

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب بليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Electronique



Mémoire de Projet de Fin d'Études

Présenté par

Silia RADJEF

Madina MAMMOU

Pour l'obtention du diplôme de Master en Electronique

Spécialité : Electronique des Systèmes Embarqués

Thème

Reconnaissance des Commandes Vocales en Dialecte Algérien pour une Gestion Intelligente des Parcs de Stationnement

Proposé par :

Promoteur :

Fayçal YKHLEF, Maître de Recherche A, CDTA, Alger.

Co-promoteur :

Farid YKHLEF, Professeur, Université Blida 1, Blida.

Année Universitaire 2023-2024

REMERCIEMENTS

Nous débutons ce mémoire en exprimant notre profonde reconnaissance envers Dieu Tout-Puissant pour son soutien précieux tout au long de ce projet. Sans sa guidance, nous n'aurions jamais pu réussir.

Nous souhaitons tout d'abord remercier Monsieur YKHLEF Fayçal, Maître de Recherche A au CDTA, qui nous a proposé ce thème de mémoire et a accepté d'en assurer la supervision. Son rôle a été déterminant dans la réussite de ce projet grâce à ses conseils éclairés et son soutien constant.

Nous exprimons également notre gratitude envers Monsieur MERAOUBI Hamid, Chargé de recherche, Madame BENZABA Wahiba, Ingénieur de recherche conseillère, Madame BOUTALEB Ratiba, Ingénieur de recherche, et Madame BOUTRIF Rania, Master 2 en informatique au CDTA, pour leur précieuse aide et leurs conseils tout au long de notre séjour.

Nous remercions notre co-encadreur, M. YKHLEF Farid, Professeur à l'Université SAAD DAHLAB, pour son précieux soutien et son engagement envers notre réussite dans ce projet.

Nous adressons nos sincères remerciements à tous les professeurs de la spécialité des systèmes embarqués, en particulier à Madame NACEUR Djamilia, Madame BOUGHRIRA Hamida et Madame BOUGHRIRA Nadia, pour leur soutien et leurs encouragements tout au long de notre parcours universitaire.

Enfin, nous tenons à exprimer notre profonde gratitude envers les membres du jury pour avoir consacré une partie de leur précieux temps à discuter et évaluer notre travail.

DEDICACE

Avant tout, je tiens à remercier Dieu, l'Unique, qui m'a offert le courage et la volonté nécessaires pour affronter les différentes difficultés de la vie

En ce jour où je marque la fin d'une étape importante de ma vie académique, je souhaite exprimer ma profonde gratitude à toutes les personnes qui m'ont soutenu et accompagné tout au long de ce parcours.

Tout d'abord, je voudrais remercier du fond du cœur mon père, ma mère, ma tante Salima, ma sœur Imene et mon frère Adem. Leur soutien indéfectible et leurs encouragements constants ont été ma force et ma motivation.

Je suis vraiment reconnaissante d'avoir croisé ton chemin, Mammou Madina. C'était un honneur de partager cette expérience avec toi. Tes efforts ont eu un impact immense, et je suis sincèrement reconnaissante de t'avoir comme binôme.

Je ne saurais oublier de mentionner les personnes qui ont été là pour nous aider dans les moments difficiles. Mes remerciements vont à Mesmoudi Mohammed Chemseddine, Bouchiba Mohammed El Amine et Benhocine Abd Raouf. Leur générosité et leur soutien ont été d'un soutien inestimable.

Enfin, un immense merci à tous mes amis proches qui ont su être présents pour moi, qui m'ont soutenu et encouragé quand j'en avais le plus besoin. Leur présence a été un réconfort précieux tout au long de cette période intense.

À vous tous, je dédie ce mémoire. Vos contributions et votre soutien ont été essentiels à sa réalisation et ont enrichi mon expérience d'étudiant. Je suis profondément reconnaissant pour chaque moment partagé et chaque geste de générosité. Merci du fond du cœur.

Silia

DEDICACE

Avant tout, je tiens à remercier Dieu, l'Unique, qui m'a offert le courage et la volonté nécessaires pour affronter les différentes difficultés de la vie.

Je dédie ce modeste travail à mes très chers parents, pour leurs sacrifices, leur amour, leurs prières et leur soutien. À mes frères. À ma grand-mère, mon grand-père, à ma chère tante Assia, et à toute ma famille Mammou et Gherensi, ainsi qu'à mes amis.

Je tiens à exprimer ma profonde gratitude à ma précieuse binôme, SILIA, pour sa collaboration, son soutien et son dévouement tout au long de ce projet. Ta contribution a été essentielle à la réussite de ce travail, et je suis profondément reconnaissante pour cette expérience enrichissante que nous avons partagée. C'est un véritable honneur de te connaître et de travailler à tes côtés.

Je remercie chaleureusement Mesmoudi Mohammed Chemseddine, Bouchiba Mohammed Amine et Benhocine Abd Raouf pour leur générosité et leur soutien inestimable.

À tous les enseignants qui ont illuminé mon parcours éducatif, et spécialement les enseignants de l'électronique des systèmes embarqués, vous avez été une source d'inspiration et de savoir précieux. Votre dévouement et votre passion ont façonné mon chemin vers la connaissance. Merci pour tout ce que vous faites.

À ma meilleure amie, Malak, à qui je souhaite beaucoup de réussite dans sa vie, et à toutes mes amies.

À Séraphina, compagne fidèle et source de joie constante. Merci pour tout l'amour que tu apportes à ma vie, mon chat.

Enfin, à tous ceux qui m'ont assisté dans la réalisation et le bon déroulement de ce travail, merci infiniment.

Madina

RESUME

ملخص: الهدف من هذا المشروع هو دمج وحدة للتعرف على ستة أوامر صوتية باللهجة الجزائرية (AD) في منصة إدارة مواقف السيارات الحالية. نحن نستخدم نموذج الشبكة العصبية التلافيفية (CNN) الذي تم تدريبه مسبقًا على مجموعة بيانات الأوامر الكلامية المصغرة وتم ضبطه على قاعدة بيانات تضم 6000 ملف صوتي. قاعدة البيانات تم تسجيلها بواسطة متحدث واحد من جنس انثى. نقترح تنقيح النموذج الأولي واستخدام تقنيات زيادة البيانات لإنشاء نموذج خاص بمتحدثين آخرين (وكلاء) من كمية صغيرة من بيانات الكلام. حققت طريقتنا معدل تعرّف بنسبة 99,22%. بالإضافة إلى ذلك، نقترح تحسين وحدة الأجهزة للاتصال مع معدات موقف السيارات، مثل الحاجز وجهاز الإنذار، من خلال اقتراح بنية لاسلكية تعتمد على دائرة ESP32.

الكلمات المفتاحية: التعرف على الصوت، اللهجة الجزائرية، إدارة مواقف السيارات، الكشف عن الصوت الاندفاعي، سي إن إن، ESP32، الضبط الدقيق.

Résumé : Ce projet vise à intégrer un module de reconnaissance de six commandes vocales en Dialecte Algérien (DA) à une plateforme existante de gestion des parcs de stationnement. Nous utilisons un modèle réseau de neurones (CNN) préentraîné sur la mini Speech Commands dataset et affiné (Fine-tuning) sur une base de données de 6000 fichiers audio. Cette base de données est en DA et a été enregistrée par un seul locuteur féminin. Nous proposons d'affiner le modèle initial et d'utiliser des techniques d'augmentation de données pour créer un modèle spécifique à d'autres locuteurs (agents) à partir de peu de données vocales. Notre méthode a atteint un taux de reconnaissance de 99.22%. En outre, nous proposons d'améliorer le module matériel de communication avec les équipements du parking, tels que la barrière et l'alarme, en proposant une connexion sans fil basée sur le circuit ESP32.

Mots clés : Reconnaissance vocale, Dialecte Algérien, Gestion de parkings, Détection des sons impulsifs, CNN, ESP32, Fine-tuning.

Abstract: The aim of this project is to integrate a recognition module for six Algerian Dialect (AD) voice commands into an existing parking management platform. We use a Convolutional Neural Network (CNN) model pre-trained on the mini Speech Commands dataset and fine-tuned on a database of 6000 audio files. This database is in AD and was recorded by a single female speaker. We propose to refine the initial model and use data augmentation techniques to create a model specific to other speakers (agents) from a small set of speech data. Our method achieved a recognition rate of 99.22%. In addition, we propose to improve the hardware module for communication with parking equipment, such as the barrier and the alarm, by proposing a wireless connection based on ESP32 circuit.

Keywords: Voice recognition, Algerian Dialect, parking management, impulsive sound detection, CNN, ESP32, fine-tuning.

LISTE DES ACRONYMES ET ABREVIATIONS

AGV: *Automated Guided Vehicle*

API: *Application Programming Interface*

AVX: *Advanced Vector Extensions*

CNN: Réseau de Neurones Convolutif

DA: Dialecte Algérien

GAN: *Generative Adversarial Network*

GPS: *Global Positioning System*

GPU: *Graphics Processing Unit*

IA: Intelligence Artificielle

IoT: Internet des Objets

NB-IoT: Internet des objets à bande étroite

OCR: reconnaissance optique de caractères

RFID: *Radio Frequency Identification*

TFCT: Transformée de Fourier à court terme

VAD: Détection Vocale d'Activité

Wi-Fi: *Wireless Fidelity*

YOLO: *You Only Look Once*

LISTE DES FIGURES

Figure 1.1: (a) Vue de dessus, (b) Vue en 3D d'un système de stationnement de type puzzle [5]	6
Figure 1.2: Système de stationnement de type empileur [5].....	7
Figure 1.3: Système de stationnement rotatif vertical [5]	8
Figure 1.4: Système de stationnement à circulation horizontale [5]	8
Figure 2.1: Interface graphique de la plateforme de gestion existante [1]	13
Figure 2.2: Méthode globale de la gestion intelligente des parkings par commandes vocales (SI=son impulsif) [1]	14
Figure 2.3: Architecture du model.....	18
Figure 2.4: Schéma de connexion de la barrière, du module relais et de l'ESP32	22
Figure 2.5: Schéma de connexion de la lampe, du module relais et de l'ESP32	23
Figure 2.6: Schéma de connexion de la sirène, du module relais et de l'ESP32	24
Figure 3.1: Jupyter notebook.....	26
Figure 3.2: ESP32	26
Figure 3.3: IDE Arduino	27
Figure 3.4: Matlab.....	27
Figure 3.5: GoldWave	28
Figure 3.6: Wo Mic.....	28
Figure 3.7: Wo Mic sur smartPhone	29
Figure 3.8: Wo Mic Client	29
Figure 3.9: QT Designer	30
Figure 3.10: Enregistrement audio de la commande "طلع"	31
Figure 3.11: Courbe de perte du modèle de reconnaissance initiale pour le DA	33
Figure 3.12: Matrice de Confusion du modèle de reconnaissance initiale pour le DA.....	33
Figure 3.13: Courbe de perte du modèle de reconnaissance pour voix féminine	34
Figure 3.14: Matrice de confusion du modèle de reconnaissance pour voix féminine.....	34
Figure 3.15: Courbe de perte du modèle de Reconnaissance pour Voix Masculine....	35
Figure 3.16: Matrice de confusion du Modèle de Reconnaissance pour Voix Masculine	35
Figure 3.17: Courbe de perte pour modèle de confirmation binaire.....	36
Figure 3.18: Matrice de confusion du modèle de confirmation binaire	36
Figure 3.19: Configuration de l'esp32 dans l'IDE Arduino.....	37
Figure 3.20: Interface des commandes.	38
Figure 3.21: Barrière ON et Barrière OFF.	39
Figure 3.22: Ouverture et fermeture de la barrière avec ESP32.....	39
Figure 3.23: Lampe ON et Lampe OFF.....	40
Figure 3.24: Allumer et éteindre la lampe avec esp32.....	40

Figure 3.25: Alarme ON et Alarme OFF.	40
Figure 3.26: Activation et Désactivation de l'alarme	41
Figure 3.27: Interface de la gestion intelligente du parking en DA.	41
Figure 3.28: Affichage de l'interface.....	42
Figure 3.29: Affichage d'Arduino connecté et non connecté.....	42
Figure 3.30: Liste des commandes.	42
Figure 3.31: Affichage de confirmation.	43
Figure 3.32: Affichage d'ouvrir et fermer la barrière.	44
Figure 3.33: Affichage du message activation et désactivation de l'alarme.....	44
Figure 3.34: Affichage d'allumer et éteindre l'éclairage.	44

TABLE DES MATIERES

Remerciements	(i)
Dédicaces	(ii)
Résumé	(iii)
Liste d'abréviations et acronymes	(iv)
Liste des figures	(v)
Table des matières	(vi)

INTRODUCTION GENERALE1

1 CHAPITRE 1 : GESTION DES PARCS DE STATIONNEMENT4

1.1	Introduction	4
1.2	Importance de la gestion intelligente des parkings de stationnement	4
1.3	Evolution des parcs de stationnement vers des solutions intelligentes	4
1.4	Types de systèmes de stationnement	6
1.5	État de l'art sur la gestion des parkings	9
1.6	Conclusion	11

2 CHAPITRE 2 : PRESENTATION DE LA SOLUTION PROPOSEE12

2.1	Introduction	12
2.2	Reconnaissance des commandes vocales	12
2.3	Communication avec le matériel et équipements du parc	22
2.4	Conclusion	24

3 CHAPITRE 3 : RESULTATS ET INTERPRETATIONS25

3.1	Introduction	25
3.2	Logiciels et langages de programmation utilisés	25
3.3	Description des données expérimentaux	30
3.4	Difficulté du choix du modèle	31
3.5	Problèmes de compatibilité matérielle et choix de la version du modèle	32
3.6	Interprétation des résultats des modèles	32
3.7	Evaluation de la reconnaissance des commandes vocales	37
3.8	Connexion WiFi ESP32 avec Arduino IDE et Python	37
3.9	Implémentation de la solution proposée	38
3.10	Conclusion	45

CONCLUSION GENERALE ET TRAVAUX FUTURE46

REFERENCES.....48

Introduction Générale

Motivations

Dans les environnements urbains d'aujourd'hui, la gestion efficace des parkings revêt une grande importance pour améliorer la fluidité du trafic et optimiser l'utilisation des infrastructures urbaines. À cet égard, les avancées technologiques telles que les capteurs intégrés aux places de stationnement, l'Internet des objets (IoT) et l'intelligence artificielle (IA) offrent des solutions innovantes pour surveiller en temps réel l'occupation des espaces disponibles. Parmi ces technologies émergentes, la reconnaissance vocale se distingue comme un outil efficace pour améliorer la gestion opérationnelle des parkings. En permettant aux gestionnaires et aux agents de sécurité d'interagir intuitivement avec les systèmes de gestion, en cas d'urgence ou de comportements suspects, les agents peuvent déclencher des alarmes instantanées par commandes vocales, renforçant la sécurité du parking et assurant un environnement sûr pour tous les utilisateurs.

Contribution

Dans la gestion du parking, les actions possibles pour un agent incluent : contrôler l'accès à la barrière (ouverture et fermeture), gérer l'éclairage (allumage et extinction), ainsi qu'activer ou désactiver l'alarme.

O. Cherir et A. Mekdour [1] ont conçu un système de gestion de parking intégrant la détection des sons impulsifs et la reconnaissance vocale. Ce système repose sur l'utilisation de deux microphones distincts. Le premier est fixe et enregistre en continu les sons environnants pour détecter les claquements des mains. Cette étape est nécessaire pour interroger la plateforme de gestion du parking et se préparer à recevoir les commandes vocales. Le second est mobile, positionné près des lèvres de l'agent pour enregistrer les commandes vocales. Maintenir une distance appropriée entre ce second microphone et les lèvres de l'agent est crucial pour assurer une reconnaissance vocale précise. Lorsqu'un microphone capte les claquements de mains, cela active le second microphone et déclenche le processus d'enregistrement des commandes vocales par l'agent du parc. La communication avec les équipements du parking, tels que la barrière et l'alarme, s'effectue par le biais d'une carte Arduino.

La plateforme utilisait l'API Google Speech Recognition pour convertir les commandes vocales en texte. Cependant, cette solution présentait plusieurs limitations, notamment une dépendance à une connexion internet constante, une reconnaissance vocale basée uniquement sur l'Arabe Standard, ce qui ne correspond pas à l'usage courant du langage local répandu en Algérie.

Pour remédier à ces limitations, nous avons opté pour un système de reconnaissance des commandes vocales prononcées en Dialecte Algérien (DA). Nous utilisons un réseau de neurones convolutif (CNN) pré-entraîné avec la base de données anglaise mini Speech Commands. Nous avons créé notre propre base de données en DA, comprenant 6 classes " اکتیفي ", " طفي ", " اشعل ", " هبط ", " طلع " et " حبس ". Les données sont enregistrées par un locuteur féminin. La base contient 1000 fichiers dans chaque classe. Nous avons entraîné notre modèle en utilisant des poids initiaux correspondant aux poids finaux du même modèle pré-entraîné sur la base de données anglaise « mini Speech Commands ». Les prédictions de ce modèle servent de commandes.

Nous avons mis en place une étape de post-traitement pour corriger des éventuelles erreurs de reconnaissance. Cette étape consiste à la création d'un modèle de confirmation comportant les classes "oui" ou "non", permettant de confirmer si la prédiction de la commande est correcte ou non.

Cependant, le déploiement du système pour être utilisé par d'autres agents nécessite des enregistrements audio de ces derniers afin de reconnaître leur timbre vocal. Ces enregistrements sont ensuite utilisés pour affiner le modèle de reconnaissance en DA. Toutefois, le processus d'affinement (fine-tuning) du modèle nécessite de nombreux enregistrements audio, ce qui ne pourrait pas être effectué par les agents à chaque installation du système. Pour résoudre cette contrainte, nous avons développé une méthode efficace basée sur l'augmentation de peu de données audio enregistrées à chaque installation du système. L'entraînement du modèle de prédiction est affiné en utilisant les poids du modèle préalablement entraîné sur les 6000 fichiers audio en DA.

La solution que nous avons proposée a permis d'améliorer le système existant en intégrant une étape de reconnaissance des commandes vocales en DA sans nécessiter une connexion internet constante.

En plus de cette mise à jour, nous proposons d'améliorer le module matériel de communication avec les équipements du parking, tels que la barrière, l'éclairage et l'alarme, en proposant une connexion sans fil basée sur le circuit ESP32. Nos tests ont démontré des performances remarquables, notamment une connectivité sans fils fiable, réduisant ainsi considérablement les risques d'électrocution par rapport à l'utilisation de l'Arduino, et réduisant les coûts d'installation du câblage.

Cependant, l'intégration de la carte ESP32 n'a pas pu être réalisée sur la plateforme de gestion en raison de contraintes de temps. Nous avons donc utilisé la solution existante qui est l'Arduino pour maintenir la continuité des opérations.

Impact de projet

En Algérie, ce projet de gestion de parking marque une avancée importante dans le domaine technologique. Il offre une solution innovante pour améliorer la gestion des parkings urbains, ce qui pourrait avoir un impact significatif sur le plan économique et social. Divers secteurs comme les centres commerciaux, les hôpitaux, les résidences, pourraient bénéficier de cette initiative. De plus, notre système est conçu pour s'intégrer facilement aux initiatives de villes intelligentes, soutenant ainsi la modernisation de nos infrastructures urbaines.

Organisation du mémoire :

Le mémoire suit une structure méthodique, articulée comme suit :

Le chapitre 1 fournit une introduction approfondie à la gestion des parkings, en mettant en lumière les différentes approches existantes.

Le chapitre 2 offre un aperçu détaillé de la solution élaborée pour optimiser la gestion des parkings.

Le chapitre 3 se concentre sur la présentation des résultats obtenus suite à la mise en œuvre de la solution, ainsi que leur analyse approfondie.

Ce mémoire se termine par une conclusion générale et perspectives.

Chapitre 1 : Gestion des parcs de stationnement

1.1 Introduction

Dans ce chapitre, nous abordons l'importance de la gestion intelligente des parkings de stationnement en offrant un aperçu de l'évolution vers des solutions de stationnement intelligentes. Nous décrivons les différentes catégories de systèmes de stationnement et citons quelques études portant sur les stratégies innovantes de gestion des parkings, le chapitre se clôt par une conclusion.

1.2 Importance de la gestion intelligente des parkings de stationnement

Nous sommes convaincus que les parkings intelligents sont indispensables pour optimiser l'utilisation des espaces de stationnement, simplifier l'accès pour les conducteurs et améliorer la performance globale [2]. En intégrant des technologies de pointe avec des méthodes de gestion efficaces, ces parkings peuvent offrir une expérience utilisateur fluide tout en utilisant au mieux l'espace disponible. Grâce aux données en temps réel et aux analyses historiques, nous pouvons prendre des décisions éclairées et pertinentes, cela améliore la gestion des espaces et génère des revenus significatifs. En rendant le stationnement plus simple, ces systèmes apportent des avantages aux conducteurs et aux commerçants, augmentant ainsi l'activité urbaine et facilitant la mobilité [2].

1.3 Evolution des parcs de stationnement vers des solutions intelligentes

Le développement de la technologie du stationnement a connu une évolution remarquable marquée par des avancées considérables dans le domaine des systèmes de stationnement intelligents.

Les progrès dans le domaine du stationnement ont vraiment changé la façon dont nous nous déplaçons en milieu urbain. Cela a commencé avec l'installation de capteurs de base dans les parkings, et maintenant, nous utilisons l'intelligence artificielle (IA) pour prédire la disponibilité des places [3].

1.3.1 Capteurs de base

De nouvelles approches ont été initiées pour résoudre les problèmes de stationnements au début des années 2000, ce qui a donné naissance au concept de stationnement intelligent. Les premiers projets consistaient à installer des capteurs de base dans les parkings pour détecter la présence de véhicules [4]. Ces capteurs activaient ensuite des indicateurs, guidant les conducteurs vers les places disponibles [3].

1.3.2 Intégration de données en temps réel

L'intégration de données en temps réel est devenue cruciale dans l'évolution des systèmes de stationnement intelligents. Les téléphones intelligents et le GPS ont ouvert la voie à de nouvelles applications et plateformes qui révolutionnent la manière dont nous trouvons les places de parking. Grâce à ces avancées, les utilisateurs peuvent maintenant obtenir des informations en temps réel sur les disponibilités de stationnement [4], de plus, ces applications facilitent le paiement et rendent le stationnement plus agréable [3].

1.3.3 IoT et réseaux de capteurs

Le concept de stationnement intelligent a été réinventé grâce à l'utilisation de l'Internet des Objets (IoT). Des capteurs connectés dans les parkings et les rues fournissent des données précises sur l'occupation des places. Ces dispositifs détectent les véhicules et surveillent la durée du stationnement [4].

Cette innovation améliore l'expérience de stationnement en offrant des fonctionnalités avancées comme la prédiction de la disponibilité des places. Les paiements mobiles simplifient le processus en éliminant le recours à l'argent liquide [3].

1.3.4 IA Prédictive : Analyse de Données

Les avancées récentes dans la technologie de stationnement intelligent sont remarquables grâce à l'intégration de l'intelligence artificielle et de l'apprentissage automatique. Les entreprises européennes sont en tête de ce mouvement [3]. Ces technologies analysent les données historiques sur le stationnement pour prédire avec précision la disponibilité des places à différentes périodes de la journée. Les conducteurs peuvent ainsi bénéficier en temps réel de ces prédictions, ce qui leur permet de mieux planifier leur stationnement et de passer moins de temps à chercher une place [4].

1.4 Types de systèmes de stationnement

Les systèmes de stationnement utilisent diverses technologies et méthodes pour gérer efficacement le stationnement des véhicules, avec pour objectif principal d'optimiser l'espace disponible et d'améliorer l'efficacité globale.

1.4.1 Système de stationnement modulaire ou de type puzzle

Le parking automatisé de type puzzle (figure 1.1) s'inspire du jeu de puzzle pour enfants. Il fonctionne avec une grille 4*4 contenant un emplacement vide. Pour garer ou retirer une voiture, il faut déplacer les autres voitures pour les placer où on le souhaite. Les principaux éléments de ce système sont les étagères/navettes, un ascenseur pour le mouvement vertical, les AGV et les points d'entrée/sortie. Les étagères/navettes peuvent être mobiles dans les directions X et Y ou fixes. Le principal inconvénient est la gestion complexe du système. Il existe différentes méthodes de récupération dans les étagères mobiles, comme la récupération optimale à double charge et les méthodes de récupération à charge multiple. Certaines stratégies de gestion des étagères incluent la stratégie autonome et celles alimentées par AGV [5].

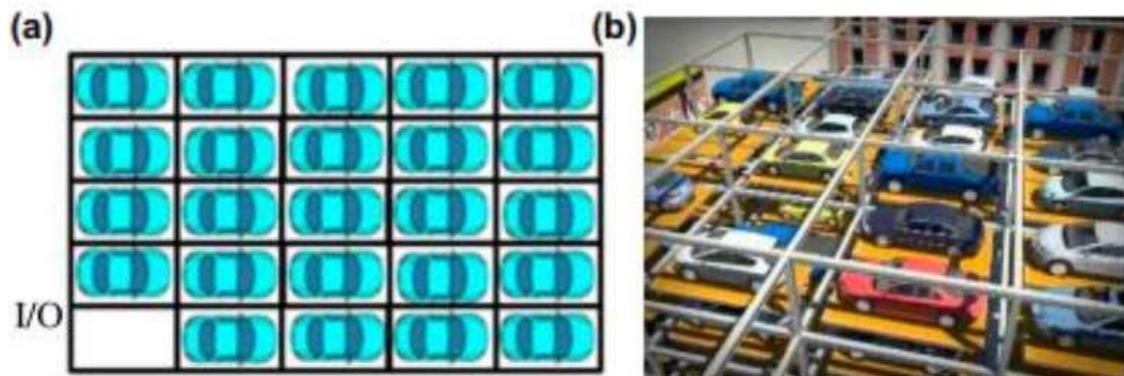


Figure 1.1: (a) Vue de dessus, (b) Vue en 3D d'un système de stationnement de type puzzle [5]

1.4.2 Système de stationnement surélevé ou en tour

Une structure de stationnement en tour ou surélevée se présente sous forme cylindrique. Un élément clé de cette conception est une grue spéciale capable de deux mouvements simultanés : monter et descendre le long de l'axe vertical, ainsi que pivoter dans les deux sens autour de son propre axe. Une navette peut alors se déplacer le long du diamètre de la tour pour atteindre une place de parking de l'autre côté [5].

1.4.3 Système de stationnement à étages

Le parking à étages permet de stationner les véhicules sur plusieurs niveaux à l'intérieur d'un bâtiment, grâce à des rampes et un ascenseur mécanisé. Cette disposition économise de l'espace au sol, augmente la capacité d'accueil et accélère le processus de stationnement. Des capteurs sont intégrés pour optimiser cette procédure, en remplissant d'abord le niveau du sol puis les niveaux supérieurs [5].

1.4.4 Système de stationnement par empilement

Le système de stationnement par empilement (figure 1.2) est une technologie conçue pour déplacer les véhicules de manière efficace à l'intérieur d'un espace restreint. Il utilise un mécanisme centralisé équipé d'un bras robotisé pour déplacer les voitures en les poussant ou en les tirant [5].



Figure 1.2: Système de stationnement de type empileur [5].

1.4.5 Système de stationnement rotatif vertical

Ce système comprend un mécanisme rotatif (figure 1.3) permettant à chaque véhicule de se déplacer dans un mouvement circulaire. Le chargement et le déchargement des véhicules s'effectuent grâce à ce mouvement rotatif. Ce système est optimal pour une flotte comprenant entre 8 et 12 voitures [5].

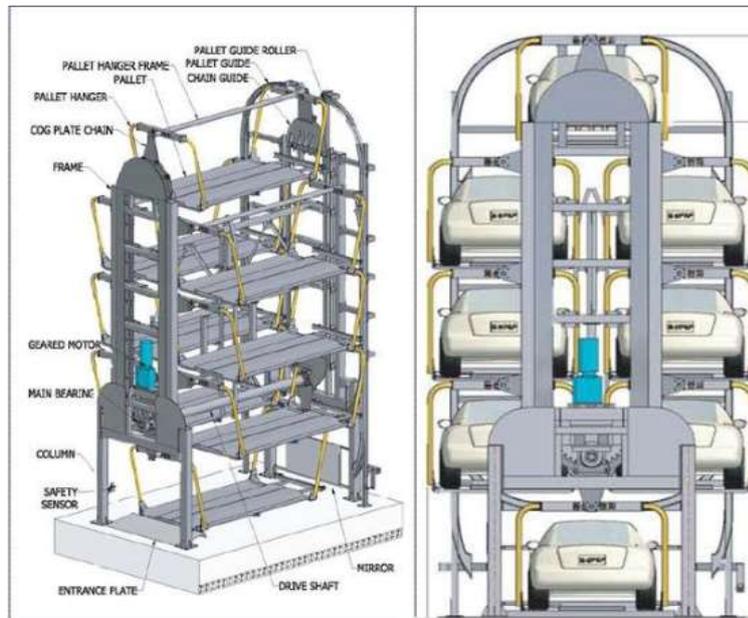


Figure 1.3: Système de stationnement rotatif vertical [5]

1.4.6 Système de stationnement à circulation horizontale

Un mécanisme conçu pour simplifier le stationnement des véhicules dans des espaces restreints en utilisant une approche similaire à un convoyeur à chaîne. Après avoir été garée, la voiture est intégrée dans un circuit de circulation [5] (figure1.4).

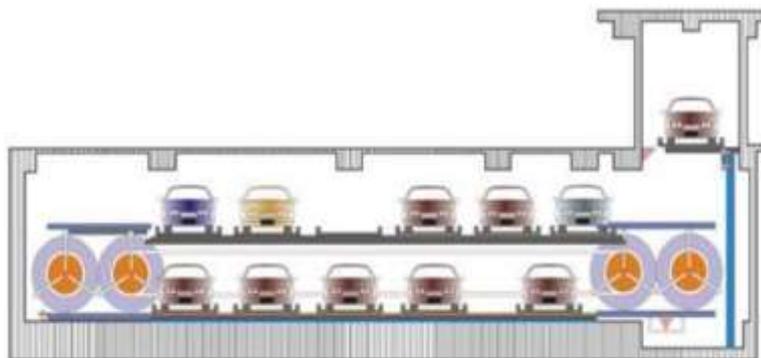


Figure 1.4: Système de stationnement à circulation horizontale [5]

Vu la complexité des systèmes de stationnement, l'agent du parking aura du mal à accomplir ses tâches de manière optimale. Donc, nous pensons que notre solution pourrait apporter de nombreuses améliorations pour la gestion de ces parkings modernes et complexes. En intégrant la reconnaissance des commandes vocales, nous visons à rendre ce système plus simple et efficaces.

1.5 État de l'art sur la gestion des parkings

Nous présentons un état de l'art des méthodes de gestion des parkings, utilisant l'Internet des objets (IoT), les applications mobiles et des capteurs pour optimiser la réservation, la détection des places disponibles et l'utilisation des espaces, avec des algorithmes avancés. Les développements récents intègrent également la reconnaissance optique de caractères (OCR) et l'intelligence artificielle pour une gestion plus performante, incluant la détection d'accidents et le suivi des véhicules.

O. Cherir et A. Mekdour [1] ont développé un système de reconnaissance vocale pour automatiser les actions des agents de parking. Le système détecte en temps réel les commandes vocales pour : ouvrir et fermer la barrière principale, contrôler l'éclairage automatique du parc, et activer ou désactiver l'alarme en cas de vol. La méthodologie inclut la détection des claquements de mains en se basant sur une technique d'identification des sons impulsifs, et l'utilisation de l'API Google pour la reconnaissance vocale, avec le logiciel développé en Python. Arduino est utilisé pour l'exécution des commandes.

J. Cynthia et all [6] ont développé un système de stationnement intelligent basé sur l'Internet des objets (IoT) et une application mobile. Ce système permet aux utilisateurs de réserver des places, de localiser les places libres adaptées à la taille de leur véhicule, de naviguer vers ces places et de recevoir des mises à jour sur leurs comptes de stationnement. Des capteurs infrarouges détectent les places disponibles, les transmettent au serveur via le Wi-Fi et les affichent dans l'application mobile. L'authentification des utilisateurs se fait via des étiquettes RFID sur leur véhicule, leur permettant de réserver des places à différents moments. Un algorithme identifie la place libre la plus proche en fonction de la taille du véhicule. Les propriétaires d'espaces de stationnement peuvent obtenir des analyses d'occupation et de revenus pour ajuster les tarifs. L'application mobile offre une expérience utilisateur conviviale.

Yanfeng Geng et all [7] ont développé un système de stationnement intelligent pour les villes. Ce système aide les conducteurs à trouver rapidement une place de stationnement proche de leur destination tout en minimisant les coûts. En utilisant des méthodes mathématiques avancées, ils ont créé un algorithme qui tient compte des informations en temps réel et des événements aléatoires, garantissant une utilisation optimale des

places disponibles. Leur innovation promet des économies de temps et d'argent pour les conducteurs, ainsi qu'une réduction de la congestion routière et de la pollution. Une étude pilote à l'Université de Boston a validé leur approche.

Jiong Shi et al [8] ont abordé le problème croissant de la congestion du trafic urbain en proposant un système novateur de stationnement intelligent. Ce système utilise la technologie Internet des objets à bande étroite (NB-IoT) pour transmettre les données des capteurs, permettant ainsi une surveillance efficace des places de stationnement. Les chercheurs ont mis en place une infrastructure sur un serveur cloud pour gérer les informations de base, surveiller les capteurs, gérer les charges et les tâches, ainsi que pour intégrer des modules d'intelligence d'affaires.

L'une des caractéristiques importantes de leur système est l'application mobile développée pour les conducteurs. Cette application offre un service de guidage de stationnement et une plateforme de paiement tierce intégrée, rendant ainsi le processus de stationnement plus facile et plus pratique pour les utilisateurs.

Le système a déjà été déployé avec succès dans deux villes, où il a permis une meilleure utilisation des installations de stationnement existantes.

In Hwan Jung et al [9] ont développé un système intelligent pour gérer les parkings en utilisant plusieurs caméras et une technique d'intelligence artificielle. Leur système identifie les véhicules à l'aide d'une caméra intégrée, détermine leur place de parking et met à jour les informations associées. De plus, ils ont mis en place un système de détection d'accidents en utilisant des caméras de surveillance. Ils ont utilisé la reconnaissance optique de caractères (OCR) sur un système Raspberry Pi pour reconnaître les numéros de plaque d'immatriculation des véhicules. En suivant ces numéros d'identification à l'entrée, ils ont pu suivre efficacement les véhicules à l'intérieur du parking et déterminer leur emplacement. Pour la détection des accidents de collision, ils ont utilisé une méthode appelée YOLO avec un processus d'apprentissage en profondeur. Ils ont entraîné leur système avec plus de 500 images de collisions potentielles, montrant que la précision de la détection s'améliore avec le nombre d'images d'entraînement. Leur système permet de gérer efficacement les emplacements de stationnement, les informations sur les places disponibles et les accidents éventuels. En utilisant un système en nuage, ils peuvent fournir aux conducteurs des informations intégrées sur les parkings dans de vastes zones géographiques.

1.6 Conclusion

Dans ce chapitre, nous avons exploré l'importance de la gestion intelligente des parkings, en examinant l'évolution historique des parcs de stationnement et en identifiant les différents types de systèmes utilisés. Dans cette optique, nous envisageons de concevoir un système de gestion de stationnement intelligent qui répondrait aux besoins des agents en utilisant des commandes vocales en dialecte algérien pour faciliter leurs tâches.

Chapitre 2 : Présentation de la solution proposée

2.1 Introduction

Ce chapitre explore notre approche complète pour la gestion du parking. Nous examinons en profondeur la solution que nous avons élaborée pour reconnaître les commandes vocales et exécuter les tâches requises. Pour finir, nous terminerons ce chapitre par une conclusion.

2.2 Reconnaissance des commandes vocales

Cette section couvre plusieurs aspects essentiels de la reconnaissance des commandes vocales : un aperçu global des concepts et objectifs principaux, la création de la base de données, la méthode de *Data Augmentation* pour enrichir les données, la définition d'un réseau de neurones à convolution (CNN), la présentation du modèle de reconnaissance vocale, et une méthodologie d'adaptation du modèle en DA.

2.2.1 Aperçu globale

Ce système existant, conçu par O. Cherir et A. Mekdour [1], offre une gestion complète et intelligente des parkings, permettant à l'agent d'exécuter plusieurs actions cruciales avec facilité. Parmi ces actions figurent l'ouverture et la fermeture de la barrière, l'allumage et l'extinction de l'éclairage, ainsi que l'activation et la désactivation de l'alarme.

Le système repose sur la détection des sons impulsifs à l'aide d'un microphone fixe, qui active un microphone mobile pour enregistrer les commandes vocales de l'agent permettant à ce dernier de prononcer sa commande vocale. Cette commande est ensuite convertie en texte par l'API Google Speech Recognition, puis envoyée à une carte Arduino pour exécution.

Les auteurs du travail ont intégré l'API Google Speech Recognition et l'Arduino dans leur algorithme, et ont également développé une interface graphique pour faciliter l'interaction avec le système (voir la figure 2.1).



Figure 2.1: Interface graphique de la plateforme de gestion existante [1]

Cependant, cette solution présente certaines failles. Elle utilise l'API Google Speech Recognition en Arabe Standard, nécessitant une connexion internet, et une connexion filaire entre le matériel et l'Arduino, ce qui est un inconvénient.

Pour pallier ces limitations, nous proposons un modèle de reconnaissance vocale en DA basé sur un réseau CNN, sans nécessiter de connexion internet. De plus, l'intégration de la carte ESP32 pour une connexion sans fil assure une meilleure sécurité (contre les hautes tensions des équipements) et facilite l'installation des équipements.

Pour garantir une interaction fluide, nous utiliserons la même approche que celle utilisée par O. Cherir et A. Mekdour [1]. L'agent énonce ensuite une commande vocale qui sera convertie en spectrogramme et reconnue en utilisant un modèle (CNN) pré-entraîné avec la base de données mini Speech Commands. En intégrant les poids finaux de ce modèle pré-entraîné comme poids initiaux dans notre propre modèle, entraîné avec notre base de données en DA, nous facilitons le processus d'apprentissage et assurons une convergence rapide.

Pour améliorer la sécurité et éviter les erreurs de prédiction potentiellement dangereuses dans notre système de gestion de parking, nous avons décidé d'incorporer un deuxième modèle de confirmation. Ce modèle intervient après que le modèle principal a fait une prédiction basée sur la commande reçue. Par exemple, si le modèle principal prédit "ouvrir la barrière", le modèle de validation demande "Êtes-vous sûr que vous voulez

ouvrir la barrière ?", avec les options "oui" ou "non". Cette approche garantit que des actions potentiellement risquées renforçant ainsi la sécurité et la fiabilité du système. Le modèle prédit avec précision la commande vocale la plus probable, en déterminant quelle action du système de gestion de parking correspond le mieux à la commande vocale donnée. Une fois la prédiction est générée et validée, elle est utilisée pour associer la commande vocale à l'action correspondante dans le programme du système. Cette action est ensuite transmise à la carte d'interface pour une exécution immédiate et précise, assurant ainsi une gestion efficace du parking en réponse aux commandes vocales de l'agent. Ceci permet une communication rapide et fiable entre le système de reconnaissance vocale et les dispositifs matériels du parking, tels que la barrière, l'éclairage et l'alarme.

La figure 2.2 représente la méthode globale de la gestion intelligente des parkings par commandes vocales.

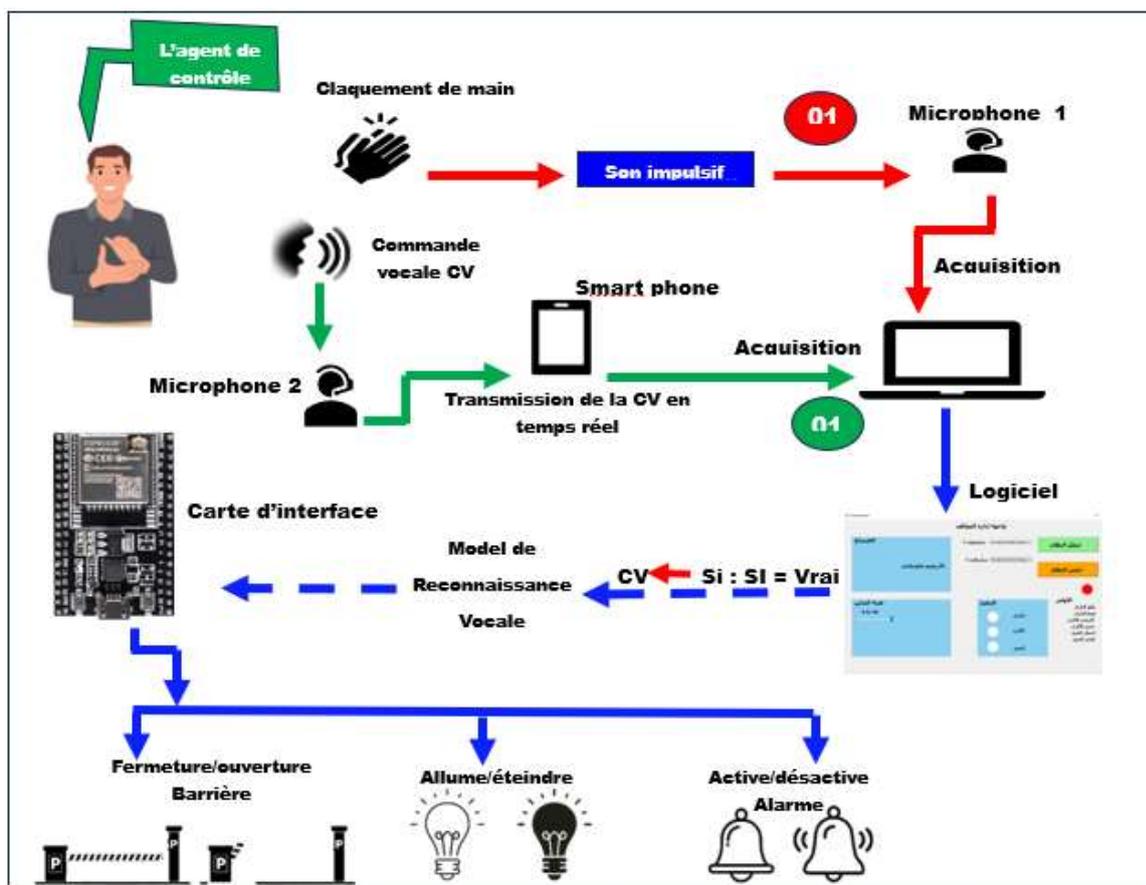


Figure 2.2: Méthode globale de la gestion intelligente des parkings par commandes vocales (SI=son impulsif) [1]

2.2.2 Création de la base de données

Dans notre projet de développement d'un modèle de reconnaissance vocale pour un parking de stationnement intelligent, nous avons pris des mesures spécifiques pour garantir que le système puisse reconnaître spécifiquement l'agent de contrôle du parking. À cet effet, nous avons délibérément opté pour enregistrer une base de données initiale en DA, prononcé par un seul locuteur féminin, celle-ci comporte six classes : "طلع", "هبط", "اشعل", "طفي", "اكتيفي" et "حبس", et chaque classe comporte 1000 audio.

Le choix des classes a été fait de telle sorte qu'elles soient phonétiquement différentes pour éviter toute confusion. Pour réduire encore plus la confusion, nous avons opté pour n'enregistrer que "طلع" au lieu de "طلع الباريار", car si nous utilisons "طلع الباريار" et "هبط الباريار", le modèle pourrait confondre avec le mot "الباريار". C'est pourquoi nous avons choisi comme commande juste le premier mot. De plus, dans notre système, nous avons fait en sorte que le micro capte uniquement les trois premières secondes. Ainsi, même si l'agent dit "طلع الباريار", le système ne captera que "طلع".

Cette décision stratégique était motivée par plusieurs raisons importantes. Tout d'abord, en enregistrant la base de données initiale, nous avons pu garantir la séparabilité et la fiabilité des classes. Cette approche nous a permis de maintenir une cohérence dans la prononciation et l'intonation de chaque commande vocale, renforçant ainsi la capacité du modèle à reconnaître efficacement les commandes vocales de l'agent de contrôle du parking.

De plus, cette approche nous a fourni une base solide avant de passer à l'étape de *Data Augmentation*. En établissant une base de données de haute qualité dès le départ, nous avons pu évaluer la performance initiale du modèle.

2.2.3 Méthode de Data Augmentation

Nous avons été confrontés à un défi majeur lors du développement de notre projet : la collecte des données vocales nécessaires. Demander à l'agent du parking d'enregistrer 6000 échantillons vocaux aurait été à la fois difficile et peu pratique. Pour surmonter cet obstacle, nous avons décidé d'utiliser la Data Augmentation.

Nous avons exploré diverses approches innovantes. Initialement, nous avons tenté d'utiliser un GAN audio via MATLAB. Cependant, cette méthode requérait une puissance

de calcul élevée, notamment un GPU performant, que nous ne possédions pas. Face à cette contrainte, nous avons exploré quatre autres GAN disponibles sur des plateformes comme GitHub. Malheureusement, chaque tentative a été entravée par des problèmes de compatibilité de version, ce qui a entraîné des pertes de temps et d'efforts considérables.

Par conséquent, nous avons développé une méthode alternative de Data Augmentation :

- Pour chaque commande vocale, enregistrement de 10 échantillons distincts.
- Concaténation de ces échantillons pour créer un fichier unique de 1000 échantillons.
- Enregistrement de ce fichier sur un autre appareil dans divers environnements acoustiques pour capturer une variété de conditions sonores.
- Utilisation de MATLAB pour découper ce fichier en 1000 échantillons individuels.
- La méthode a été appliquée à toutes les commandes

Nous avons créé deux bases de données en utilisant la méthode de data augmentation pour deux nouveaux individus, un masculin et un féminin. Cela nous a permis d'augmenter et de diversifier efficacement nos bases de données audio. Ainsi, même avec un nombre limité d'échantillons au départ, nous avons pu enrichir notre jeu de données de manière réaliste et significative.

Notre base de données initiale a permis d'entraîner notre modèle, fournissant ainsi des poids finaux utilisés comme point de départ pour l'entraînement ultérieur avec les bases de données augmentées. Cela permet d'accélérer ainsi le processus d'apprentissage avec les bases de données augmentées et peut aider à prévenir le surapprentissage, favorisant une généralisation efficace à de nouveaux échantillons.

2.2.4 Réseaux de Neurones à Convolution

Un réseau de neurones à convolution (CNN) est un type de réseau de neurones profond spécialement conçu pour traiter les données disposées en grille, telles que les images. Il se compose de couches de convolution qui appliquent des filtres pour détecter des motifs locaux (comme les bords et les textures), suivies de couches de pooling qui réduisent la dimensionnalité des données tout en conservant les informations essentielles. Les couches finales sont entièrement connectées, permettant au réseau de classifier ou de prédire les données traitées, les CNNs sont particulièrement efficaces pour des tâches de

reconnaissance d'images, de détection d'objets et d'analyse vidéo, car ils apprennent automatiquement à extraire des caractéristiques pertinentes directement à partir des données brutes, éliminant ainsi le besoin d'une extraction manuelle des caractéristiques [10].

2.2.5 Présentation du modèle de reconnaissance

Le modèle de reconnaissance vocale est un réseau de neurones convolutif (CNN) pré-entraîné avec la base de données mini Speech Commands, une version réduite de la base de données Speech Commands. Ce modèle transforme les enregistrements audio en spectrogrammes, qui sont des représentations visuelles montrant l'évolution des fréquences sonores dans le temps. En analysant ces spectrogrammes, le modèle identifie et classe les différentes commandes vocales telles que "stop", "left", "yes", etc [11].

Cette technologie permet une interaction intuitive et efficace avec les systèmes, notamment dans des domaines comme la gestion automatisée des parkings .

2.2.6 Fonctionnement du modèle

Préparation des données : Les données audio sont converties en spectrogrammes en utilisant la Transformée de Fourier à court terme (TFCT).

Prétraitement : Les spectrogrammes sont normalisés et mis à l'échelle pour faciliter l'entraînement du modèle.

Modélisation avec CNN : Les spectrogrammes sont introduits dans un CNN, une architecture spécifique qui permet d'extraire des caractéristiques pertinentes des données audio.

Entraînement et validation : Le CNN est entraîné sur un ensemble de données où il apprend à associer des spectrogrammes à des étiquettes de commandes vocales correspondantes. Pendant l'entraînement, l'ensemble de validation est utilisé pour surveiller la performance du modèle et ajuster les paramètres afin d'éviter le surajustement.

Évaluation et prédiction : Une fois l'entraînement terminé, le modèle est évalué sur un ensemble de test distinct pour évaluer sa performance sur de nouvelles données. Enfin, le modèle est utilisé pour faire des prédictions sur de nouvelles données audio, classant ainsi les commandes vocales avec une précision accrue.

2.2.7 Architecture du modèle CNN

Le modèle comporte les couches suivantes (figure 2.3) :

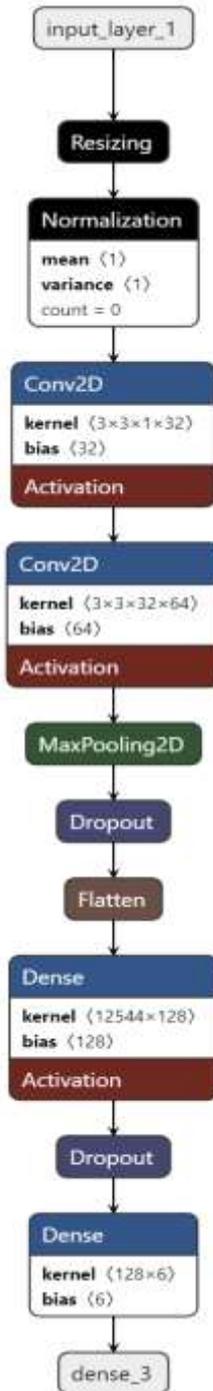


Figure 2.3: Architecture du model

Couche d'Entrée (Input Layer)

Définir la forme des données d'entrée pour le modèle.

Assure que les données reçues par le modèle ont une forme cohérente, facilitant le traitement et la compatibilité avec les couches suivantes.

Couche de Redimensionnement : `layers.Resizing(32, 32)`

Redimensionne les spectrogrammes pour les adapter à une taille standard de 32x32 pixels. Elle facilite le traitement ultérieur par les couches convolutives en garantissant des dimensions d'entrée cohérentes.

Couche de Normalisation : Normalization

Normalise les valeurs des spectrogrammes pour les stabiliser autour d'une moyenne et d'une variance définies. Elle accélère et stabilise l'entraînement du modèle en réduisant la variance des données d'entrée.

Première Couche Convulsive : `layers.Conv2D(32, 3, activation='relu')`

Applique 32 filtres convolutifs de taille 3x3 pour détecter des motifs de bas niveau dans les images, tels que les bords et les textures. La convolution capture les caractéristiques locales des images, et l'activation ReLU (Rectified Linear Unit) introduit de la non-linéarité dans le modèle.

Deuxième Couche Convulsive : `layers.Conv2D(64, 3, activation='relu')`

Applique 64 filtres convolutifs de taille 3x3 pour détecter des motifs plus complexes. En augmentant le nombre de filtres, cette couche permet au modèle de capturer des caractéristiques plus riches et complexes des images.

Couche de Max-Pooling : `layers.MaxPooling2D()`

Réduit la dimension spatiale des données (downsampling) en conservant les caractéristiques les plus importantes (par exemple, la valeur maximale de chaque patch). Le pooling réduit la taille des représentations intermédiaires, diminue le nombre de paramètres et contrôle le surapprentissage tout en préservant les caractéristiques importantes.

Première Couche de Dropout : `layers.Dropout(0.25)`

Désactive aléatoirement 25 % des neurones pendant l'entraînement. Le dropout aide à prévenir le surapprentissage en introduisant de la régularisation dans le modèle, ce qui le rend plus robuste.

Couche de Flatten : `layers.Flatten()`

Convertit la matrice 2D en un vecteur 1D. Elle prépare les données pour les couches denses (fully connected) suivantes en linéarisant les caractéristiques extraites.

Couche Dense : `layers.Dense(128, activation='relu')`

Applique une transformation linéaire suivie d'une activation ReLU, avec 128 unités neuronales.

La couche dense permet de combiner les caractéristiques de manière globale et de modéliser des relations complexes. L'activation ReLU introduit de la non-linéarité.

Deuxième Couche de Dropout : `layers.Dropout(0.5)`

Désactive aléatoirement 50 % des neurones pendant l'entraînement.

Couche de Sortie : `layers.Dense(num_labels)`

Applique une transformation linéaire pour obtenir les scores pour chaque classe. Elle génère les logits (scores non normalisés) pour chaque classe de l'ensemble de données. Le nombre de neurones correspond à `num_labels`, c'est-à-dire le nombre de classes à prédire.

2.2.8 Méthodologie d'adaptation du modèle en Dialecte Algérien

Pour adapter notre modèle à notre propre ensemble de données en DA, nous avons travaillé avec la base de données Mini Speech Commands qui comprend à l'origine 8 classes. Afin de garantir son alignement avec notre ensemble de données spécifique, nous avons choisi de supprimer 2 classes de Mini Speech Commands. Le modèle divise les données en ensembles d'entraînement (80%), de validation (10%) et de test (10%). Il utilise également le *early stopping*, une technique qui arrête l'entraînement dès que la performance du modèle cesse de s'améliorer sur l'ensemble de validation, prévenant ainsi le *surapprentissage* et améliorant la généralisation. Après l'entraînement, nous avons sauvegardé le modèle au format H5 et l'avons chargé en utilisant ces poids comme poids initiaux pour son entraînement avec notre base de données en DA.

En utilisant les poids finaux d'un modèle pré-entraîné comme poids initiaux dans notre modèle entraîné avec notre base de données initiale, nous avons pu bénéficier d'une meilleure adaptation aux caractéristiques spécifiques de notre tâche. Cela a également permis une convergence plus rapide lors de l'entraînement avec notre ensemble de données, en tirant parti des connaissances générales encapsulées dans les poids finaux du modèle pré-entraîné.

Nous avons entraîné deux modèles : l'un avec une base de données pour un individu masculin et l'autre pour un individu féminin. Pour démarrer l'entraînement de chaque modèle, nous avons utilisé les poids finaux du modèle préalablement entraîné avec les données en DA comme points de départ, facilitant ainsi le processus d'apprentissage et assurant une convergence rapide.

Après avoir développé notre modèle principal pour prédire les commandes de gestion du parking en DA, nous avons introduit un modèle de confirmation utilisant une classification binaire afin de prévenir les erreurs critiques de prédiction. Nous avons constitué une base de données spécifique pour ce modèle en utilisant la data augmentation. Pour son entraînement, nous avons initialisé ce modèle avec les poids finaux du modèle principal. Initialement conçu pour traiter 6 classes, nous avons restreint le modèle à 2 classes en supprimant 4 classes. Ensuite, nous avons entraîné ce modèle. Ces poids ont ensuite été utilisés comme initialisation pour le modèle de confirmation.

Ce modèle de confirmation intervient immédiatement après que le modèle principal a formulé sa prédiction. Il demande à l'utilisateur de confirmer la commande proposée, par exemple en posant la question : " Êtes-vous sûr que vous voulez ouvrir la barrière ?", nécessitant une réponse "oui" ou "non" de l'agent. Si la réponse est "oui", la commande est exécutée sinon, la commande est annulée et le modèle principal demande à l'agent de redonner la commande correcte. Cette étape de confirmation supplémentaire permet de prévenir les accidents, comme dans le cas où une commande erronée "ferme la barrière" pourrait être exécutée par erreur alors qu'une voiture passe sous la barrière. En ajoutant cette fonctionnalité, nous augmentons la sécurité et la fiabilité du système tout en maintenant une utilisation simple et intuitive grâce aux réponses binaires.

En parallèle, nous avons intégré la Détection Vocale d'Activité (VAD) dans notre modèle de reconnaissance de commandes vocales. La VAD identifie les périodes de parole dans le flux audio, filtrant ainsi le bruit de fond et les sons non pertinents captés par le microphone. Cela garantit que seules les commandes vocales clairement prononcées sont traitées, améliorant ainsi la précision et la fiabilité de notre système. Cette fonctionnalité est essentielle pour minimiser les erreurs d'interprétation.

2.3 Communication avec le matériel et équipements du parc

Dans notre démarche de conception, nous avons exploré l'utilisation de l'ESP32. Après des tests concluants démontrant leur fiabilité dans un contexte sans fil, nous avons décidé d'opter pour ce composant dans notre solution.

L'agent dispose désormais d'une solution simple et efficace pour commander à distance plusieurs opérations via des commandes vocales. Il peut ainsi contrôler la barrière principale, l'éclairage du parc et l'alarme en toute simplicité, ce qui facilite grandement la gestion sécurisée du parc de stationnement.

2.3.1 Barriere

Nous avons choisi un module relais 5V pour contrôler la barrière en raison de sa capacité à commuter des charges électriques avec un signal de commande à faible tension. Pour intégrer ce module à l'ESP32, nous avons relié ses entrées (GND, 5V, IN) aux ports correspondants. En particulier, l'entrée IN (INPUT) du relais est connectée à la broche 4 de l'esp32, et les sorties du relais (COM, NF) sont reliées à la barrière (voir la figure 2.4).

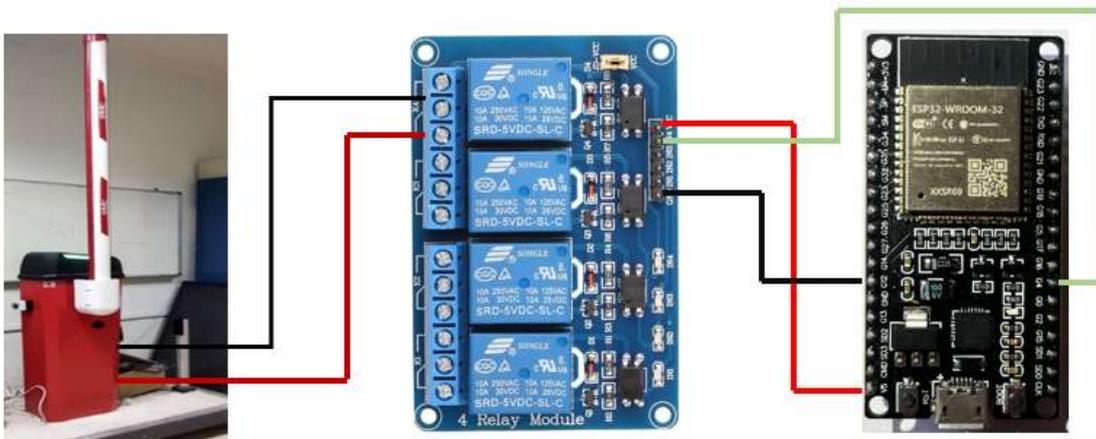


Figure 2.4: Schéma de connexion de la barrière, du module relais et de l'ESP32

2.3.2 Eclairage

Nous avons opté pour une lampe fonctionnant sous une tension de 220V. Pour cette réalisation, nous aurons besoin d'un relais pour monter en toute sécurité une lampe avec une carte ESP32. Nous devons commencer par déconnecter la lampe de toute source d'alimentation électrique. Ensuite, on connecte le relais à l'ESP32 en reliant les broches

GND et 5V du relais aux broches correspondantes de l'ESP32, et la broche IN à la broche 16. Pour brancher la lampe, on connecte la phase de l'alimentation secteur (220V) à la borne commune (COM) du relais et la phase de la lampe à la borne normalement fermé (NF). On s'assure que toutes les connexions sont bien fixées et que les fils ne sont pas exposés. Une fois le câblage terminé, on branche la lampe au secteur et on teste le montage en utilisant l'ESP32 pour contrôler l'allumage et l'extinction de la lampe (voir la figure 2.5)

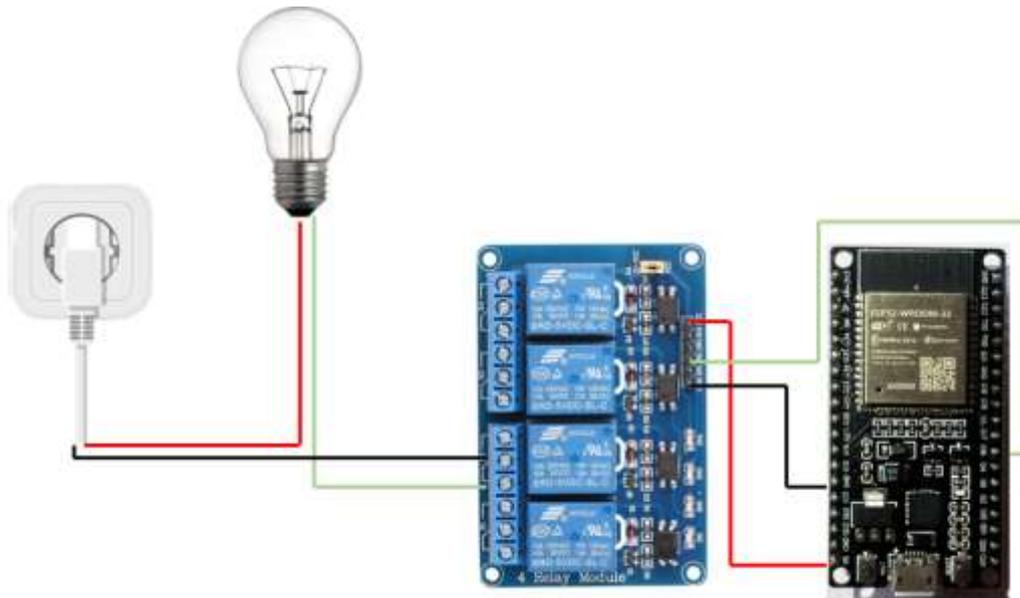


Figure 2.5: Schéma de connexion de la lampe, du module relais et de l'ESP32

2.3.3 Sirène

Pour connecter une sirène à une carte ESP32, nous utilisons un relais pour contrôler son activation. Nous relierons les broches GND et 5V du relais aux broches correspondantes de l'ESP32, et la broche IN à la broche 2. Ensuite, nous connectons la phase de l'alimentation secteur (220V) à la borne commune (COM) du relais et la phase de la sirène à la borne NF du relais (voir la figure 2.6).

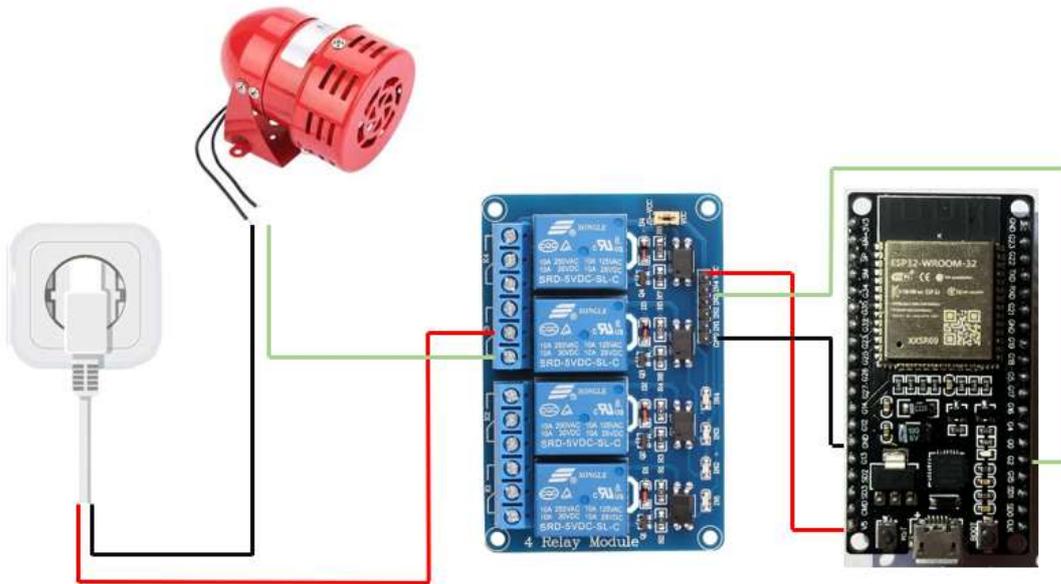


Figure 2.6: Schéma de connexion de la sirène, du module relais et de l'ESP32

2.4 Conclusion

Dans ce chapitre, nous avons présenté notre solution pour une gestion optimisée des parkings, incluant un système de reconnaissance vocale en DA, des améliorations matérielles telles que le remplacement du buzzer par une sirène pour l'alarme, la mise à niveau de l'éclairage à 220V, ainsi que le remplacement de l'Arduino par l'ESP32. Dans le prochain chapitre, nous examinerons les résultats expérimentaux pour évaluer l'efficacité de notre solution.

Chapitre 3 : Résultats et interprétations

3.1 Introduction

Dans ce chapitre, nous exposerons les résultats issus de l'application de la solution pour la gestion intelligente des parkings. Nous avons abordé description des données expérimentaux, la difficulté de choisir le modèle de reconnaissance, les problèmes de compatibilité et de choix de la version ainsi que l'interprétation des résultats. Nous aborderons trois aspects essentiels : les logiciels et langages de programmation utilisés, la connexion WiFi ESP32 avec Arduino IDE et Python, et la mise en œuvre de la solution. Enfin, nous concluons ce chapitre par une conclusion.

3.2 Logiciels et langages de programmation utilisés

Pour notre projet de gestion intelligente des parkings, nous utiliserons l'interface conçu par O. Cherir et A. Mekdour [1] avec Qt Designer, tout en apportant des modifications nécessaires. Python avec Jupyter Notebook a été choisi pour sa programmation interactive, et nous avons également utilisé WO Mic pour la collecte de données sonores via un smartphone. En complément, GoldWave et MATLAB ont été utilisés pour créer la base de données sonores. Nous avons utilisé l'ESP32 pour la gestion des composants électroniques.

3.2.1 Langage python

Python, créé par Guido van Rossum [12] en 1991, est un langage de programmation open source de haut niveau. Il se distingue par sa syntaxe claire, sa lisibilité et une large gamme de bibliothèques, ce qui en fait un outil très polyvalent. On l'emploie notamment dans les domaines du développement web, de l'analyse de données, de l'intelligence artificielle et de l'automatisation. Python utilise un interpréteur pour exécuter le code, ce qui accélère le processus de développement.

3.2.2 Editeur de programmation

Pour notre projet, nous avons choisi d'utiliser Jupyter Notebook (figure 3.1), un environnement de programmation interactif largement reconnu. Jupyter Notebook

permet l'écriture, l'exécution et la documentation efficace du code, avec des fonctionnalités avancées de visualisation des données via Matplotlib et Plotly. Il offre un support pour plusieurs langages, notamment Python, ce qui en fait un outil privilégié pour l'analyse exploratoire et la présentation des résultats. Sa flexibilité et son interactivité, notamment avec l'exécution de code par blocs, facilitent les processus de développement.

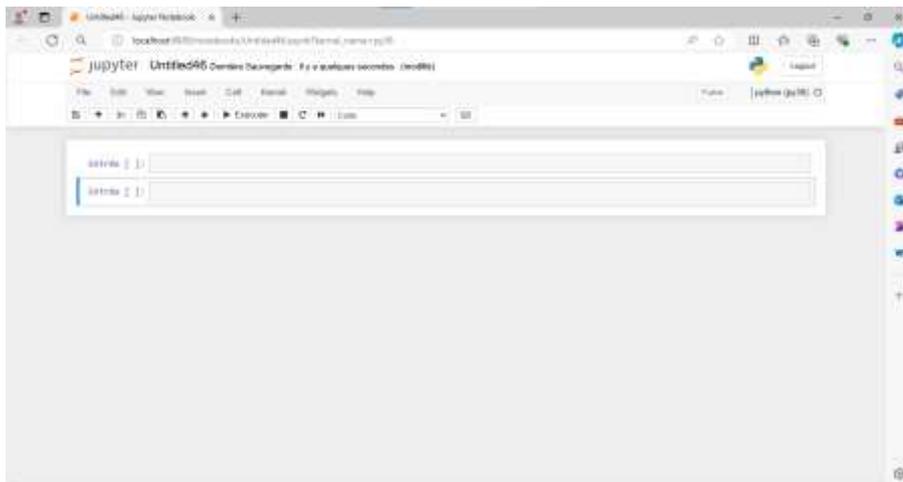


Figure 3.1: Jupyter notebook

3.2.3 ESP32

L'ESP32 est un puissant microcontrôleur SoC (System on Chip) développé par Espressif Systems (figure 3.2). Il intègre des fonctionnalités avancées telles que le Wi-Fi 802.11 b/g/n, le Bluetooth bimode version 4.2, ainsi qu'une variété de périphériques. Succédant à la puce 8266, il se distingue par ses deux cœurs cadencés jusqu'à 240 MHz et une mémoire flash de 4MB [13].



Figure 3.2: ESP32

3.2.4 IDE Arduino

L'IDE Arduino est un environnement de développement intégré open source et gratuit (figure 3.3). Il permet d'écrire des programmes, appelés croquis, en langage C ou C++ simplifié, spécifiquement conçus pour les microcontrôleurs comme l'ESP32.

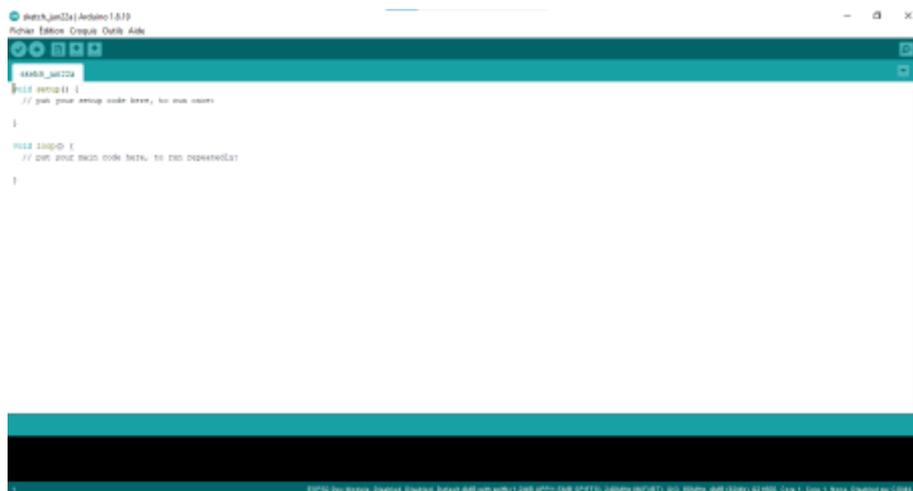


Figure 3.3: IDE Arduino

3.2.5 MATLAB

MATLAB est un environnement de calcul numérique et de programmation très populaire parmi les ingénieurs et les scientifiques (figure 3.4). Il offre des outils spécialisés pour l'analyse de données, la création d'algorithmes et la modélisation. Avec son interface de bureau optimisée pour les processus itératifs, MATLAB emploie un langage de programmation avancé permettant de traiter directement les mathématiques des matrices et des tableaux [14].

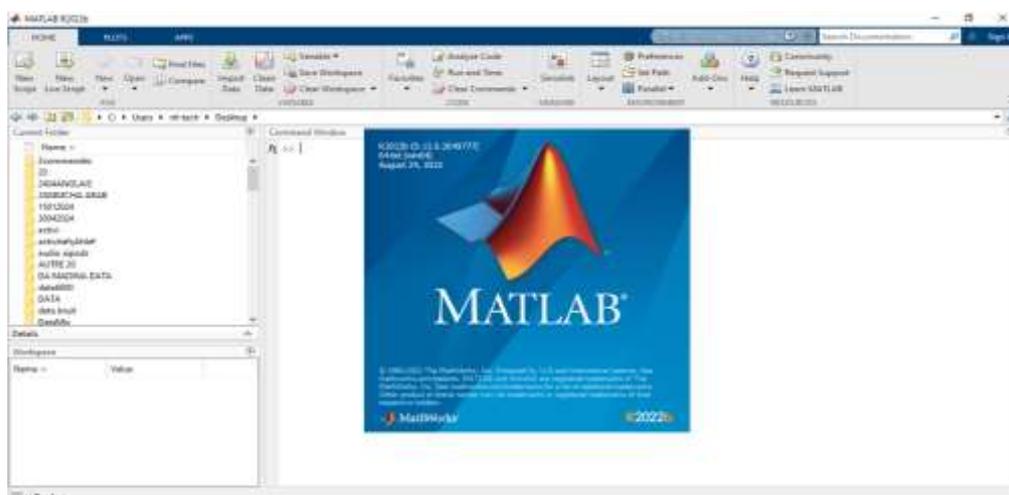


Figure 3.4: Matlab

3.2.6 GoldWave

GoldWave est un éditeur audio numérique reconnu pour sa simplicité et son efficacité (figure 3.5). Il permet d'enregistrer, de restaurer, d'améliorer et de convertir des fichiers audio en enregistrant des sons ou en extrayant l'audio de vidéos. Il prend en charge la lecture et l'écriture de fichiers audio dans différents formats tels que .WAV, .MP3, .WMA, .AIFF, etc. Ses fonctionnalités d'édition incluent le découpage, le collage, le mixage et le fondu enchaîné, ainsi que divers effets audio comme la réduction du bruit, l'ajustement du volume [15].

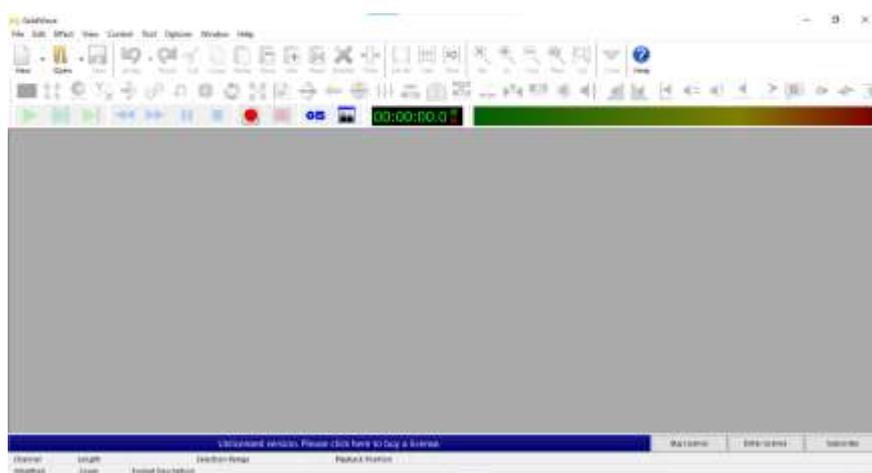


Figure 3.5: GoldWave

3.2.7 Wo Mic

Wo Mic (voir figure 3.6) est une application gratuite qui permet de convertir un smartphone Android en microphone pour ordinateur (figure 3.6). Développé par Beijing Wolicheng Technology Co., Ltd., il est simple à installer et fonctionne sous Windows. Il supporte des connexions Bluetooth, sans fil ou par câble USB, et est compatible avec tous les logiciels nécessitant un micro [16].



Figure 3.6: Wo Mic

Tout d'abord, il est nécessaire d'installer le client Wo Mic ainsi que le pilote correspondant sur notre ordinateur. Ensuite, sur notre smartphone, nous devons télécharger et installer l'application Wo Mic. Les deux appareils sont connectés au même réseau Wi-Fi.

Une fois l'application mobile lancée, on choisit l'option « Wi-Fi » dans les paramètres de connexion et on obtient l'adresse IP nécessaire (figure 3.7).

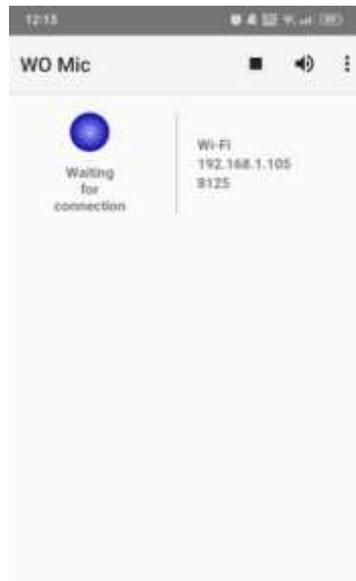


Figure 3.7: Wo Mic sur smartPhone

Enfin, sur le client PC, on sélectionne également la méthode de connexion « Wi-Fi » et on choisit l'adresse IP fournie par l'application mobile. On clique sur « Connecter » pour établir la connexion (figure 3.8).

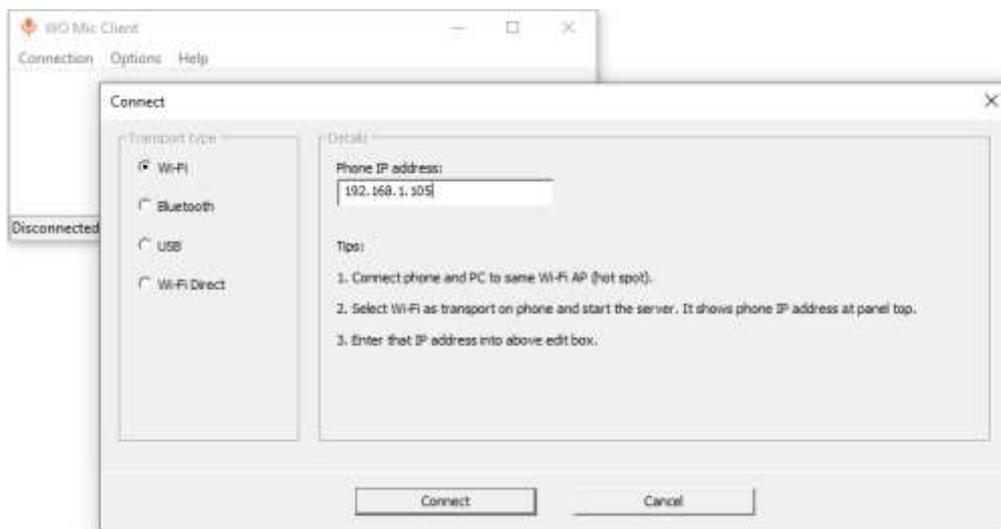


Figure 3.8: Wo Mic Client

Caractéristiques des audio enregistrés :

- **Durée** : Chaque échantillon audio a une durée d'une seconde, suffisante pour capturer la commande spécifique.
- **Fréquence d'échantillonnage** : Les enregistrements ont été effectués à une fréquence de 16000 Hz, optimale pour la reconnaissance vocale et le traitement du signal.
- **Format** : Les fichiers audio sont au format WAV, préservant ainsi la qualité audio sans compression pertinente. Tous les enregistrements sont en mono, chaque fichier enregistrant un seul canal audio.

Voici un exemple (figure 3.10) :

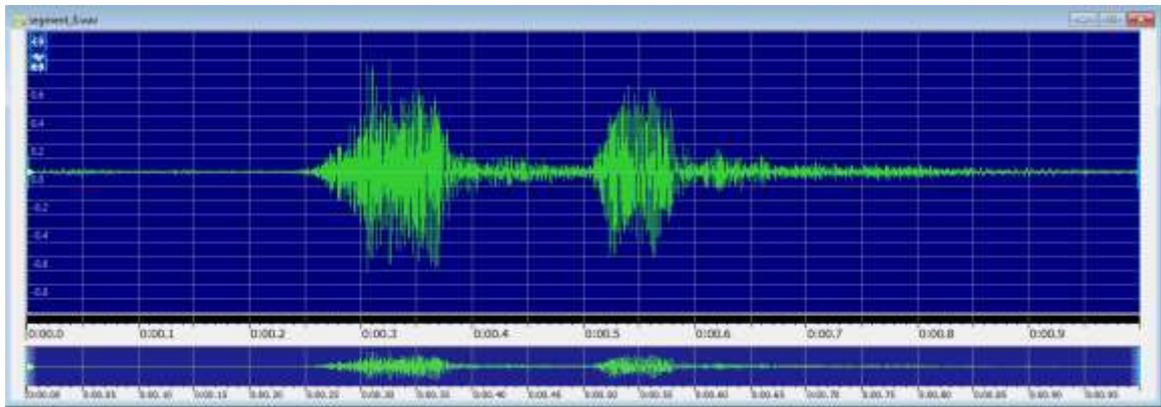


Figure 3.10: Enregistrement audio de la commande "طلع"

3.4 Difficulté du choix du modèle

La difficulté de choisir un modèle adapté à nos besoins spécifiques, en raison de notre manque de connaissance dans le domaine, a été un obstacle majeur au début de notre projet. Nous avons rencontré plusieurs difficultés, principalement dues à notre manque d'expérience dans le domaine de l'intelligence artificielle, et plus spécifiquement de l'apprentissage profond (deep learning). En conséquence, il nous a fallu un certain temps pour nous familiariser avec les concepts et les techniques nécessaires pour bien comprendre, ce qui a entraîné une perte significative de temps. Nous avons dû investir beaucoup de temps et d'efforts pour acquérir les bases théoriques et pratiques indispensables. Grâce à l'expertise de notre promoteur, cette phase d'apprentissage a été grandement facilitée. Son soutien et ses conseils nous ont permis de mieux comprendre les concepts complexes, de sélectionner un modèle approprié et d'appliquer plus

efficacement les connaissances acquises à notre projet. Cette phase d'apprentissage, soutenue par notre promoteur, s'est révélée essentielle pour poser des bases solides au projet et pour travailler de manière productive par la suite.

3.5 Problèmes de compatibilité matérielle et choix de la version du modèle

Nous avons rencontré des problèmes lors du transfert de notre travail sur l'ordinateur de notre promoteur en raison du manque de support AVX. Comme il est compréhensible que travailler avec la version la plus récente du modèle puisse poser des problèmes de compatibilité sur des machines plus anciennes ou moins puissantes, nous avons délibérément opté pour une version antérieure. Cette décision garantit une meilleure portabilité de notre projet, ce qui est crucial si nous prévoyons de le déployer sur différentes plateformes ou de le distribuer à des clients potentiels. En travaillant avec une version moins récente, nous nous assurons que le modèle fonctionnera de manière optimale même sur des machines plus anciennes ou moins performantes, élargissant ainsi considérablement notre audience cible et facilitant l'adoption de notre solution. De plus, cette approche nous a permis d'évaluer les performances et la compatibilité des deux versions du modèle sur différentes configurations matérielles, ce qui nous a aidés à prendre une décision éclairée sur la version à utiliser dans notre projet.

3.6 Interprétation des résultats des modèles

Cette analyse porte sur les performances de différents modèles de reconnaissance des commandes vocales en dialecte algérien, incluant le modèle de reconnaissance initiale, un modèle spécifique pour la voix féminine, un autre pour la voix masculine, ainsi qu'un modèle de confirmation binaire. Les performances sont évaluées à travers l'analyse des courbes de perte et des matrices de confusion, offrant un aperçu de l'évolution et de la précision de chaque modèle dans la classification des commandes vocales.

L'axe des X représente les époques d'entraînement. L'axe des Y indique la valeur de la perte, mesurant l'erreur du modèle à chaque époque.

▪ **Modèle de reconnaissance initiale pour le Dialecte Algérien**

La courbe de perte montre une descente progressive et régulière, indiquant une convergence efficace du modèle pendant l'entraînement (figure 3.11).

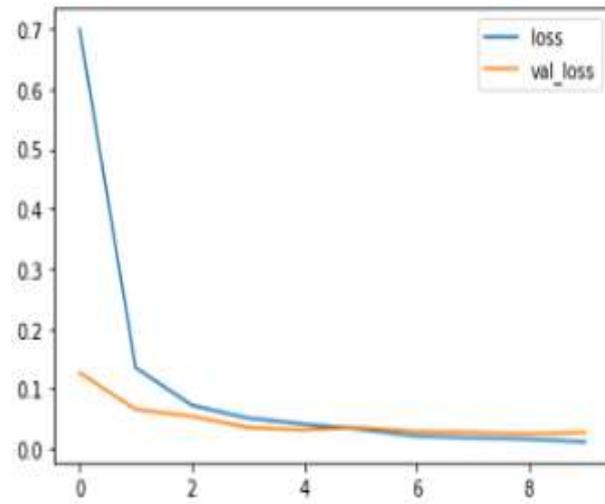


Figure 3.11: Courbe de perte du modèle de reconnaissance initiale pour le DA

Une précision de test de 99,89% dans la matrice de confusion confirme la capacité exceptionnelle du modèle à classifier avec précision les commandes vocales en DA (figure 3.12)

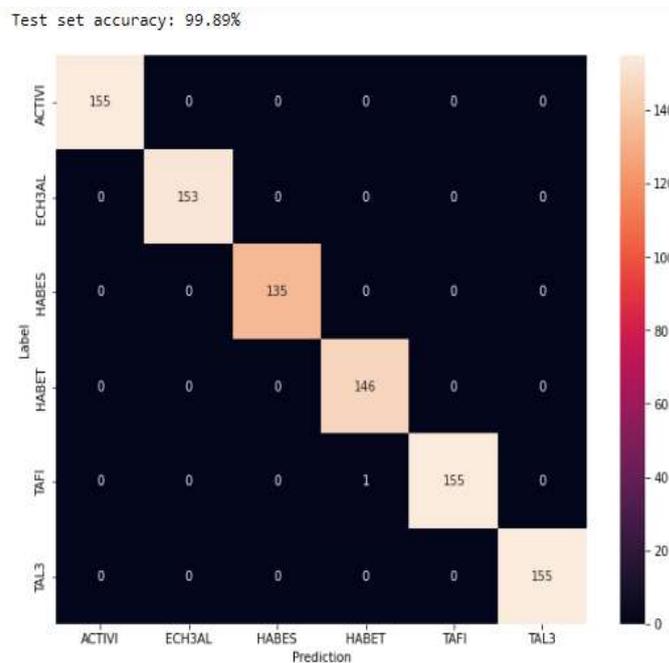


Figure 3.12: Matrice de Confusion du modèle de reconnaissance initiale pour le DA.

- **Modèle de reconnaissance pour voix féminine en Dialecte Algérien**

Les courbes de perte montrent une diminution progressive et une stabilité similaire entre les données d'entraînement et de validation (figure 3.13).

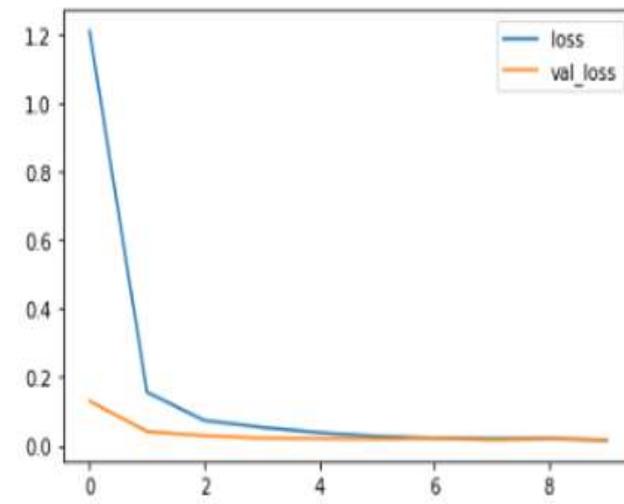


Figure 3.13: Courbe de perte du modèle de reconnaissance pour voix féminine

Malgré une légère baisse par rapport au modèle initiale la précision de test de 98,78% dans la matrice de confusion montre une performance solide et constante (figure 3.14).

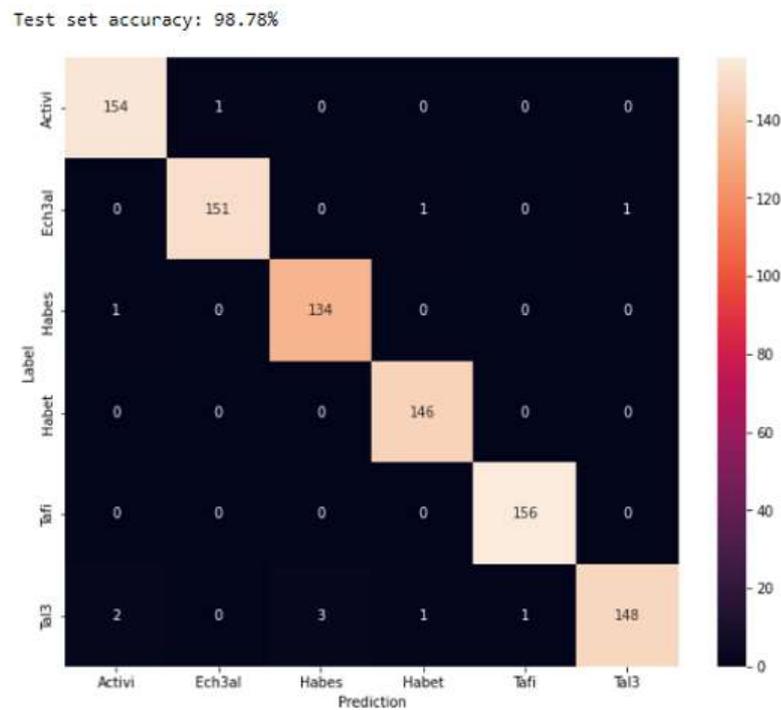


Figure 3.14: Matrice de confusion du modèle de reconnaissance pour voix féminine

- **Modèle de Reconnaissance pour Voix Masculine en Dialecte Algérien**

À la troisième époque, les courbes de perte montrent une diminution progressive et une stabilité similaire entre les données d'entraînement et de validation (figure 3.15).

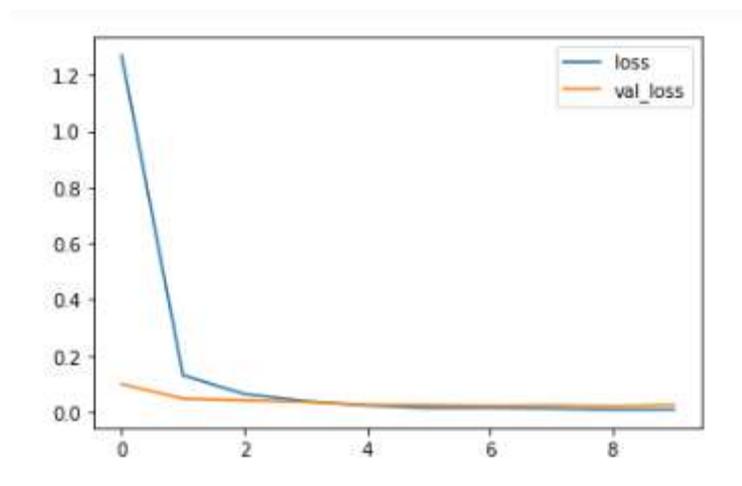


Figure 3.15: Courbe de perte du modèle de Reconnaissance pour Voix Masculine

La matrice de confusion révèle une précision remarquable de 99,67%, témoignant de la capacité exceptionnelle du modèle à classifier précisément les données d'évaluation (figure 3.16).

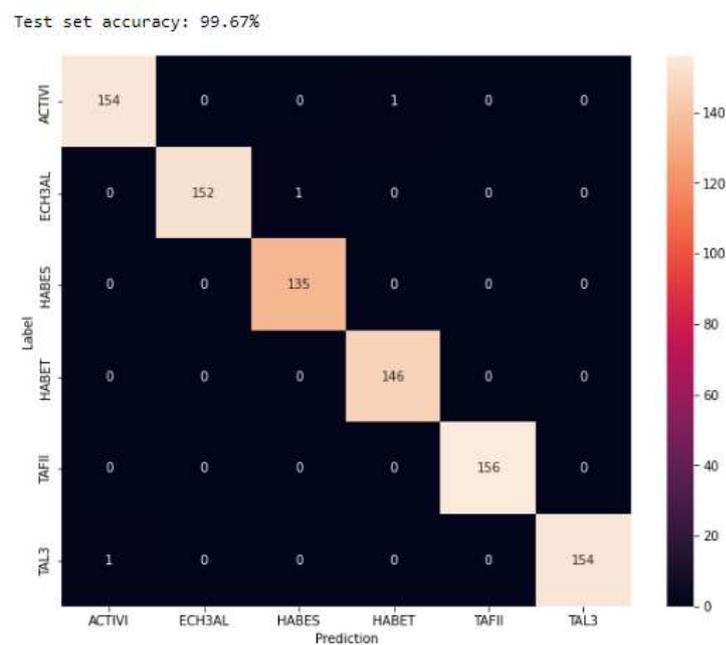


Figure 3.16: Matrice de confusion du Modèle de Reconnaissance pour Voix Masculine

- **Modèle de confirmation binaire pour la reconnaissance vocale**

La courbe de perte pour le modèle de confirmation montre une diminution continue et reste stable, ce qui signifie que le modèle améliore constamment sa capacité à généraliser à partir des données de validation tout en maintenant une performance constante (voir figure 3.17).

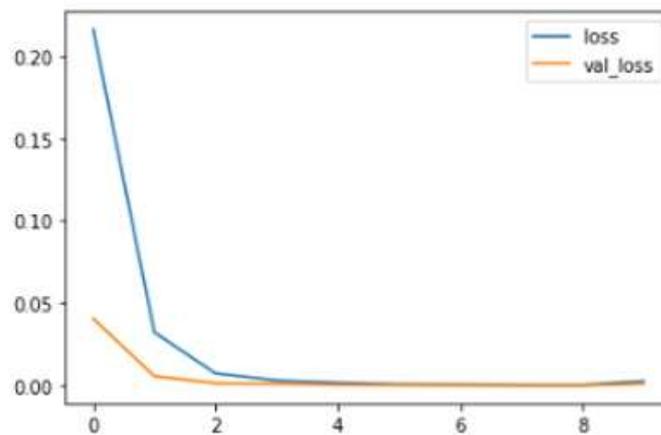


Figure 3.17: Courbe de perte pour modèle de confirmation binaire

La matrice de confusion montre une capacité parfaite à différencier les commandes "oui" et "non", garantissant une interprétation précise des intentions de l'utilisateur avec une précision de 100% (voir la figure 3.18).

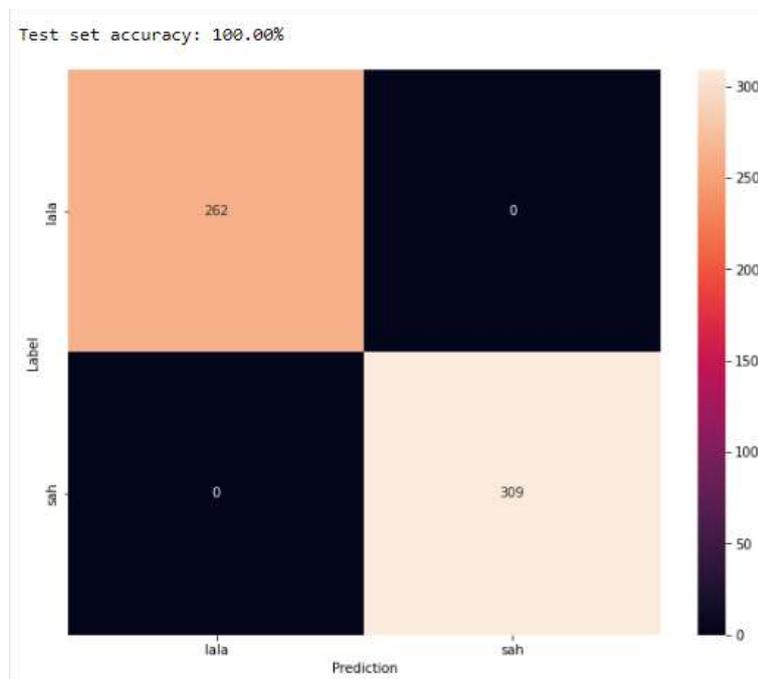


Figure 3.18: Matrice de confusion du modèle de confirmation binaire

3.7 Evaluation de la reconnaissance des commandes vocales

- Le modèle de reconnaissance vocale fonctionne sans nécessiter une connexion internet, ce qui constitue un avantage significatif.
- La précision et l'efficacité du modèle ont été évaluées en vérifiant sa capacité à reconnaître et transcrire correctement les commandes vocales en DA. Cette évaluation a été réalisée avec deux individus (une femme et un homme), chacun ayant sa propre base de données vocale. Les résultats ont montré une précision remarquable, avec un taux de reconnaissance de 98,78% pour l'individu féminin et de 99,67% pour l'individu masculin. Ces résultats démontrent l'efficacité élevée de la méthode de reconnaissance vocale dépendante du locuteur.
- La robustesse et la fiabilité du programme ont également été testées dans des environnements acoustiques difficiles, tels que la présence de bruit ou de conversations simultanées. Contrairement à l'utilisation de l'API de Google Speech Recognition, le programme n'a pas réussi à reconnaître les commandes dans ces conditions, révélant ainsi une faiblesse en environnements bruyants.

3.8 Connexion WiFi ESP32 avec Arduino IDE et Python

Pour intégrer l'ESP32 dans notre projet, nous avons suivi plusieurs étapes importantes.

Tout d'abord, nous avons installé le logiciel Arduino sur notre ordinateur pour programmer le microcontrôleur ESP32.

Ensuite, nous avons ajouté le support ESP32 dans Arduino (figure 3.19) en suivant les instructions d'Espressif. Cela nous a permis de choisir et de programmer facilement l'ESP32.

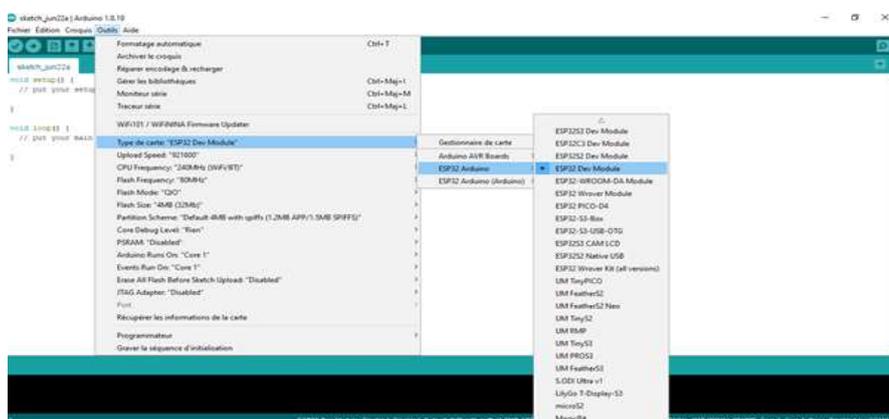


Figure 3.19: Configuration de l'esp32 dans l'IDE Arduino

En utilisant les bibliothèques WiFi.h et ESPAsyncWebServer.h dans l'IDE Arduino, nous avons écrit notre programme pour gérer la connexion Wi-Fi et créer un serveur web asynchrone sur l'ESP32.

Une fois le programme prêt, nous l'avons téléverser vers l'ESP32. En ouvrant le moniteur, nous avons obtenu l'adresse IP attribuée à l'ESP32 par le routeur Wi-Fi.

Cette adresse IP a été intégrée dans notre programme Python, utilisant la bibliothèque requests pour établir une communication bidirectionnelle avec l'ESP32. Grâce à cela, nous avons pu envoyer des commandes et recevoir des données, permettant un contrôle efficace de l'ESP32 à distance.

3.9 Implémentation de la solution proposée

Nous avons évalué l'ESP32 avant d'essayer de l'intégrer au système de gestion du parc de stationnement. Les tests détaillés ont montré que l'ESP32 possède une capacité solide à interagir avec divers équipements critiques du parking via une connexion sans fil, assurant ainsi la sécurité des agents en minimisant le risque lié à la haute tension (220V) des équipements.

Nous avons configuré l'ESP32 via l'IDE Arduino pour qu'il se connecte à notre réseau Wi-Fi et crée un serveur HTTP avec des routes pour contrôler la lampe, la barrière et l'alarme. En parallèle, nous avons développé un script Python avec PyQt5 pour une interface graphique. Après avoir vérifié et configuré l'adresse IP de l'ESP32 dans le script, nous avons utilisé les bibliothèques 'pyqt5' et 'requests' pour commander ces dispositifs via l'interface graphique (figure 3.20).

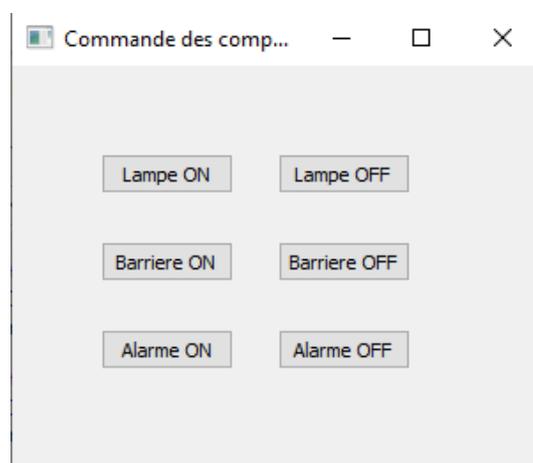


Figure 3.20: Interface des commandes.

Nous présentons les résultats obtenus :

- **Contrôle de la barrière :**

En cliquant sur "Barrière ON", l'ESP32 reçoit une requête et envoie une commande au relais connecté. Ce relais active alors le mécanisme de la barrière, permettant le passage, et l'interface confirme que la barrière est ouverte.

En cliquant sur "Barrière OFF", une nouvelle requête est envoyée par l'ESP32. Cette requête est traitée par le relais connecté, qui à son tour désactive le mécanisme de la barrière, bloquant ainsi le passage. L'interface indique alors que la barrière est fermée (figures 3.21 et 3.22).



Figure 3.21: Barrière ON et Barrière OFF.



Figure 3.22: Ouverture et fermeture de la barrière avec ESP32

- **Gestion de l'éclairage :**

En activant "Lampe ON", l'ESP32 envoie une commande au relais pour allumer la lampe en activant son circuit électrique. L'interface montre que la lampe est allumée.

En choisissant "Lampe OFF", une requête est envoyée au relais pour éteindre la lampe. L'ESP32 désactive le circuit électrique de la lampe, et l'interface confirme que la lampe est éteinte (figures 3.23 et 3.24).

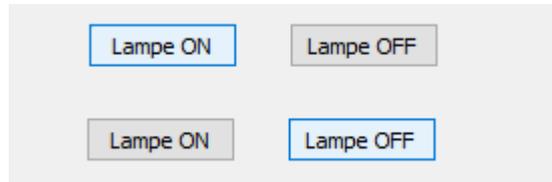


Figure 3.23: Lampe ON et Lampe OFF.



Figure 3.24: Allumer et éteindre la lampe avec esp32.

- **Activation/désactivation de l'alarme :**

En cliquant sur "Alarme ON", l'ESP32 envoie une commande au relais et active le système d'alarme (sirène). L'interface indique que l'alarme est active.

En sélectionnant "Alarme OFF", une requête est envoyée au relais pour désactiver l'alarme. L'ESP32 arrête la sirène. L'interface confirme que l'alarme est désactivée. (Figures 3.25 et 3.26).

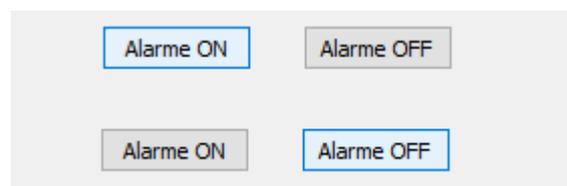


Figure 3.25: Alarme ON et Alarme OFF.

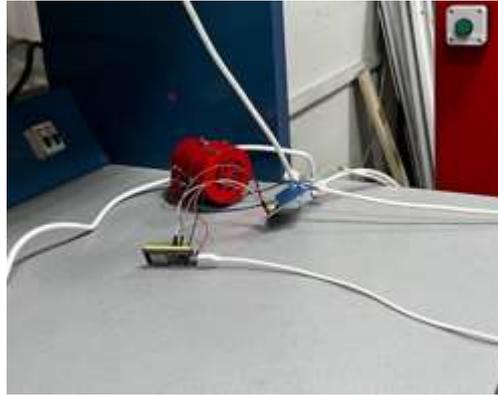


Figure 3.26: Activation et Désactivation de l'alarme

Malgré les résultats prometteurs obtenus lors des tests, des contraintes de temps ont influencé notre décision de ne pas intégrer complètement l'ESP32 dans notre système de gestion.

En conséquence, nous avons opté pour l'utilisation de l'Arduino, déjà intégré et fonctionnel pour interpréter les résultats du système de gestion de parking en utilisant notre modèle de reconnaissance vocale.

3.9.1 Intégration du modèle de la reconnaissance vocale

Après avoir intégré notre modèle H5 et apporté des modifications à l'interface selon nos spécifications dans l'algorithme final, lors de l'exécution du code avec Arduino, l'interface s'affiche. Nous sélectionnons alors les deux microphones et saisissons la variance (عتبة التباين). L'interface se présente comme suit (voir la figure 3.27).

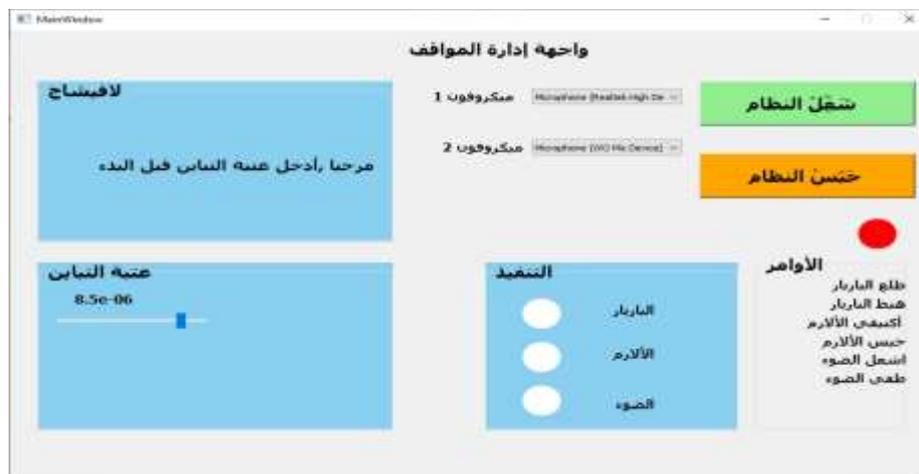


Figure 3.27: Interface de la gestion intelligente du parking en DA.

Ensuite, avant de cliquer sur le bouton de démarrage (شَغْلُ النظام), il faut connecter l'Arduino (الأردوينو مكوينيكي) et claquer des mains. L'écran affiche le message « واش تحوس ؟ » Ensuite, le cercle rouge à droite deviendra vert et affichera devant lui « راني نسمع » afin de pouvoir prononcer les commandes. L'Arduino doit être connecté. Sinon (الأردوينو ماشي مكوينيكي) il est impossible de claquer les mains et dire les commandes (figure 3.28 et 3.29).



Figure 3.28: Affichage de l'interface.



Figure 3.29: Affichage d'Arduino connecté et non connecté.

La figure ci-dessous présente les commandes que nous avons choisies en DA, à prononcer (voir Figure 3.30).



Figure 3.30: Liste des commandes.

Après avoir prononcé la commande souhaitée, le système va faire la prédiction et demander une confirmation avant l'exécution. Une fois la confirmation obtenue, il procédera à l'exécution. Sinon, il demandera de prononcer à nouveau la commande.

Prédiction de la commande :

Quand le système entend la commande, va faire une prédiction ("طلع", "هبط", "اشعل", "حس", "اكتيفي" ou "حس"). Ensuite, il va demander si cette prédiction est compatible avec la commande voulue en affichant le message « 'La prédiction' ؟ راک سووور ؟ » (voir figure 3.31) .



Figure 3.31: Affichage de confirmation.

Confirmation de la commande :

Après que le système demande une confirmation pour sa prédiction en prononçant (ايه ou لالا), il anticipera cette confirmation et exécutera la commande finale avec le message « ' La commande ' راک أکدت ؟ » (figures 3.32, 3.33 et 3.34)



Figure 3.32: Affichage d'ouvrir et fermer la barrière.



Figure 3.33: Affichage du message activation et désactivation de l'alarme.



Figure 3.34: Affichage d'allumer et éteindre l'éclairage.

3.10 Conclusion

Dans ce chapitre, nous avons exposé les résultats expérimentaux de notre projet de gestion intelligente des parkings utilisant des commandes vocales en DA. Nous avons d'abord décrit les différents logiciels et langages de programmation employés. Ensuite, nous avons expliqué la mise en place de la connexion Wi-Fi ESP32 avec Arduino IDE et Python. Nous avons également détaillé l'implémentation du projet en décrivant les étapes et les choix techniques effectués. Enfin, nous avons évalué le model en analysant ses performances.

Les résultats obtenus montrent que notre model fonctionne efficacement. Les avantages de notre model sont les suivants :

- Fonctionne en DA.
- Ne nécessite pas de connexion internet.
- Le modèle est dépendant du locuteur, c'est-à-dire qu'il est spécifiquement conçu pour fonctionner avec l'agent.

Cependant, notre solution présente quelques limitations :

- Ne fonctionne pas correctement en présence de bruit élevé. Par exemple, si une alarme se déclenche, le modèle ne pourra pas entendre la commande pour l'éteindre.

CONCLUSION GENERALE ET TRAVAUX FUTURE

L'objectif de ce projet est d'améliorer un système de gestion intelligente des parkings déjà existant. Ce système est conçu pour reconnaître en temps réel les commandes vocales données par l'agent de contrôle et pour exécuter les tâches suivantes : (1) ouvrir et fermer la barrière principale, (2) allumer et éteindre automatiquement l'éclairage du parking, et (3) activer et désactiver l'alarme en cas de vol.

La méthodologie adoptée dans ce projet comporte deux étapes principales : la reconnaissance des commandes vocales en DA et l'intégration de la carte ESP32. Nous avons d'abord créé une base de données en DA, composée de 6000 enregistrements audio pour une seule personne, répartis en six classes de 1000 enregistrements chacune. Pour éviter de devoir enregistrer 6000 fichiers audio en une seule fois, nous avons mis au point une méthode permettant d'augmenter la base de données à partir d'un nombre limité d'enregistrements. Un modèle CNN pré-entraîné a été employé pour la reconnaissance des commandes vocales. Nous avons également testé la capacité de l'ESP32 à contrôler les équipements du parking (220V) de manière sans fil, bien que nous n'ayons pas eu le temps d'intégrer complètement cette fonctionnalité, nous avons utilisé l'Arduino préalablement intégré.

Le deuxième chapitre de notre étude détaille davantage les aspects techniques mentionnés. Le troisième chapitre est consacré aux résultats expérimentaux et à leur interprétation. Nous avons utilisé divers logiciels et outils, tels que Jupyter Notebook, Arduino, ESP32, Wo Mic, Matlab et GoldWave. Les performances du modèle de reconnaissance vocale ont été évaluées avec deux personnes (un individu masculin et un individu féminin), atteignant un taux de précision de 99,22 %.

Notre système améliore le travail existant en reconnaissant les commandes en dialecte algérien et en fonctionnant sans connexion Internet. Toutefois, il reste très sensible au bruit dans des environnements acoustiques difficiles.

Les travaux futurs se concentreront sur :

- Le développement d'une solution robuste pour la reconnaissance vocale dans des environnements acoustiques difficiles.

- Mettre en place une méthode de vérification de locuteur afin de garantir que seule l'agent est autorisé à accéder à la plateforme de gestion du parking.
- L'intégration complète de l'ESP32 pour une connexion sans fil sécurisée sera priorisée. Cette intégration vise à garantir la sécurité des communications tout en simplifiant l'installation des équipements du parking. Des mesures seront prises pour assurer la robustesse et la stabilité du réseau sans fil, en particulier pour le contrôle des équipements nécessitant une alimentation de 220V.

References

- [1] O. Cherir and A. Mekdour, "Gestion Intelligente Des Parkings Par Commandes Vocales," Mémoire de Master, Département d'électronique, Université Blida 1, 2023.
- [2] Y. Tikialine and K. Khedda, "État De L'art De « Smart Parking », Étude Et Conception D'un Prototype De Stationnement Intelligent," Mémoire de Master, Département de Technologie, Université Djilali Bounaama Khemis Miliana, 2018.
- [3] " linkedin ", [Online]. Available: https://www.linkedin.com/pulse/smart-parking-evolution-journey-from-sensors-ai-driven-ayman-najim-o0fcc_ [Accessed 01 06 2024].
- [4] " sotefinparking ", [Online]. Available: <https://www.sotefinparking.com/story-of-the-evolution-of-car-parking-systems-4-things-you-must-know-and-understand/>. [Accessed 01 06 2024].
- [5] A. Shah, D. Shah, A. Satpute, M. Shinde, S. Shinde, "Literature Review on Parking System," International Journal of Engineering Research & Technology (IJERT), vol. 10, no 10, p. 100-102, 2021.
- [6] J. Cynthia, C. Bharathi Priya, P. A. Gopinath, "IOT based Smart Parking Management System," International Journal of Recent Technology and Engineering (IJRTE), vol. 7, no 4S, p. 374-379, 2018.
- [7] Y_Geng, CG_Cassandras, "A new "Smart Parking" System Infrastructure and Implementation," Procedia-Social and Behavioral Sciences, vol. 54, p. 1278-1287, 2012.
- [8] J Shi, L Jin, J Li, Z Fang, "A Smart Parking System Based on NB-IoT and Third-party Payment Platform," 2017 17th international symposium on communications and information technologies (ISCIT), IEEE, p. 1-5, 2017.
- [9] IH Jung, JM Lee, K Hwang, "Advanced Smart Parking Management System Development Using AI," Journal of System and Management Sciences, vol. 12, no 1, p. 53-62, 2022.
- [10] N. Houacine et N. Khelifa, "Classification Des Textures Par Les Réseaux De Neurones Convolutifs," Memoire De Master, Departement D'automatique, Université Mouloud Mammeri De Tizi-Ouzou, 2018.
- [11] "Tensorflow," [Online]. Available: https://www.tensorflow.org/tutorials/audio/simple_audio/. [Accessed 20 06 2024].
- [12] D. Pratik, Python Programming for Arduino, Royaume-Uni, Birmingham: Packt Publishing, 2015.

[13] M. Babiuch, P. Foltýnek, P. Smutný, "Using The ESP32 Microcontroller For Data Processing," 2019 20th International Carpathian Control Conference (ICCC), IEEE, 2019.

[14] "MATLAB," [Online]. Available: <https://fr.mathworks.com/>. [Accessed 22 06 2024].

[15] R. Cao, "GoldWave in Pronunciation Education: The Application of Voice Editing Software," 2015 7th International Conference on Information Technology in Medicine and Education (ITME), IEEE, 2015, p. 468-472, 2015.

[16] "Wo Mic," [Online]. Available: <https://wo-mic.fr.download.it/>. [Accessed 22 06 2024].