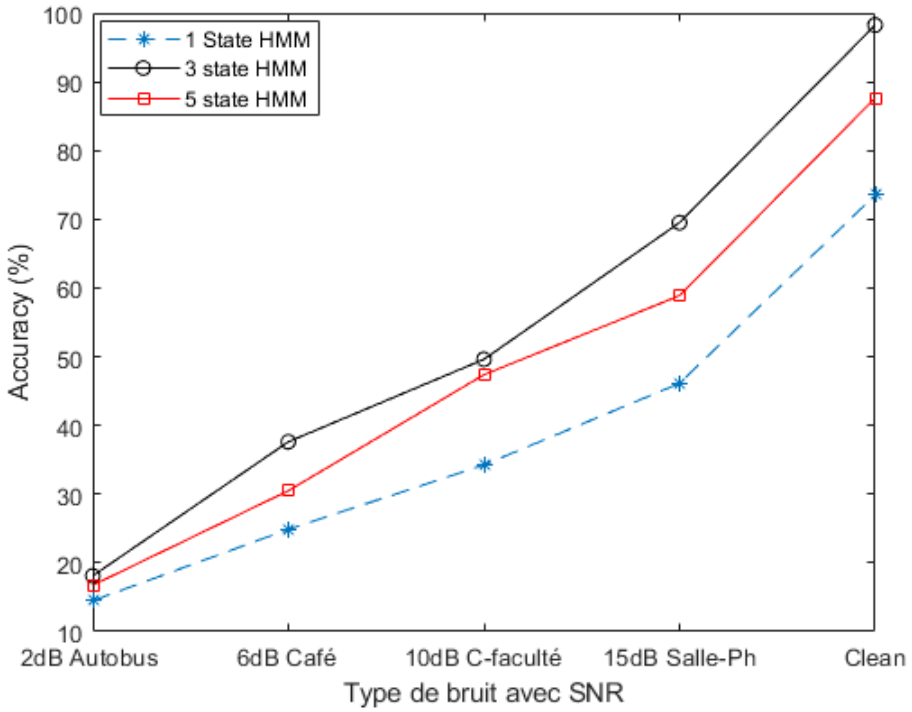

Figure 4.33: Evolution du (WRR %) en fonction du nombre d'états HMM et les conditions de test pour GMM=8.

comparativement avec les deux types précédents lorsque nous le testons directement avec le microphone, il a atteint dans ce cas 68,5% et 78% successivement. Les autres types de bruit (cour de la faculté, salle de laboratoire) ont l'impact presque similaire. Cela peut s'expliquer par le bavardage plus dans le bus et le café, où les mots s'affectent dans les modèles acoustiques comparativement avec les autres types (salle de photocopie, maison, cour de la faculté, laboratoire) ont des niveaux de bruit inférieurs. Dans les conditions propre (réduction du bruit naturel) le système de RAP de base atteint un taux WRR de 98,25%.

Dans le tableau 4.20 et la figure 4.30 (Test 4), nous pouvons observer aussi que, l'accroissement du nombre de gaussiennes fait augmenter le taux de reconnaissance, les meilleures valeurs de WRR obtenue sont pour 8 GMM et à 3 état par HMM. Dans le tableau 4.21 et la figure 4.31 (Test 4), nous extrayons que le changement du nombre d'états HMM affecte de manière significative sur les résultats obtenus. Par conséquent, les résultats idéals sont atteints pour le cas de 3 états par HMM.

Le travail présenté a été comparé aux travaux similaires existants. Dans le document [56]. Les auteurs ont présenté un système de reconnaissance vocale des chiffres Amazighs dans un milieu bruyant de l'intérieur d'une voiture. Ils ont utilisé le modèle de Markov caché pour modéliser les unités phonétiques correspondant aux mots extraits de la base d'apprentissage sous l'open source CMU Sphinx4. Les résultats expérimentaux pour 1800 des données présentent un taux de reconnaissance de 88,22% dans un environnement propre, et des taux de 59,26% et 33,83% en condition bruyante pour un SNR de 10 dB et de 20 dB, respectivement.

Dans le papier,[49] ils ont proposé un système de reconnaissance vocale pour de la langue Amazigh en se basant sur l'open source CMU Sphinx-4. Le système proposé comprend les étapes d'extraction des caractéristiques, la modélisation acoustique à l'aide de méthode HMM. La taille de la base de données pour ce travail est de 2970 mots donne 88% de précision.

Dans l'étude scientifique,[153], les auteurs ont construit un système d'identification du locuteur en dialecte marocain basant sur MFCC pour extraire les caractéristiques de la parole inconnue et les modèles de Markov cachés pour la classification .Le code a été développé avec le logiciel MATLAB, ils ont arrivé à identifier le locuteur de manière satisfaisante.
Pour notre travail d'implémentation du système SRAP7, en comparant avec les résultats des travaux connexes et similaires existants, la performance de notre système est très satisfaisant (98,12% WRR dans les conditions propres et d'une moyenne de 50,98% WRR dans les différentes conditions de bruit réel). Compte tenu, de la petite taille des données et qu'on a construit le modèle de langue et le modèle acoustique du dialecte arabe marocaine "DARIJA" comme une nouvelle langue. Donc, il est considéré parmi les premiers travaux traitant le dialecte marocain dans les conditions réelles utilisant la librairie PocketSphinx.

4.4.8 Etude comparative entre les différents modèles implémentés dans les milieux bruités

4.4.8.1 Comparaison des méthodes et outils open source utilisés

Les tableaux 4.27 et 4.28 représentent la comparaison des résultats obtenus par les systèmes SRAP3, SRAP4, SRAP5 et SRAP6 en termes du taux de reconnaissance de mots dans différents conditions bruitées ainsi de leurs durée d'exécution de l'entraînement et de décodage. Les meilleurs résultats sont édités en gras.

Table 4.27: Comparaison en termes de taux de reconnaissance de mots (WRR %) des quatre systémes de RAP dans toutes les conditions de test.

Système Méthode et outils	WRR dans les conditions Propres	WRR dans les conditions de bruit									
		Babble noise			**Street noise**			**Pink noise**			
		5dB	**0dB**	**-5 dB**	**5dB**	**0dB**	**-5 dB**	**5dB**	**0dB**	**-5 dB**	**Moyen**
SRAP3 : GMM-HMM sous Pocketsphnix	96.2	80.4	42.6	24.6	75.1	44.2	24.8	70.5	35.1	17.9	46.13
SRAP4 : DNN-HMM basé sur Kaldi	**97.1**	**81.4**	**46.7**	**33.5**	**76.1**	**49.8**	**32**	**70.8**	**39.2**	**21.4**	**50.1**
SRAP5 : HMM sous HTK	94.6	81.2	44	25.2	74.2	39.6	22	74.5	36	10	45.19
SRAP6 : CNN en Matlab	95.20	78.23	50.2	30.17	79.25	46.74	26.82	65.32	30.43	8.33	46.17

Table 4.28: Comparaison de la durée d'exécution entre les outils open source utilisé.

Système Open source En Single CPU	Temps d'exécution		Performances Dans les conditions propres	Performances Dans les conditions propres
	Entraînement	Décodage		
SRAP3 : Pocketsphinx	**1 h 04 m**	**9 m**	96.2	46.13
SRAP4 : Kaldi	2 h 22 m	14 m	**97.1**	**50.1**
SRAP5 : HTK	2 h 16 m	41 m	94.6	45.19
SRAP6 : Matlab	2 h 10 m	24 m	95.20	46.17

4.4.8.2 Analyse des résultats obtenus

Les résultats présentés dans le tableau 4.27 montrent d'une manière générale que les performances des systèmes de mots parlés en arabes se dégradent bien sûr en présence du bruit. En effet, le degré de dégradation varie d'un système à l'autre et selon le type de bruit additif. Le système SRAP4 basé sur Kaldi et le modèle hybride DNN-HMM est le meilleur en terme du taux WRR et de robustesse au bruit par rapports aux trois autres systèmes suivi par le système SRAP3 basé sur Pocketsphinx. Ce dernier est mieux que les deux autres systèmes SRAP5 basé sur HTK et SRAP6 basé sur Matlab dans toutes les conditions de test. Le système SRAP4 est plus performant avec le meilleur taux de reconnaissance 97,1 % en conditions propres et 50,1 % en moyenne en conditions bruyantes. Tandis que, le système SRAP3 présente 96,2 % dans les conditions de test avec les données propres et de 46,13% WRR en moyenne dans le test avec les conditions bruitées. De plus, notons que les systèmes SRAP5 et SRAP6 ont arrivé à 94,6 % et 90,6 % de WRR pour les données propres et à 45,18 % et 41,02 % en moyenne de WRR pour les données bruitées, respectivement.
Comme prévu, on constate une baisse de précision pour ces approches avec un SNR décroissant, nous remarquons également que les performances des systèmes utilisés dans les tests avec le bruit rose sont nettement pires que pour les autres types de bruit (babillage et rue).
Dans des environnements très bruités, nous observons que le système basé sur DNN-HMM fonctionne mieux que les autres systèmes basés sur des modèles. Par exemple, à 0 dB dans le bruit rose, la précision est de 39,2 contre 35,1 pour l'architecture GMM-HMM basée sur le CMU Sphinx. Cela est dû au fait que nous avons construit les modèles insonorisés avec une formation multi-conditionnelle à Kaldi. Tandis que, Pocketsphinx utilise des fonctionnalités robustes au bruit, la soustraction spectrale et le masquage temporel sont utilisés.
D'après le tableau 4.28, le temps passé à configurer, préparer, exécuter et optimiser les boîtes à outils était principalement rapide pour PocketSphinx, moins pour Kaldi, moins pour Matlab et HTK. La duration pour les système Pocketsphinx (SRAP3) prenait **1 heure et 13** minutes pour s'entraîner et décoder le modèle GMM-HMM. Tandis que, le système basé (SRAP4) sur Kaldi a pris 2 heures et 36 minutes pour s'entraîner et décoder le modèle triphone DNN-HMM. La composante en temps réel de l'entraînement HTK (SRAP5) a duré environ 2 heures 16 minutes, tandis que le décodage a pris environ 41 minutes. Pour le dernier système SRAP6 implémenté dans Matlab a duré 2 heures et 34 minutes au totale pour former, valider puis tester le modèle basé sur les réseaux CNN.
Le system SRAP4 basé sur Kaldi possède la meilleure précision globale par rapport aux autres systèmes, Kaldi surpasse PocketSphinx. Malgré cela, la profondeur des techniques avancées offertes par

la boîte à outils Kaldi suggère que les meilleures performances pourraient être obtenues si des méthodes plus avancées mais aussi plus gourmandes en calcul sont utilisées. Les systèmes PocketSphinx ASR ont tout de même réussi à obtenir de bonnes performances tout en maintenant un niveau de charge de calcul relativement faible. Par conséquent, il permet d'entraîner et décoder le modèle avec une durée plus moins que Kaldi. Le système SRAP5 mis en place par HTK est prend dans notre cas beaucoup de temps pour rivaliser avec la précision des performances atteinte par les deux outils précédents. De même, pour le système SRAP6 basé sur Matlab il prend aussi de temps.
En générale, nous pouvons déduire que la durée d'exécution des outils open source n'ont rien à voir avec la nature des données (propres ou très bruités) au moment de l'entraînement et de décodage, mais leur taille influe naturellement sur la vitesse et le temps d'exécution. Cependant, les principaux facteurs affectant le temps d'exécution sont les caractéristiques et les structures internes de chaque boite à outils (type de la méthode de classification : HMM, DNN…, nombres des paramètres, type de modèle acoustique monophone, triphone …, etc.) ainsi les configurations de la machine utilisé (Nombre et type de CPU ou GPU, RAM, etc.).

4.4.8.3 Conclusion

Quatre système basé sur quatre boîtes à outils open source ont été implémenté et comparé pour la langue Arabe en milieu bruité ; PocketSphinx, Kaldi, Matlab et HTK. Les systèmes de RAP ont été formés avec succès sur les quatre boîtes à outils pour le corpus arabe NASCIW. Kaldi a produit les systèmes les plus précis en général. Cependant, les systèmes basés sur PocketSphinx ont atteint une précision presque similaire. Les outils CMUSphinx offraient la convivialité la plus simple, la plus pratique, plus robuste au bruit et la plus rapide, donc, ils constituent la meilleure option pour la formation des systèmes de RAP. PocketSphinx a également réalisé les temps les plus efficaces pour l'entraînement et le décodage. Cependant, Kaldi n'est pas loin derrière. Alors que Kaldi nécessite plus de temps d'installation, l'inclusion d'exemples de scripts. De plus, Kaldi fournit un support prêt à l'emploi pour les techniques de pointe avancées telles que les DNN, DBN, Dan's Hybrid DNN (tri4-nnet), Karel's Hybrid DNN, les SGMM, etc. Kaldi est plus robuste au bruit par rapport aux autres. HTK prend du temps à utiliser et à configurer, obtient de mauvais résultats avec la configuration de base. De plus, il est très inefficace. Par conséquent, l'utilisation de HTK n'est pas recommandée. Matlab n'est pas un environnement spécialisé de développement et de la construction d'un système de reconnaissance vocale avec tous ces détails de la production de modèles acoustiques, de langages et de dictionnaires, mais récemment il a développé des méthodes et des algorithmes concernant l'apprentissage et utilisation de réseaux profonds en reconnaissance vocale. De plus, il est plus efficace dans le traitement des signaux vocales bruités et l'amélioration de la parole.
Traditionnellement, CMU Sphinx et Kaldi sont également très cités dans la littérature académique. Mais il existe d'autres outils open source qui peuvent être utilisés à l'avenir comme par exemple : Simon, Julius, Wav2Letter++, DeepSpeech, Tansorflow, OpenSeq2Seq, PyTorch-Kaldi, etc.

Conclusions et perspectives

Les travaux menés dans cette thèse s'intègrent dans le cadre des recherches sur les systèmes de reconnaissance automatique de la parole. L'étude est réalisé dans les environnements bruités et réels avec plusieurs outils et techniques. Le but est d'une part améliorer les performances et la robustesse des systèmes de RAP dans les conditions réelles, et d'autre part évaluer notre approche proposée d'apprentissage multi-styles avec les données bruitées. Cette problématique a été abordée spécialement pour la parole arabe standard et pour le dialecte marocain dans un milieu bruité.
Nous avons présenté dans les chapitres 1 et 2 le contexte théorique de notre étude qui concerne les différentes techniques, les outils ainsi que toutes les connaissances théoriques dont nous avons besoin. Nous avons commencé dans le premier chapitre par un état de l'art des composantes des systèmes de reconnaissance automatique de la parole, spécialement les systèmes de RAP pour l'arabe au milieu bruité. Les SRAP s'appuient généralement sur cinq modules : la paramétrisation acoustique, le lexique, le modèle acoustique, le modèle de langage et le décodage. Aussi, nous avons donné un aperçu sur les différentes techniques existant dans la littérature scientifique et technique. Après, Dans le deuxième chapitre, nous avons continué l'aspect théorique lié à la présentation de divers techniques que nous avions utilisé dans notre thèse. Les méthodes de classifications et d'apprentissage des modèles acoustiques telles que : les modèles basés sur HMM, VQ, GMM, CNN ainsi que les modèles hybrides GMM-HMM et DNN-HMM, les outils open source les plus courantes comme Kaldi, PocketSphinx, HTK et Matlab sont présentés. Enfin, nous avons décrit les sources des différents types de bruits issus des bases de données vocales universelles parmi eux : NOISEX-92, AURORA, CHiME-3 et d'autres types que nous avions enregistré. Dans le troisième chapitre, nous avons expliqué le mécanisme du traitement de signal / bruit ainsi que les caractéristiques du bruit. Puis nous avons présenté les différents algorithmes de débruitage du signal de parole comme la méthode de la soustraction spectrale, la méthode de filtrage de Wiener et le débruitage avec le logiciel Audacity ®. Nous avons proposé ensuite dans le quatrième chapitre, un protocole expérimental qui consiste à tester l'approche d'injection du bruit dans les données utilisés dans l'apprentissage. En effet, nous avons implémenté plusieurs systèmes avec plusieurs tests par différents outils. Cette diversité des techniques de mise en œuvre de chaque module a été exploité pour construire différents systèmes. Afin d'atteindre plusieurs objectifs, notamment de comparer ces approches, donc de déduire le système et la méthode la plus efficace d'une part, et d'autre part d'évaluer l'apprentissage par les données vocales bruitées dans l'amélioration des performances des systèmes de RAP.
les résultats obtenus à travers les expériences que nous avions réalisé ont donné des résultats très satisfaisant comparativement aux méthodes existantes dans la littérature. L'utilisation de la combinaison entre l'approche d'entraînement en multi-conditions de bruit et les approches hybrides

VQ-GMM, GMM-HMM, et DNN-HMM dans la phase de décodage des systèmes de RAP que nous avons développé, ils sont plus robustes et performants par rapport au modèle utilisant une GMM, VQ et HMM seulement dans cette phase. En termes de WRR, Le modèle hybride DNN-HMM permet d'améliorer les performances avec une augmentation relative du taux WRR de 5,1% en moyen par rapport à l'utilisation de HMM sous la même base bruité NASCIW. De même, pour le modèle hybride VQ+GMM qui à ajouter une augmentation relative de 16,33% relativement à un système de reconnaissance basé sur le modèle GMM et une amélioration de 22,61% relativement à un système de reconnaissance basé sur le modèle VQ dans le cas de SNR=5 dB.
Durant les travaux de notre thèse, nous avons rencontré divers limitations, difficultés et défis.

1. Le manque des larges corpus bruités pour la langue arabe un des défis les plus importants auxquels nous avons été confrontés. Nous avons modifié certains corpus accessibles au public, comme ce fut le cas avec le corpus ASCIW[137]. Certaines approches, telles que les réseaux profonds, nécessitent des grandes quantités des données, donc, nous avons construit certaines corpus (SDDN) en ajoutons des bruits avec des SNR différentes.

2. Les machines utilisées sont également limitées et ne sont pas utiles pour l'entraînement avec des données à grande taille ou bien le choix d'une architecture complexe telle que le choix d'un réseau de neurones multicouches.

Pour la suite des travaux, nous envisageons des perspectives propres à chaque thématique de recherche abordée dans cette thèse. En d'autres termes, une certaine amélioration et des extensions significatives seront appliquées sur les algorithmes et méthodes développés dans le cadre de cette thèse. Les résultats que nous avons obtenus par nos systèmes sont très intéressants. Le travail présenté dans ce manuscrit est une démarche pour répondre à la problématique soulevée. Nous suggérons quand même quelques axes de recherche en guise de prolongement à ce travail.

- Elargir le vocabulaire du corpus d'apprentissage et varier les conditions d'enregistrement en ajoutant beaucoup de types de bruits et plusieurs locuteurs mixtes, utiliser des données enregistrées en situation réelle, l'entraîner sur plusieurs bases de données en même temps, pour contribuer aux tests de la robustesse des systèmes de reconnaissance automatique de la parole Arabe ou la dialecte marocaine ou pour les dialectes Amazigh. L'impact direct serait l'amélioration de la qualité des modèles acoustiques et du langage pour un meilleur taux de reconnaissance.
- Les solutions proposées concernant le traitement et la reconnaissance de la parole, qui fonctionne dans un nouvel environnement composé de différents bruits, comme une tâche d'adaptation du domaine. Elles sont certainement incomplètes pour généraliser ou de la considèrer comme une solution finale, mais une analyse approfondie doit être accomplie en entrant dans le cœur des algorithmes qui composent les approches que nous avons utilisées, y compris la ré-étude du paramètre de bruit sur les données d'entraînement et son interférence avec différentes unités acoustiques. Nos recherches peuvent également être orientées vers la modélisation du bruit à tous les niveaux SNR dans les systèmes de traitement du signal.
- Récemment, de nouvelles techniques ont été développées à partir de Deep Learning, qui ont influencé plusieurs études sur le traitement du signal et la reconnaissance de la parole. Dans

cette thèse nous avons proposé deux type des réseaux de nouerons CNN et DNN pour valider notre approche d'entrainement bruité, nous pouvons également utiliser une autre architecture différente de type réseaux de neurones convolutifs ou d'utiliser d'autres types des réseaux profonds en fournirent des quantités de données croissantes avec des machines appropriées disponibles. Comme par exemple, des réseaux de neurones récurrents (RNN), les réseaux de longue mémoire à court terme (LSTM) ou l'apprentissage par renforcement. Le modèle combiné CNN et RNN est peut-être aussi une meilleure implémentation pour la reconnaissance des commandes vocales que le modèle CNN seul.

- Afin d'améliorer les performances de nos systèmes proposés, il faut fournir des ressources de calcul et de stockage plus robustes et plus et avancés, qui permettent d'augmenter la puissance de calcul des ordinateurs ce qui a permis d'entraîner des réseaux plus grands en un temps raisonnable. Peut être prise en compte des nouveaux GPUs de Nvidia, les High-performance computing (HPC), ou des superordinateurs comme TensorBook, MacBook Pro 15″, etc.
- Il faut étendre ce système et appliquer une approche analytique, en utilisant des modèles de phonèmes pour la reconnaissance de la parole continue.
- Aujourd'hui, le Deep Learning (DNN, DBN.) est une technique émergente dans le domaine de la reconnaissance de parole, donc nous suggérons à faire intervenir les DNN dans le processus de segmentation dont le but est d'améliorer les performances des systèmes de reconnaissance de la parole.
- Nous envisageons d'étudier la distance entre le dialecte marocain et de la langue arabe afin de générer des représentations spécifiques et les intégrer dans notre système de base sur CNN appris sur des données arabes.
- Utiliser les corpus que nous avons construit pour comparer les performances des algorithmes de débruitage ou d'amélioration de la parole, ou pour les différentes techniques d'analyse acoustique. On peut aussi faire une comparaison entre notre approche d'apprentissage bruité et les algorithmes de rehaussement de la parole en termes d'efficacité, de performance et de coût.
- La langue arabe contient beaucoup de règles de grammaire et des signes diacritiques. Par conséquent, nous recommandons de travailler sur des système en traitant ces paramètres.
- Les résultats obtenus sont très satisfaisant compte tenu du délai imparti. Néanmoins, il est possible d'aller encore plus loin. De manière générale, l'accès à plus de données permettrait d'améliorer les performances à tous les niveaux (données audio pour le modèle acoustique et données textuelles pour le modèle de langage).
- Par ailleurs, le modèle acoustique est améliorable par l'utilisation d'architecture de réseaux de neurones encore plus complexes et par autre classifieur (RNN, LSTM, SVM, i-vector, ...). Dont l'implémentation est simplifiée par l'utilisation de l'outil Pytorch-Kaldi. L'idée étant d'entraîner ces nouveaux modèles à partir des meilleurs alignements disponibles (modèle DNN Kaldi).
- Il existe de nombreux projets applicables qui peuvent être intégrés dans les appareils intelligents et robotiques lesquels répondre et interagir avec la parole ou les dialectes arabes nous y travaillerons à l'avenir.

Références

[1] M. Nilsson and M. Ejnarsson, 'Speech Recognition using Hidden Markov Model performance evaluation in noisy environment', 2002.
[2] D. Yu and L. Deng, 'Introduction', in *Automatic Speech Recognition: A Deep Learning Approach*, D. Yu and L. Deng, Eds. London: Springer, 2015, pp. 1–9.
[3] Daniel Jurafsky and James H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Prentice-Hall, Inc., Upper Saddle River. NJ, USA, 2009.
[4] Jinyu Li, Li Deng, Reinhold Haeb-Umbach, and Yifan Gong, *Robust Automatic Speech Recognition*. Elsevier, 2016.
[5] R. Haeb-Umbach and A. Krueger, 'Reverberant Speech Recognition', in *Techniques for Noise Robustness in Automatic Speech Recognition*, John Wiley & Sons, Ltd, 2012, pp. 251–281.
[6] K. Kinoshita *et al.*, 'The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech', in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 2013, pp. 1–4.
[7] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, 'The third "CHiME" speech separation and recognition challenge: Dataset, task and baselines', in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015, pp. 504–511. doi: 10.1109/ASRU.2015.7404837.
[8] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker, 'Perplexity—a measure of the difficulty of speech recognition tasks', *The Journal of the Acoustical Society of America*, 1977.
[9] 'An algorithm for determining the endpoints of isolated utterances - Nokia Bell Labs Journals & Magazine'.
[10] S. Davis and P. Mermelstein, 'Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences', *IEEE Trans. Acoust. Speech Signal Process.*, vol. 28, no. 4, pp. 357–366, Aug. 1980, doi: 10.1109/TASSP.1980.1163420.
[11] H. Hermansky and L. A. Cox, 'Perceptual Linear Predictive (PLP) Analysis-Resynthesis Technique', in *Final Program and Paper Summaries 1991 IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 1991, p. 0_37-0_38. doi: 10.1109/ASPAA.1991.634094.
[12] D. Pati and S. R. M. Prasanna, 'Processing of linear prediction residual in spectral and cepstral domains for speaker information', *Int. J. Speech Technol.*, vol. 18, no. 3, pp. 333–350, Sep. 2015, doi: 10.1007/s10772-015-9273-9.
[13] K. Gupta and D. Gupta, 'An analysis on LPC, RASTA and MFCC techniques in Automatic Speech recognition system', in *2016 6th International Conference - Cloud System and Big Data En-*

gineering (Confluence), Jan. 2016, pp. 493–497. doi: 10.1109/CONFLUENCE.2016.7508170.
[14] F. J. Harris, 'On the use of windows for harmonic analysis with the discrete Fourier transform', *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978, doi: 10.1109/PROC.1978.10837.
[15] F. Lahouti, A. R. Fazel, A. H. Safavi-Naeini, and A. K. Khandani, 'Single and double frame coding of speech LPC parameters using a lattice-based quantization scheme', *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 5, pp. 1624–1632, Sep. 2006, doi: 10.1109/TSA.2005.858560.
[16] Bhadragiri Jagan Mohan and Ramesh Babu N., 'Speech recognition using MFCC and DTW', in *2014 International Conference on Advances in Electrical Engineering (ICAEE)*, Jan. 2014, pp. 1–4.
[17] 'Study of MFCC and IHC Feature Extraction Methods With Probabilistic Acoustic Models for Speaker Biometric Applications', *Procedia Comput. Sci.*, vol. 143, pp. 267–276, Jan. 2018, doi: 10.1016/j.procs.2018.10.395.
[18] M. Dua, R. K. Aggarwal, and M. Biswas, 'Performance evaluation of Hindi speech recognition system using optimized filterbanks', *Eng. Sci. Technol. Int. J.*, vol. 21, no. 3, pp. 389–398, Jun. 2018, doi: 10.1016/j.jestch.2018.04.005.
[19] H. M. S. Naing, Y. Miyanaga, R. Hidayat, and B. Winduratna, 'Filterbank Analysis of MFCC Feature Extraction in Robust Children Speech Recognition', in *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*, Aug. 2019, pp. 1–6. doi: 10.1109/ISMAC.2019.8836181.
[20] M. A. Hossan, S. Memon, and M. A. Gregory, 'A novel approach for MFCC feature extraction', in *2010 4th International Conference on Signal Processing and Communication Systems*, Dec. 2010, pp. 1–5. doi: 10.1109/ICSPCS.2010.5709752.
[21] D. Desai and M. Joshi, 'Speaker Recognition Using MFCC and Hybrid Model of VQ and GMM', in *Recent Advances in Intelligent Informatics*, Cham, 2014, pp. 53–63. doi: 10.1007/978-3-319-01778-5_6.
[22] E. S. Wahyuni, 'Arabic speech recognition using MFCC feature extraction and ANN classification', in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Nov. 2017, pp. 22–25. doi: 10.1109/ICITISEE.2017.8285499.
[23] S. Lokesh and M. R. Devi, 'Speech recognition system using enhanced mel frequency cepstral coefficient with windowing and framing method', *Clust. Comput.*, vol. 22, no. 5, pp. 11669–11679, Sep. 2019, doi: 10.1007/s10586-017-1447-6.
[24] Aradilla, Guillermo, 'Acoustic models for posterior features in speech recognition', Lausanne, EPFL, LIDIAP, 2008. doi: 10.5075/epfl-thesis-4164.
[25] F. A. A. Laleye, 'Contributions à l'étude et à la reconnaissance automatique de la parole en Fongbe', phdthesis, Université du Littoral Côte d'Opale, 2016.
[26] Alsteris, Leigh David / Paliwal, Kuldip K, 'ASR on Speech Reconstructed from Short-time Fourier Phase Spectra', Jeju Island, Korea, Oct. 2004, pp. 565–568.
[27] S. Sen, A. Dutta, and N. Dey, *Audio Processing and Speech Recognition: Concepts, Techniques and Research Overviews*. Springer Singapore, 2019. doi: 10.1007/978-981-13-6098-5.
[28] O. Siohan, 'Reconnaissance automatique de la parole continue en environnement bruité : application à des modéles stochastiques de trajectoires', phdthesis, Université Henri Poincaré - Nancy 1, 1995. Accessed: Oct. 24, 2020.
[29] J.-P. Haton, 'Méthodes robustes pour la reconnaissance automatique de la parole', 2002, Ac-

cessed: Oct. 24, 2020.
[30] T. Virtanen, R. Singh, and B. Raj, *Techniques for Noise Robustness in Automatic Speech Recognition.* John Wiley & Sons, 2012.
[31] H. Hermansky and N. Morgan, 'RASTA processing of speech', *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994, doi: 10.1109/89.326616.
[32] R. M. Stern, A. Acero, F.-H. Liu, and Y. Ohshima, 'Signal Processing for Robust Speech Recognition', in *Automatic Speech and Speaker Recognition: Advanced Topics*, C.-H. Lee, F. K. Soong, and K. K. Paliwal, Eds. Boston, MA: Springer US, 1996, pp. 357–384. doi: 10.1007/978-1-4613-1367-0_15.
[33] Jae Lim, 'Spectral root homomorphic deconvolution system', *IEEE Trans. Acoust. Speech Signal Process.*, vol. 27, no. 3, pp. 223–233, Jun. 1979, doi: 10.1109/TASSP.1979.1163234.
[34] T. Kobayashi and S. Imai, 'Spectral analysis using generalised cepstrum', *IEEE Trans. Acoust. Speech Signal Process.*, vol. 32, no. 6, pp. 1235–1238, Dec. 1984.
[35] M. J. Hunt and C. Lefebvre, 'A comparison of several acoustic representations for speech recognition with degraded and undegraded speech', in *International Conference on Acoustics, Speech, and Signal Processing,* May 1989, pp. 262–265 vol.1. doi: 10.1109/ICASSP.1989.266415.
[36] A. Ouisaadane, S. Safi, and M. Frikel, 'Arabic digits speech recognition and speaker identification in noisy environment using a hybrid model of VQ and GMM', *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 18, no. 4, Art. no. 4, Aug. 2020, doi: 10.12928/telkomnika.v18i4.14215.
[37] E. Guerra, C. J., and J. Meng, 'Noise-robust speech recognition system based on power spectral subtraction', Apr. 2012.
[38] T. Fux and D. Jouvet, 'Evaluation of PNCC and extended spectral subtraction methods for robust speech recognition', Nice, France, Aug. 2015. Accessed: Oct. 28, 2020. [Online]. Available: https://hal.inria.fr/hal-01183645
[39] F.-H. Liu, 'Environmental adaptation for robust speech recognition', phd, Carnegie Mellon University, USA, 1995.
[40] J. Ramírez and J. M. Górriz, *Recent Advances in Robust Speech Recognition Technology.* Bentham Science, 2011.
[41] J. Rajnoha and P. Pollak, 'ASR Systems in Noisy Environment: Analysis and Solutions for Increasing Noise Robustness', Apr. 2011, Accessed: Oct. 29, 2020.
[42] Chia-Ping Chen and J. Bilmes, 'Noise robustness in automatic speech recognition', Computer Science, 2004.
[43] J.-P. Haton, 'Automatic Recognition of Noisy Speech', in *Speech Recognition and Coding*, Berlin, Heidelberg, 1995, pp. 3–13. doi: 10.1007/978-3-642-57745-1_1.
[44] A. Ouisaadane, S. Safi, and M. Frikel, 'English Spoken Digits Database under noise conditions for research: SDDN', in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, Apr. 2019, pp. 1–5. doi: 10.1109/WITS.2019.8723698.
[45] L. Buniet, 'Traitement automatique de la parole en milieu bruité : étude de modéles connexionnistes statiques et dynamiques', phdthesis, Université Henri Poincaré - Nancy 1, 1997.
[46] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero, 'Noise Adaptive Training for Robust Automatic Speech Recognition', *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 8, pp. 1889–1901, Nov. 2010, doi: 10.1109/TASL.2010.2040522.

[47] K. Palomäki, U. Remes, and M. Kurimo, 'Studies on Noise Robust Automatic Speech Recognition', *978-952-248-087-3*, Jun. 2013.
[48] A. Touazi and M. Debyeche, 'An experimental framework for Arabic digits speech recognition in noisy environments', *Int. J. Speech Technol.*, vol. 20, no. 2, pp. 205–224, Jun. 2017, doi: 10.1007/s10772-017-9400-x.
[49] M. Telmem and Y. Ghanou, 'Amazigh Speech Recognition System Based on CMUSphinx', in *Innovations in Smart Cities and Applications*, Cham, 2018, pp. 397–410. doi: 10.1007/978-3-319-74500-8_37.
[50] A. I. Amrous, M. Debyeche, and A. Amrouche, 'Robust Arabic speech recognition in noisy environments using prosodic features and formant', *Int. J. Speech Technol.*, vol. 14, no. 4, p. 351, Sep. 2011, doi: 10.1007/s10772-011-9113-5.
[51] A. I. Amrous, 'Coopération de connaissances dans les systémes de reconnaissance automatique de la parole', Thesis, 2015. Accessed: Oct. 30, 2020.
[52] M. Hamidi, H. Satori, O. Zealouk, and K. Satori, 'Amazigh digits through interactive speech recognition system in noisy environment', *Int. J. Speech Technol.*, vol. 23, no. 1, pp. 101–109, Mar. 2020, doi: 10.1007/s10772-019-09661-2.
[53] S. El Ouahabi, M. Atounti, and M. Bellouki, 'Toward an automatic speech recognition system for amazigh-tarifit language', *Int. J. Speech Technol.*, vol. 22, no. 2, pp. 421–432, Jun. 2019, doi: 10.1007/s10772-019-09617-6.
[54] A. I. Amrous and M. Debyeche, 'Robust Arabic Multi-stream Speech Recognition System in Noisy Environment', in *Image and Signal Processing*, Berlin, Heidelberg, 2012, pp. 571–578. doi: 10.1007/978-3-642-31254-0_65.
[55] A. Alalshekmubarak and L. S. Smith, 'A noise robust Arabic speech recognition system based on the echo state network', *J. Acoust. Soc. Am.*, vol. 135, no. 4, pp. 2195–2195, Apr. 2014, doi: 10.1121/1.4877154.
[56] O. Zealouk, M. Hamidi, H. Satori, and K. Satori, 'Amazigh Digits Speech Recognition System Under Noise Car Environment', in *Embedded Systems and Artificial Intelligence*, Singapore, 2020, pp. 421–428. doi: 10.1007/978-981-15-0947-6_40.
[57] H. H. O. Nasereddin and A. A. R. Omari, 'Classification techniques for automatic speech recognition (ASR) algorithms used with real time speech translation', in *2017 Computing Conference*, Jul. 2017, pp. 200–207. doi: 10.1109/SAI.2017.8252104.
[58] J. Meng, J. Zhang, and H. Zhao, 'Overview of the Speech Recognition Technology', in *2012 Fourth International Conference on Computational and Information Sciences*, Aug. 2012, pp. 199–202. doi: 10.1109/ICCIS.2012.202.
[59] L. R. Rabiner and B. H. Juang, 'Hidden Markov Models for Speech Recognition — Strengths and Limitations', in *Speech Recognition and Understanding*, Berlin, Heidelberg, 1992, pp. 3–29. doi: 10.1007/978-3-642-76626-8_1.
[60] V. Barreaud, 'Reconnaissance automatique de la parole continue : compensation des bruits par transformation de la parole', These de doctorat, Nancy 1, 2004.
[61] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. USA: Prentice-Hall, Inc., 1993.
[62] K. Audhkhasi, O. Osoba, and B. Kosko, 'Noisy hidden Markov models for speech recognition', in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1–6. doi:

10.1109/IJCNN.2013.6707088.
[63] D. Nagajyothi and P. Siddaiah, 'Voice Recognition Based on Vector Quantization Using LBG', in *Computer Communication, Networking and Internet Security*, Singapore, 2017, pp. 503–511. doi: 10.1007/978-981-10-3226-4_51.
[64] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, 'Speaker recognition using Mel frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques', in *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*, Feb. 2012, pp. 248–251. doi: 10.1109/CONIELECOMP.2012.6189918.
[65] Y. Linde, A. Buzo, and R. Gray, 'An Algorithm for Vector Quantizer Design', *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, Jan. 1980, doi: 10.1109/TCOM.1980.1094577.
[66] W. S. McCulloch and W. Pitts, 'A logical calculus of the ideas immanent in nervous activity', *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.
[67] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. Mahwah, N.J, 2002.
[68] I. T. Brain and F. Rosenblatt, *The Perceptron: A Probabilistic Model for Information Storage and Organization*.
[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 'ImageNet classification with deep convolutional neural networks', *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
[70] Center for Electronics and Information Studies, 'Deep Learning: History and State-of-the-Arts', in *The Development of Deep Learning Technologies: Research on the Development of Electronic Information Engineering Technology in China*, Singapore: Springer, 2020, pp. 1–11. doi: 10.1007/978-981-15-4584-9_1.
[71] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, 'Speech Recognition Using Deep Neural Networks: A Systematic Review', *IEEE Access*, vol. 7, pp. 19143–19165, 2019, doi: 10.1109/ACCESS.2019.2896880.
[72] H. I. Suk, 'An Introduction to Neural Networks and Deep Learning', *Deep Learn. Med. Image Anal.*, pp. 3–24, Jan. 2017, doi: 10.1016/B978-0-12-810408-8.00002-X.
[73] C. Wu Chunyang, 'Structured Deep Neural Networks for Speech Recognition', Thesis, Cambridge, Wolfson College, 2018.
[74] D. Fohr, O. Mella, and I. Illina, 'New Paradigm in Speech Recognition: Deep Neural Networks', presented at the IEEE International Conference on Information Systems and Economic Intelligence, Apr. 2017. Accessed: Nov. 18, 2020.
[75] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Qadir, and B. W. Schuller, 'Deep Representation Learning in Speech Processing: Challenges, Recent Advances, and Future Trends', *ArXiv200100378 Cs Eess*, Jan. 2020, Accessed: Nov. 20, 2020.
[76] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, 'Learning representations by back-propagating errors', *Nature*, vol. 323, no. 6088, Art. no. 6088, Oct. 1986, doi: 10.1038/323533a0.
[77] J. J. Hopfield, 'Neural networks and physical systems with emergent collective computational abilities', *Proc. Natl. Acad. Sci.*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982, doi: 10.1073/pnas.79.8.2554.
[78] T. N. Sainath *et al.*, 'Two-Pass End-to-End Speech Recognition', *ArXiv190810992 Cs Eess*, Aug. 2019, Accessed: Nov. 20, 2020.
[79] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, 'Convolutional Neural Networks for Speech Recognition', *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 10,

pp. 1533–1545, Oct. 2014, doi: 10.1109/TASLP.2014.2339736.
[80] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, 'Deep Learning for Environmentally Robust Speech Recognition: An Overview of Recent Developments', May 2017, Accessed: Nov. 20, 2020.
[81] I. J. Goodfellow *et al.*, 'Generative Adversarial Networks', *ArXiv14062661 Cs Stat*, Jun. 2014, Accessed: Nov. 20, 2020.
[82] G. Hinton *et al.*, 'Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups', *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: 10.1109/MSP.2012.2205597.
[83] A. Abraham, 'Continous Speech Recognition Using Long Term Memory Cells', Ohio University, 2013. Accessed: Nov. 24, 2020.
[84] A. M. A. M. Ali, 'Multi-dialect Arabic broadcast speech recognition', Jul. 2018, Accessed: Nov. 24, 2020.
[85] W. Algihab, N. Alawwad, A. Aldawish, and S. AlHumoud, 'Arabic Speech Recognition with Deep Learning: A Review', in *Social Computing and Social Media. Design, Human Behavior and Analytics*, Cham, 2019, pp. 15–31. doi: 10.1007/978-3-030-21902-4_2.
[86] N. Hammami and M. Sellam, 'Tree distribution classifier for automatic spoken Arabic digit recognition', in *2009 International Conference for Internet Technology and Secured Transactions, (ICITST)*, Nov. 2009, pp. 1–4. doi: 10.1109/ICITST.2009.5402575.
[87] M. Elmahdy, R. Gruhn, and W. Minker, *Novel Techniques for Dialectal Arabic Speech Recognition*. New York: Springer-Verlag, 2012. doi: 10.1007/978-1-4614-1906-8.
[88] Z. Ling, 'An Acoustic Model for English Speech Recognition Based on Deep Learning', in *2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Apr. 2019, pp. 610–614. doi: 10.1109/ICMTMA.2019.00140.
[89] L. Deng, G. Hinton, and B. Kingsbury, 'New types of deep neural network learning for speech recognition and related applications: an overview', in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8599–8603. doi: 10.1109/ICASSP.2013.6639344.
[90] D. Wang, X. Wang, and S. Lv, 'An Overview of End-to-End Automatic Speech Recognition', *Symmetry*, vol. 11, no. 8, Art. no. 8, Aug. 2019, doi: 10.3390/sym11081018.
[91] G. E. Dahl, D. Yu, L. Deng, and A. Acero, 'Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition', *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012, doi: 10.1109/TASL.2011.2134090.
[92] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. London: Springer-Verlag, 2015. doi: 10.1007/978-1-4471-5779-3.
[93] A. E. V.- Luna *et al.*, 'De-Noising Audio Signals Using MATLAB Wavelets Toolbox', *Eng. Educ. Res. Using MATLAB*, Oct. 2011, doi: 10.5772/19851.
[94] A. Varga and H. J. M. Steeneken, 'Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems', *Speech Commun.*, vol. 12, no. 3, pp. 247–251, Jul. 1993, doi: 10.1016/0167-6393(93)90095-3.
[95] D. Pearce and H.-G. Hirsch, 'The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy condition', Jan. 2000, vol. 4, pp. 29–32.
[96] A. Ouisaadane, S. Safi, and M. Frikel, 'Reconnaissance Automatique de la dialecte marocain en

milieu réel à l'aide de PocketSphinx', CAEN, France, Jun. 2020. Accessed: Dec. 04, 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02974050

[97] P. Lamere *et al.*, *The CMU SPHINX-4 Speech Recognition System*. 2003.

[98] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, 'Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices', in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, May 2006, vol. 1, p. I–I. doi: 10.1109/ICASSP.2006.1659988.

[99] Povey D *et al.*, 'The Kaldi Speech Recognition Toolkit', *ASRU2011 IEEE*, 2011.

[100] S. Young *et al.*, *The HTK Book (version 3.5a)*. 2015.

[101] S. P. Gongi, 'Assessment of the Extent and Perceived Effects of Noise Pollution in Manufacturing Industries in Nairobi City', Thesis, JKUAT COETEC, 2018.

[102] NIOSH, 'Criteria for a recommended standard... occupational noise exposure, revised criteria 1998.', 1998, doi: 10.26616/NIOSHPUB98126.

[103] N. Das, S. Chakraborty, J. Chaki, N. Padhy, and N. Dey, 'Fundamentals, present and future perspectives of speech enhancement', *Int. J. Speech Technol.*, Jan. 2020, doi: 10.1007/s10772-020-09674-2.

[104] Colin H Hansen, 'FUNDAMENTALS OF ACOUSTICS', *WHO Inoccupationalhealtht*.

[105] J. Price and T. Goble, '10 - Signals and noise', in *Telecommunications Engineer's Reference Book*, F. Mazda, Ed. Butterworth-Heinemann, 1993, pp. 10–1. doi: 10.1016/B978-0-7506-1162-6.50016-2.

[106] Pedro J. Moreno, 'Speech Recognition in Noisy Environments', Department of Electrical and Computer Engineering Carnegie Mellon University Pittsburgh, Pennsylvania, 1996.

[107] U. Shrawankar and V. M. Thakare, 'Adverse Conditions and ASR Techniques for Robust Speech User Interface', *ArXiv13035515 Cs*, Mar. 2013.

[108] Yariv Ephraim, 'A Brief Survey of Speech Enhancement', in *Microelectronics*, 2nd Edition., CRC Press, 2018, pp. 20-1-20–12. doi: 10.1201/9781315220482-20.

[109] J. Ramírez and J. M. Górriz, *Recent Advances in Robust Speech Recognition Technology*. Bentham Science, 2011.

[110] S. Gulati, 'Comprehensive Review of Various Speech Enhancement Techniques', in *Computational Vision and Bio-Inspired Computing*, Cham, 2020, pp. 536–540. doi: 10.1007/978-3-030-37218-7_61.

[111] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Noise Reduction in Speech Processing*, 2009th edition. Dordrecht ; London ; New York: Springer, 2009.

[112] A. Chaudhari and S. B. Dhonde, 'A review on speech enhancement techniques', in *2015 International Conference on Pervasive Computing (ICPC)*, Jan. 2015, pp. 1–3. doi: 10.1109/PERVASIVE.2015.7087096.

[113] D. M. Badescu, 'Speech Enhancement Using Deep Learning', Universitat Politécnica de Catalunya. Escola Técnica Superior d'Enginyeria de Telecomunicació de Barcelona, 2017.

[114] S. S. Nidhyananthan, R. S. S. Kumari, and A. A. Prakash, 'A review on speech enhancement algorithms and why to combine with environment classification', *Int. J. Mod. Phys. C*, vol. 25, no. 10, p. 1430002, Mar. 2014, doi: 10.1142/S0129183114300024.

[115] A. H. Moore, P. Peso Parada, and P. A. Naylor, 'Speech enhancement for robust automatic

speech recognition: Evaluation using a baseline system and instrumental measures', *Comput. Speech Lang.*, vol. 46, pp. 574–584, Nov. 2017, doi: 10.1016/j.csl.2016.11.003.
[116] S. Watanabe, M. Delcroix, F. Metze, and J. R. Hershey, *New Era for Robust Speech Recognition: Exploiting Deep Learning*. Springer International Publishing, 2017.
[117] K. Rohith and K. Chethan, 'Comparitive Analysis of Speech Enhancement Techniques: A Review', in *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, Sep. 2017, pp. 562–565. doi: 10.1109/CTCEEC.2017.8455202.
[118] K. K. Paliwal and K. Yao, 'Chapter 6 - Robust Speech Recognition Under Noisy Ambient Conditions', in *Human-Centric Interfaces for Ambient Intelligence*, H. Aghajan, R. L.-C. Delgado, and J. C. Augusto, Eds. Oxford: Academic Press, 2010, pp. 135–162. doi: 10.1016/B978-0-12-374708-2.00006-1.
[119] M. V. Lakshmi, S. Sudha, and A. Aswini, 'A comparative study on speech enhancement methods — performance evaluation of SNR for hearing aid listeners', in *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, Jul. 2016, pp. 81–85. doi: 10.1109/TIAR.2016.7801218.
[120] Siddala Vihari, A. Sreenivasa Murthy, Priyanka Soni, and D.C.Naik, 'Comparison of Speech Enhancement Algorithms', *Procedia Comput. Sci.*, vol. 89, pp. 666–676, Jan. 2016.
[121] Navneet Upadhyay and Abhijit Karmakar, 'Speech Enhancement using Spectral Subtraction-type Algorithms: A Comparison and Simulation Study', *Procedia Comput. Sci.*, vol. 54, pp. 574–584, Jan. 2015, doi: 10.1016/j.procs.2015.06.066.
[122] S. Singh, M. Tripathy, and R. S. Anand, 'Binary mask based method for enhancement of mixed noise speech of low SNR input', *Int. J. Speech Technol.*, vol. 18, no. 4, pp. 609–617, Dec. 2015, doi: 10.1007/s10772-015-9305-5.
[123] T. D. Tran, Q. C. Nguyen, and D. K. Nguyen, 'Speech enhancement using modified IMCRA and OMLSA methods', in *International Conference on Communications and Electronics 2010*, Aug. 2010, pp. 195–200. doi: 10.1109/ICCE.2010.5670709.
[124] S. Boll, 'Suppression of acoustic noise in speech using spectral subtraction', *IEEE Trans. Acoust. Speech Signal Process.*, vol. 27, no. 2, pp. 113–120, Apr. 1979, doi: 10.1109/TASSP.1979.1163209.
[125] M. A. Abd El-Fattah *et al.*, 'Speech enhancement with an adaptive Wiener filter', *Int. J. Speech Technol.*, vol. 17, no. 1, pp. 53–64, Mar. 2014, doi: 10.1007/s10772-013-9205-5.
[126] J. Franklin, 'The Sheer Audacity: How to Get More, in Less Time, from the Audacity Digital Audio Editing Software', in *2006 IEEE International Professional Communication Conference*, Oct. 2006, pp. 92–105. doi: 10.1109/IPCC.2006.320394.
[127] van Khanh Mai, 'Méthodes avancées de traitement de la parole et de réduction de bruit pour les terminaux mobiles', These de doctorat, Ecole nationale supérieure Mines-Télécom Atlantique Bretagne Pays de la Loire, 2017. Accessed: Feb. 01, 2021.
[128] D. Y. Zhao and W. B. Kleijn, 'HMM-Based Gain Modeling for Enhancement of Speech in Noise', *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 3, pp. 882–892, Mar. 2007, doi: 10.1109/TASL.2006.885256.
[129] H. Veisi and H. Sameti, 'Speech enhancement using hidden Markov models in Mel-frequency domain', *Speech Commun.*, vol. 55, no. 2, pp. 205–220, Feb. 2013, doi: 10.1016/j.specom.2012.08.005.
[130] N. Mohammadiha, R. Martin, and A. Leijon, 'Spectral Domain Speech Enhancement Using

HMM State-Dependent Super-Gaussian Priors', *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 253–256, Mar. 2013, doi: 10.1109/LSP.2013.2242467.

[131] E. Vincent, T. Virtanen, and S. Gannot, *Audio source separation and speech enhancement*. Wiley, 2018, p. 504. doi: 10.1002/9781119279860.

[132] Chul Min Lee, Jong Won Shin, and Nam Soo Kim, 'DNN-based residual echo suppression', Dresden, Germany, Sep. 2015, pp. 1775–1779. Accessed: Feb. 22, 2021.

[133] H. Erdogan, J. R. Hershey, S. Watanabe, and J. L. Roux, 'Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks', in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 708–712. doi: 10.1109/ICASSP.2015.7178061.

[134] A. Kumar and D. Florencio, 'Speech Enhancement In Multiple-Noise Conditions using Deep Neural Networks', *ArXiv160502427 Cs*, May 2016, Accessed: Jan. 21, 2021.

[135] A. Ouisaadane, S. Safi, and M. Frikel, 'English Spoken Digits Database under noise conditions for research: SDDN', in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, Apr. 2019, pp. 1–5. doi: 10.1109/WITS.2019.8723698.

[136] P. Warden, 'Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition', *ArXiv180403209 Cs*, Apr. 2018, Accessed: Apr. 15, 2021.

[137] A. Alalshekmubarak and L. S. Smith, 'On Improving the Classification Capability of Reservoir Computing for Arabic Speech Recognition', in *Artificial Neural Networks and Machine Learning – ICANN 2014*, Cham, 2014, pp. 225–232. doi: 10.1007/978-3-319-11179-7_29.

[138] H. M. M. Eljagmani, 'Arabic Speech Recognition Systems', Thesis, 2017. Accessed: Apr. 22, 2021.

[139] R. E. Gruhn, W. Minker, and S. Nakamura, *Statistical Pronunciation Modeling for Non-Native Speech Processing*. Berlin Heidelberg: Springer-Verlag, 2011. doi: 10.1007/978-3-642-19586-0.

[140] A. Senior, G. Heigold, M. Bacchiani, and H. Liao, 'GMM-free DNN acoustic model training', in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 5602–5606. doi: 10.1109/ICASSP.2014.6854675.

[141] B. Karan, J. Sahoo, and P. K. Sahu, 'Automatic speech recognition based Odia system', in *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, Dec. 2015, pp. 353–356. doi: 10.1109/ICMOCE.2015.7489765.

[142] J. Guglani and A. N. Mishra, 'DNN based continuous speech recognition system of Punjabi language on Kaldi toolkit', *Int. J. Speech Technol.*, vol. 24, no. 1, pp. 41–45, Mar. 2021, doi: 10.1007/s10772-020-09717-8.

[143] P. S. Praveen Kumar, G. Thimmaraja Yadava, and H. S. Jayanna, 'Continuous Kannada Speech Recognition System Under Degraded Condition', *Circuits Syst. Signal Process.*, vol. 39, no. 1, pp. 391–419, Jan. 2020, doi: 10.1007/s00034-019-01189-9.

[144] P. Mizera, J. Fiala, A. Brich, and P. Pollak, 'KALDI Recipes for the Czech Speech Recognition Under Various Conditions', in *Text, Speech, and Dialogue*, Cham, 2016, pp. 391–399. doi: 10.1007/978-3-319-45510-5_45.

[145] D. Povey, X. Zhang, and S. Khudanpur, 'Parallel training of DNNs with Natural Gradient and Parameter Averaging', Oct. 2014, Accessed: May 29, 2021.

[146] E. R. Abdelmaksoud, A. Hassen, N. Hassan, and M. Hesham, 'CONVOLUTIONAL NEURAL

NETWORK FOR ARABIC SPEECH RECOGNITION', *Egypt. J. Lang. Eng.*, vol. 8, no. 1, pp. 27–38, Apr. 2021, doi: 10.21608/ejle.2020.47685.1015.

[147] J. A. Qadir, A. K. Al-Talabani, and H. A. Aziz, 'Isolated Spoken Word Recognition Using One-Dimensional Convolutional Neural Network', *Int. J. Fuzzy Log. Intell. Syst.*, vol. 20, no. 4, pp. 272–277, Dec. 2020, doi: 10.5391/IJFIS.2020.20.4.272.

[148] K.-K. Fatima, M. Larbi, and H. Fatiha, 'Recognition of Noisy Words Using the Time Delay Neural Networks Approach', *Int. J. Comput. Inf. Eng.*, vol. 8, no. 9, pp. 1433–1437, Aug. 2014.

[149] B. Dendani, H. Bahi, and T. Sari, 'Speech Enhancement Based on Deep AutoEncoder for Remote Arabic Speech Recognition', in *Image and Signal Processing*, Cham, 2020, pp. 221–229. doi: 10.1007/978-3-030-51935-3_24.

[150] N. Zerari, S. Abdelhamid, H. Bouzgou, and C. Raymond, 'Bidirectional deep architecture for Arabic speech recognition', *Open Comput. Sci.*, vol. 9, no. 1, pp. 92–102, Jan. 2019, doi: 10.1515/comp-2019-0004.

[151] A. S. Mahfoudh BA WAZIR and J. Huang CHUAH, 'Spoken Arabic Digits Recognition Using Deep Learning', in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Jun. 2019, pp. 339–344. doi: 10.1109/I2CACIS.2019.8825004.

[152] R. A. Rajagede, C. K. Dewa, and Afiahayati, 'Recognizing Arabic letter utterance using convolutional neural network', in *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Jun. 2017, pp. 181–186. doi: 10.1109/SNPD.2017.8022720.

[153] Bezoui Mouaz, 'Speech Recognition of Moroccan Dialect Using Hidden Markov Models', *Procedia Comput. Sci.*, vol. 151, pp. 985–991, Jan. 2019, doi: 10.1016/j.procs.2019.04.138.