UNIVERSITÉ DE BLIDA 1

Faculté des sciences

Département d'informatique

MÉMOIRE DE MASTER

En Informatique

Option : Ingénierie Des Logiciels

THÈME :

Plateforme de reconnaissance des sons de l'environnement en temps réel pour la surveillance audio des parcs de stationnement

Réalisé par :

LEBRES Imad Eddine & OURARI Niema

Devant un jury composé de :

BACHA Sihem, Maitre de conférence B, Blida 1	Président
TEBBI, Maitre-Assistant B, Blida 1	Examinatrice
YKHLEF Fayçal, Maître de Recherche, CDTA	Promoteur
YKHLEF Hadjer, Maitre de Conférence B, Blida1	Co-Promoteur

2021/2022

REMERCIEMENTS

En préambule à ce mémoire, nous remercions ALLAH qui nous a aidé et donné la patience et le courage durant cette longue d'année d'étude.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Ces remerciements vont tout d'abord à notre promoteur, Dr. Fayçal YKHLEF, Maitre de Recherche B au CDTA, Pour nous avoir proposé ce sujet que nous trouvons intéressant, pour son encadrement, le temps qu'il a consacré à la correction et à la relecture de ce document et son suivi ainsi que ses conseils précieux qui nous ont permis d'aboutir à la production de ce mémoire.

Nous remercions très chaleureusement aussi, Dr. Hadjer YKHLEF, Maitre de Conférence au niveau de l'Université SAAD DAHLEB, Notre encadreur, pour sa confiance et ses conseils.

Nous présentons nos respects et nos sincères remerciements aux membres du jury qui nous ont fait l'honneur d'avoir accepté de faire partie du jury et de nous avoir consacré de leur temps précieux.

Merci à tous qui ont participé de près ou de loin à l'aboutissement de ce travail.

DEDICACE

Je dédie ce modeste travail

A mes chers parents,

Que nulle dédicace ne puisse exprimer ce que je leur dois, pour leur bienveillance, leur affection et leur soutien. Trésors de bonté, de générosité et de tendresse, en témoignage demon profond amour et ma grande reconnaissance « que DIEU vous garde ».

A mes chères sœurs,

Je leur dédie ce modeste travail en témoignage de mon grand amour et ma gratitude infinie.

A toutes les personnes que j'aime et qui m'aiment,

Pour leur aide et leur soutien moral durant l'élaboration du travail de fin d'études et lecomble de ma gratitude et respect vont vers mon fiancé et toute sa famille.

A mon binôme Imad,

Avec qui j'ai pu réaliser ce travail grâce àson aide sa compréhension et sa présence.

Que DIEU vous protège tous.

Ourari Niema

DEDICACE

Je dédie ce modeste travail

A mes Chers parents,

Qui sont les meilleurs parents dans ce monde, c'est grâce à leur soutien moral, leur encouragement et leur confiance en moi que j'ai pu mener à terme ce projet. Tout en témoignant de ma profonde gratitude et de mon incontestable reconnaissance pour tous leurssacrifices et tout l'amour dont ils m'entourent.

A mes chers frères et A ma chère sœur,

Je leur dédie ce modeste travail en témoignage de mon grand amour et ma gratitude infinie.

A mes chers proches,

Qui étaient toujours à mes côtés et m'ont été d'un soutien précieux. Je leur souhaite à eux aussi plein de succès dans leur vie.

A mon binôme Niema,

Avec qui j'ai partagé des moments inoubliables et avec qui j'ai pu réaliser ce travail grâce à son aide sa compréhension et sa présence.

Que DIEU les protège tous.

Lebres Imad Eddine

RESUME

ملخص:

يهدف مشروعنا إلى تصميم وتنفيذ منصة التعرف على الأحداث الصوتية في الوقت الفعلي لمراقبة مواقف السيارات. نحن مهتمون بالتعرف على ثلاث فئات من الأصوات: صراخ البشر، وأجهزة إنذار السيارات، وتكسير الجليد. تم اعتماد الذكاء الاصطناعي القائم على مركز البيانات لتطوير نظامنا. ويتألف نهجنا من مرحلتين للمعالجة: '1' الكشف عن الأحداث الصوتية المتهورة و '2' التعرف السليم. وتستند طريقة الكشف عن الأحداث إلى تحليل تباين الطاقات القصيرة الأجل للموجة الصوتية. ومع ذلك، تعتمد مرحلة الاستطلاع على Dynamic Time Warping. تم استخدام معاملات Cepstral (MFCCs) كخصائص صوتية. تمت مقارنة نتائج التعرف لدينا بنموذج التعلم باستخدام BI-LSTM (ذاكرة قصيرة المدى طويلة الاتجاه ثنائية الاتجاه). يتعرف البرنامج الذي طورناه على الفئات الثلاث للأصوات المذكورة أعلاه في الوقت الفعلي. النتائج مشجعة للغاية..

الكلمات الدالة: الكشف عن الأحداث الصوتية، التعرف على الأحداث الصوتية، في الوقت الفعلي، الذكاء الاصطناعي على أساس مركز البيانات، التشوه الزمني الديناميكي، BI-LSTM-RNN، MFCC.

Résumé :

Notre projet vise à concevoir et implémenter une plateforme de reconnaissance des événements acoustiques en temps réel en vue de surveiller les parcs de stationnements. Nous nous intéressons à la reconnaissance de trois catégories de sons : cris humain, alarmes de voitures et bris de glace. L'IA à base du Data Centric est adoptée pour développer notre système. Notre approche est composée de deux étages de traitement : (i) la détection des évènements sonores impulsifs et (ii) la reconnaissance des sons. La méthode de détection des évènements est basée sur l'analyse de la variance des énergies à courts termes de l'onde acoustique. Cependant, l'étage de reconnaissance est basé sur le Dynamic Time Warping. Nous avons utilisé les coefficients cepstraux (MFCCs) comme attributs acoustiques. Nos résultats de reconnaissance ont été comparés à un modèle d'apprentissage utilisant le BI-LSTM (Bidirectional long short-term memory). Le software que nous avons développé reconnait les trois catégories de sons cités ci-dessus en temps réel. Les résultats obtenus sont très encourageants.

Mots clés : Détection des évènements sonores**,** reconnaissance des évènements acoustiques, en temps réel, l'IA à base du Data Centric, la déformation temporelle dynamique, BI-LSTM, MFCCs.

Abstract:

Our project aims to design and implement a real-time acoustic event recognition platform to monitor parking lots. We are interested in the recognition of three categories of sounds: human screams, car alarms and ice breaking. Data Centric-based AI is adopted to develop our system. Our approach consists of two processing stages: (i) detection of impulsive sound events and (ii) sound recognition. The method of detecting events is based on the analysis of the variance of the short-term energies of the acoustic wave. However, the reconnaissance stage is based on the Dynamic Time Warping. Cepstral coefficients (MFCCs) were used as acoustic attributes. Our recognition results were compared to a learning model using BI-LSTM (Bidirectional long short-term memory). The software we developed recognizes the three categories of sounds mentioned above in real time. The results are very encouraging.

Keywords: Detection of sound events, recognition of acoustic events, in real time, AI based on Data Centric, dynamic temporal deformation, BI-LSTM, MFCC.

LISTE DES ACRONYMES ET ABREVIATIONS

BLSTM: Bidirectional long short-term memory

DTW: Dynamic Time Warping

DCAI: Data Centric Artificial Intelligence

Fe: Fréquence d'échantillonnage

FN: False Negative

FP: False Positive

GSS : Première Génération des Systèmes de Surveillance

LSTM: Long short-term memory

MFCCs: Mel-Frequency Cepstral Coefficients

MEMS: Micro-Electrical-Mechanical Systems

SI: Son impulsive

TFP : Taux de faux positifs

TFN : Taux de faux négatifs

TVP : Taux de vrais positifs

TBC : Taux de bonne classification

TP: True Positive

TN: True Negative

TPR : True Positive Rate

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

Chapitre 1 : Introduction

1.1 Motivations

La surveillance audio est un mécanisme de sécurité crucial pour tous les pays du monde. Par conséquent, la mise en place des systèmes efficaces de surveillance est devenue essentielle dans les lieux publics et privés. Les systèmes de surveillance de troisième génération intègrent une variété de capteurs aux caméras qui fournissent des informations complémentaires sur les évènements anormaux. Plusieurs types de capteurs peuvent être exploités, on peut citer : les capteurs infrarouges, les capteurs de chaleur, les microphones et les capteurs de mouvement.

Les évènements audio capturés par les microphones en dehors du champ de vision des caméras, ou lorsque les conditions deviennent défavorables permettent de pivoter les caméras vers les sources d'évènements acoustiques dangereuses (cris humains, coups de feu, bris de verre, aboiement de chiens, accident de voitures et autres).

La reconnaissance de ces évènements est une phase clé pour l'implémentation d'un système de surveillance de troisième génération. L'incorporation de l'information audio pour la surveillance des lieux est l'objectif de ce projet.

Notre stage au CDTA consiste à concevoir et implémenter le module de reconnaissance des sons de l'environnement en temps réel en vue de surveiller les parcs de stationnements.

1.2 Contributions

Nous nous focalisons dans le cadre de ce projet à la détection des évènements impulsifs et à la reconnaissance de trois catégories de sons : (i)cris humains, (ii)bris de glace et (iii) alarme de voiture.

Notre contribution consiste à concevoir et implémenter une plateforme de reconnaissance des évènements sonore en temps réel. Notre approche est composée de deux étages principaux : (i) la détection des évènements impulsifs et la (ii) classification des sons.

La méthode de détection est basée sur l'analyse de la variance des énergies a courts termes[1]. Nous avons optimisé les performances de cette méthode en se basant sur plusieurs

paramètres algorithmiques. Nous utilisons des microphones de type MEMS pour l'acquisition des sons.

La méthode de reconnaissance est basée sur le Dynamic Time Warping. Les coefficient cepstraux (En anglais : Mel-Frequency Cepstral coefficients MFCC) est utilisé pour l'extraction des attributs. Le schéma de classification proposé repose sur l'approche l'IA à base du Data Centric.

1.3 Impact du projet

Le travail effectué dans notre projet de fin d'étude est très important pour assurer la sécurité des citoyens et des biens dans les parcs de stationnements. En Algérie, la technologie de surveillance audio est moins courante. De ce fait, la maîtrise de ces techniques peut contribuer au développement du secteur socio-économique. Elles peuvent intéresser sans aucun doute plusieurs clients et institutions. Nous citons : les Hypermarchés, les sociétés, les banques, les ministères et la direction générale de la sureté interne.

1.4 Organisation du mémoire

Notre mémoire se subdivise donc comme suit :

Le premier chapitre : ce chapitre présente des généralités sur les parcs de stationnement et les systèmes de surveillance et se focalise principalement sur l'apport de la modalité audio à ce genre de système. Il présente aussi quelques travaux sur la surveillance des parcs de stationnement.

Le deuxième chapitre : Dans ce chapitre nous décrivons l'approche proposée pour la reconnaissance en temps réel des sons de l'environnement.

Le troisième chapitre : Ce chapitre est consacré à la présentation des résultats expérimentaux ainsi que leurs interprétations.

Chapitre 2 : Surveillance des parcs de stationnement

2.1 Introduction

Dans ce chapitre nous présentons des généralités sur les parcs de stationnement et les systèmes de surveillance. Nous nous focalisons sur l'apport de la modalité audio à ce genre de système ainsi que la description de la détection, la reconnaissance et la localisation des évènements impulsifs. Nous citons par la suite quelques travaux sur la surveillance des parcs de stationnement. Nous terminons ce chapitre par une conclusion.

2.2 Parcs de stationnement

La croissance rapide de la population urbaine ainsi que l'augmentation du nombre de véhicules engendrent des problèmes de congestions routiers à l'intérieur des villes. Les parcs de stationnement représentent la meilleure solution pour diminuer les congestions [2]. Par définition, un parc de stationnement est une zone protégée qui permet le remisage, à l'extérieur de la voie publique, des véhicules automobiles et de leurs remorques. Cette zone peut être une annexe d'un ou de plusieurs bâtiments d'habitation[3].

Au début des années 1920, il n'y avait pas encore de stationnement réglementé dans les villes. Les gens gareraient simplement leurs voitures dans les rues et les laisseraient là jusqu'à ce qu'ils en aient à nouveau besoin. Cette pratique a engendré des problèmes de congestion du trafic dans les grandes villes. Les managers des villes ont rapidement commencé à réfléchir à un moyen de réglementer le temps de stationnement. Le 16 juillet 1935, le premier parcomètre au monde, connu sous le nom de Park-O-Meter N ° 1, a été installé à l'Oklahoma par monsieur Carl C. Magee [4]. Par la suite, des espaces de stationnement, privées et publics, ont été créés dans plusieurs villes aux états unis et en Europe. Pour une longue période, seuls les parkings traditionnels ont été utilisés. Au fil des ans, la technologie a pris le dessus et les garages automatisés ont émergé [5].

Les parcs de stationnement constituent une entité importante pour la gestion routière. Au début, le management des parcs consistait à compter le nombre de voitures garées et à estimer le temps écoulé par chaque véhicule. Cependant, de nos jours, un système intelligent de gestion de parkings ne gère pas simplement le fonctionnement interne du parc, mais il identifie et contrôle également tous les espaces de stationnement [6].

Il existe plusieurs types de parc de stationnement. On peut citer les parkings à étages, les parcs souterrains, et les parkings automatisés avec des ascenseurs [3]**.**

Les parkings sont souvent surpeuplés. Les accidents qui se produisent à l'intérieur de ces espaces sont fréquents surtout dans les parcs souterrains où la visibilité n'est pas excellente. Pour sécuriser ces espaces, les lieux de stationnement sont souvent équipés de systèmes de surveillance (Figure 2.1) [5]. Les technologies utilisées pour assurer la surveillance des lieux sont variées. Nous présentons dans la section suivante un aperçu sur les systèmes les plus repandus.

Figure 2.1: Surveillance d'un parc de stationnement [7] .

2.3 Systèmes de surveillance

La garantie de la sécurité des biens et des personnes est une problématique qui suscite un intérêt croissant dans le monde contemporain. Les technologies liées au marché de la surveillance, particulièrement la surveillance vidéo, sont devenues incontournables dans tout environnement urbain modern.

Un système de surveillance est un ensemble d'outils informatiques et électroniques destiné à assister les agents de sécurité [8].

Le premier système a été conçu en 1942, par la société Allemande Siemens AG afin de suivre les différentes étapes de lancement des fusées. Ce n'est que 7 ans après, que le premier

système conçu a été commercialisé aux Etats-Unis. L'utilisation des caméras de surveillance à cette époque ne concernait pas un usage domestique [9]. Par la suite, à partir des années 90, la vidéosurveillance a été déployée dans les villes, les routes, les transports, les immeubles et même les parkings [10].

D'un point de vue technologique, les systèmes de surveillance sont classés en quatre générations principales (voir Figure 2.2). Ces systèmes ont évolué d'une surveillance contrôlée par l'opérateur à une surveillance automatisée, intelligente et intégrée [11].

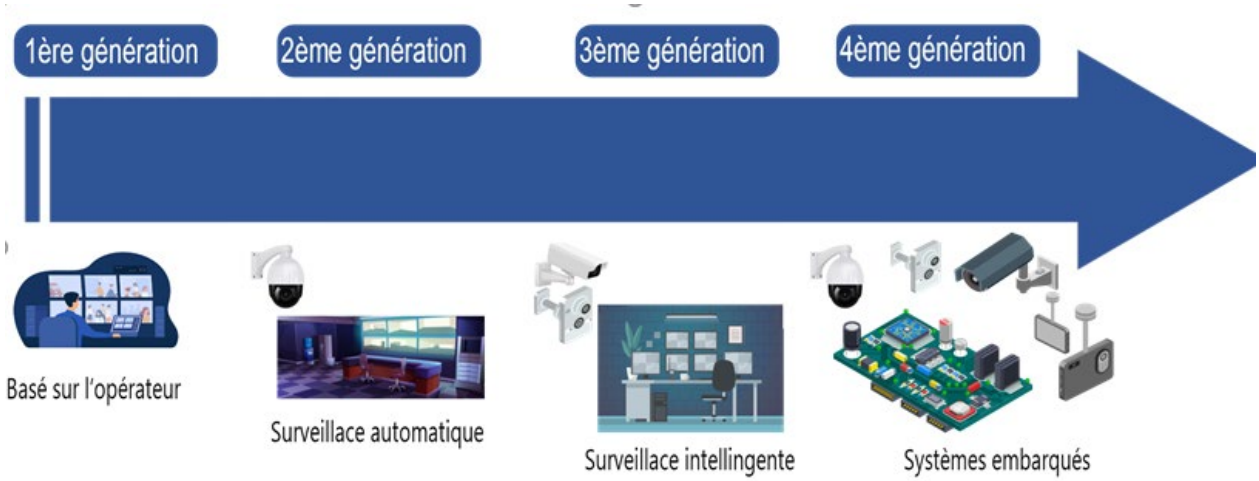

Figure 2.2: Evolution des systèmes de surveillance

La première Génération des Systèmes de Surveillance (1er GSS) (1960-1980) repose sur des systèmes anthropocentriques (axés sur l'exploitant) destinés à l'acquisition et la transmission d'images et vidéos [12]. Les flux vidéo, captés en tant que signaux analogiques, apparaissent sur de gros moniteurs. L'agent de sécurité a pour mission d'analyser, d'interpréter et de catégoriser des observations. Toutefois, ces systèmes souffrent de plusieurs inconvénients et limitations. On peut citer : (i) les difficultés d'archivage et de récupération des flux vidéo, (ii) la faible qualité des images enregistrées, et (iii) les erreurs d'observation commises par les agents de surveillance [12].

Pour réduire ces erreurs, les systèmes de surveillance de deuxième génération (2eme GSS, 1980-2000) ont été adoptés. Ils sont basés sur des cameras IP pour acquérir les flux vidéo ainsi que des méthodes avancées de traitement d'image et de reconnaissance de formes pour détecter les évènements anormaux en temps réel. Les flux vidéo captées sont transmis via des commutateurs

réseau et affichés sur un moniteur de PC. La qualité des images a été nettement améliorée par rapport à la génération précédente vu l'incorporation des dispositifs numériques [11], [12].

Avec la disponibilité des unités de calcul, le faible coût du matériel et la possibilité de communiquer avec des appareils mobiles, les systèmes de surveillance de troisième génération ont vu le jour (3eme GSS, à partir de 2000). Ces systèmes consistent à déclencher une alerte en temps réel pour permettre aux opérateurs de localiser immédiatement les évènements dangereux en utilisant des solutions entièrement numériques et une multitude de capteurs pour identifier les informations provenant de différents modalités (sonore, visuelle, mouvement, etc.) [12] [13]. Ces systèmes ne peuvent pas garantir une surveillance dans les zones non habitées, les forêts, et les montagnes car ils ne sont pas réalisables dans ces endroits en raison de leur consommation d'énergie, et de leur faible stabilité puisqu'elles dépendent du PC. Parmi les technologies de cette génération, on cite le système de détection de coups de feu installé par la police de San Diego [14]. Ce système envoie automatiquement une alarme aux patrouilles de police si un coup de feu est détecté. Il permet aussi de différencier entre des balles réelles et fausses.

Récemment, les systèmes de surveillance de quatrième génération (4eme GSS) sont apparus afin de faire face aux échecs de 3eme génération. Des progrès considérables ont été accomplis en ce qui concerne les plateformes embarquées qui offrent une plus grande adaptabilité que les plateformes basées sur PC[11]. Ces systèmes peuvent fournir un contrôle plus approprié et un traitement plus précis sans qu'il soit nécessaire de disposer d'un poste de calcul central (ordinateur). Ces systèmes utilisent des capteurs distribués à faible consommation qui conviennent mieux aux zones inhabitées [11].

En plus de la surveillance des parcs de stationnement, les technologies de surveillance sont exploitées pour (i) patrouiller les frontières nationales, (ii) évaluer le flux des réfugiés, (iii) assurer la sécurité des zones entourant les bases militaires et (iv) mesurer la vitesse des véhicules [11].

2.4 Apport de la modalité audio dans la surveillance des parcs de stationnement

Aujourd'hui, l'utilisation de la modalité audio pour la surveillance est en pleine croissance. En fait, l'information audio est efficace pour décrire les activités humaines et les liens sociaux (cris, discussion, etc.) dans les environnements urbains [11].

Cette modalité est en effet complémentaire à la vidéo puisqu'elle capture les évènements qui se produisent hors du champ de la caméra ou dans des conditions défavorables (Figure 2.3) [11].

Figure 2.3: Apport de la modalité audio pour la surveillance.

2.5 Détection, reconnaissance et localisation des sons

2.5.1 Sons de l'environnement dangereux

Le son impulsif (SI), dit évènement audio, est un son transitoire provoqué par une libération soudaine d'énergie, émis au cours d'événement acoustique peut être identifié en tant qu'indicateur d'une situation dangereuse[15]. Celle-ci est une désignation qui peut être adaptée à de nombreux sons, tels que les cris humains, les alarmes de voiture, les brise-glace, les explosions de pneus. Nos oreilles peuvent détecter différents facteurs caractéristiques de ce type de bruit[16].

2.5.2 Détection des événements audio

Une des phases les plus importantes pour la conception des systèmes de surveillance audio est la détection des sons impulsives (SI). Cette phase consiste à déterminer l'instant de début des événements acoustiques impulsifs. A la fin de cette opération, l'information donnée est qu'un évènement spécial (souvent dangereux) s'est produit à la suite d'une brusque augmentation

d'énergie [15]. Cette étape est considérée comme une étape de pré-traitement avant la phase de reconnaissance [15].

2.5.3 Reconnaissance des sons

La reconnaissance des évènements sonores est une technologie utilisée dans plusieurs domaines d'application tels que la surveillance. Cette opération consiste à identifier le type exact du son impulsif produit et le catégoriser selon différentes classes [15]. La Figure 2.4 donne un aperçu sur la détection et la reconnaissance des SI.

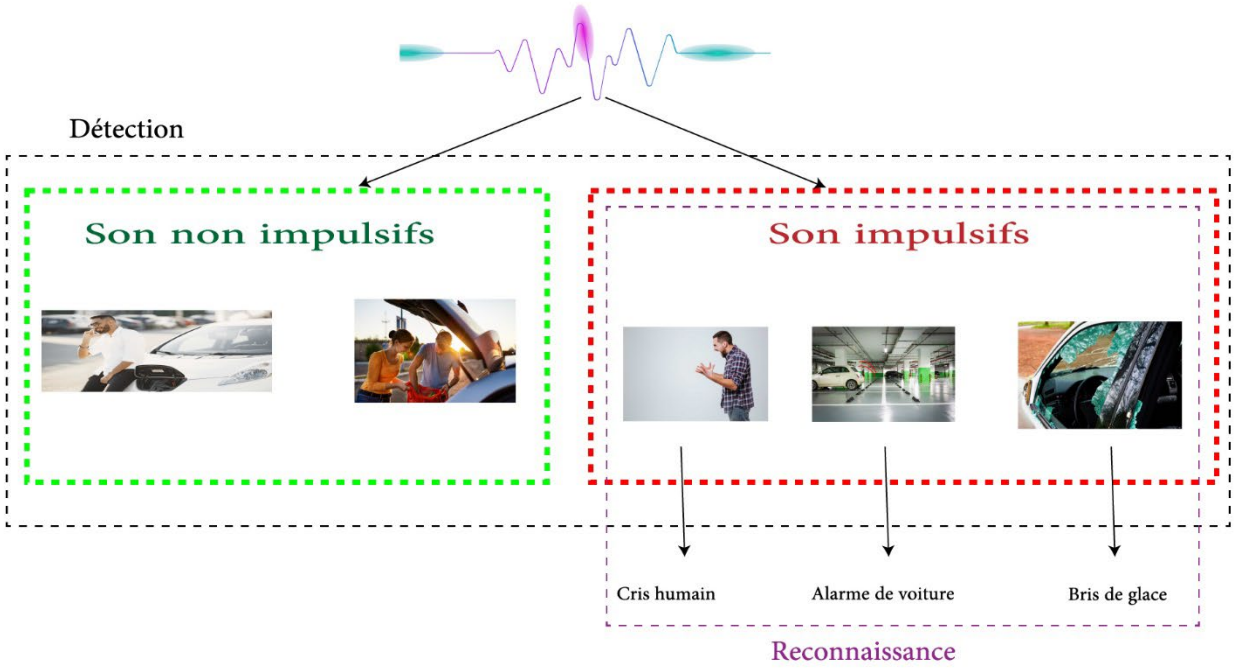

Figure 2.4: Exemples de détection et de reconnaissance des sons impulsifs

2.5.4 Localisation des sources sonores

La localisation des sons permet d'identifier l'emplacement de l'événement acoustique dans un champ sonore. Une fois la source sonore est localisée, le système utilise des caméras pour surveiller la scène de l'événement.

2.6 Etat de l'art sur la surveillance des parcs de stationnement

Dans cette section, nous présentons quelques recherches sur la surveillance des parcs de stationnements. Les résumés des travaux sont donnés comme suit :

G. Ciaburro et al. [17] ont élaboré une méthode basée sur les réseaux de neurones convolutifs pour identifier automatiquement les bruits dans les garages. La base de données des sons a été enregistrée dans un garage souterrain. Ces sons ont été par la suite labellisés manuellement en deux classes : No Crash, Crash. Les résultats obtenus ont été très satisfaisants.

L. Lin, Ng, H. Siang et Chua [18] ont présenté un nouveau schéma de reconnaissance des événements pour la vidéosurveillance des parcs de stationnements. Le système proposé est basé sur les mélanges des Gaussiennes adaptatifs et l'analyse des composants connectés en vue de modéliser le fond et suivre les objets. L'événement audio est représenté par un vecteur d'attributs contenant des informations dynamiques de la trajectoire et des données contextuelles de l'objet. La classification des événements est effectuée en tenant compte de : (i) la mesure de la similarité du vecteur d'attributs avec la définition étiquetée des événements, (ii) et l'analyse de l'information contextuelle de l'événement détecté. Les résultats obtenus montrent la précision de l'algorithme proposé.

K. Na, Y. Kim, et H. Cha [19] ont décrit la conception, la mise en œuvre et l'évaluation d'un système de surveillance des parcs de stationnement fondé sur les capteurs acoustiques. Le système utilise des nœuds de capteurs équipés de microphones à faible coût pour localiser les événements acoustiques comme les alarmes et les bruits d'accident des voitures. Une fois l'événement acoustique est localisé, les caméras sont dirigées vers la source du son et enregistrent la scène en cours. Les résultats expérimentaux montrent que le système proposé est très performant.

M. Leo et al. [20] ont introduit une technique d'apprentissage basée sur les exemples pour détecter les personnes dans les scènes dynamiques. Une classification de la forme des personnes a été par la suite appliquée. Un réseau neuronal supervisé à trois couches a été utilisé pour classer correctement les objets. Des expériences ont été réalisées sur des séquences d'images réelles acquises dans une aire de stationnement. Les résultats ont montré que la méthode proposée est robuste, fiable, rapide et qu'elle peut être facilement adaptée à la détection de tout autre objet en mouvement.

2.7 Conclusion

Dans ce chapitre, nous avons présenté une vue globale sur les parcs de stationnements et les systèmes de surveillance. Nous avons montré la contribution de la modalité audio pour la

surveillance. Un état de l'art sur les techniques de monitoring des parcs de stationnement a été aussi présenté. Nous avons trouvé que peu de travaux sur la surveillance audio des parkings ont été reportés dans la littérature.

Dans le prochain chapitre, nous proposons un schéma de détection et reconnaissance des sons de l'environnement en vue de surveiller les parcs de stationnement.

Chapitre 3 : Approche proposée pour la reconnaissance en temps réel des sons de l'environnement

3.1 Introduction

Dans ce chapitre, nous présentons trois sections importantes : (i) le schéma global de détection et reconnaissance des sons, (ii) les métriques utilisées pour évaluer les performances des méthodes proposées et (iii) la description du fonctionnement temps réel de notre méthode. Nous nous focalisons dans la première section sur l'approche de l'Intelligence Artificielle centrée sur les données. Une description détaillée des solutions que nous avons adoptée pour la détection et la reconnaissance des évènements acoustiques est présentée dans ce chapitre. Nous terminons ce chapitre par une conclusion.

3.2 Collecte de données

La collecte de données est souvent considérée comme un événement ponctuel et est négligée au profit de la construction d'une meilleure architecture de modèle. En conséquence, des centaines d'heures sont perdues à peaufiner des modèles basés sur des données imparfaites. Selon le pionnier de l'Intelligence Artificiel (IA), Andrew Ng[21] ,« nous devons passer d'une approche centrée sur le modèle à une approche centrée sur les données (*Data centric approach)* [21]. Professeur Andrew [20], a été le premier à utiliser les processeurs graphiques (GPU) pour former des modèles d'apprentissage profond avec ses étudiants de l'université de Stanford à la fin des années 2000 [22]. Il a cofondé Google Brain en 2011, puis a été, pendant trois ans, le scientifique en chef de Baidu, où il a contribué à la création du groupe d'IA du géant chinois de la technologie [23]. La recommandation actuelle du professeur Andrew est d'adopter le Data Centric pour les applications pratiques de l'IA.

a) Intelligence artificielle centrée sur les données

L'IA centrée sur les données (en anglais : Data Centric Artificial Intelligence « DCAI ») est une discipline qui étudie le problème de la construction d'ensembles de données de haute qualité pour l'apprentissage automatique et la création avec succès un système à base d'IA [21]. C'est un nouveau domaine de recherche passionnant dans la communauté de l'IA qui est considéré comme une programmation axée sur les données plutôt que sur le code. Il représente un déplacement de

l'intérêt des mégadonnées vers les bonnes données. Cela implique que cette solution consacre plus de temps à la modification et à l'amélioration systématiques des ensembles de données afin d'améliorer la précision des applications [23]. C'est une approche qui ne met pas nécessairement l'accent sur la taille de l'ensemble de données, mais se préoccupe davantage de la qualité des données. À savoir, les données n'ont pas besoin d'être volumineuses, mais étiquetées correctement et soigneusement sélectionnées pour couvrir les bases [23].

b) Avantages du Data Centric

En adoptant une approche d'IA centrée sur les données, les entreprises de divers secteurs tels que l'automobile, l'électronique et la production de dispositifs médicaux ont constaté des améliorations dans le déploiement de solutions basées sur l'IA par rapport aux implémentations traditionnelles basées sur des règles [21], [22]. La construction de systèmes d'IA avec des données de qualité aide les équipes à atteindre le niveau de performance requis et supprime le temps inutile d'essais et d'erreurs consacré à l'amélioration du modèle sans modifier les données incohérentes [23].

c) Pourquoi l'IA à base du Data Centric est adaptée au développement des systèmes de surveillance des parcs de stationnement ?

Un système de surveillance est souvent mis en place dans un espace très fréquenté tel que les parcs de stationnement. Les agents du poste de contrôle ne peuvent assurer une surveillance permanente des lieux. L'IA centrée sur les données est une approche prometteuse pour le développement des applications pratiques. Elle a la capacité de traiter peu de données et produire des modèles qui d'agissent de manière autonome et efficace. En d'autres termes, cette approche permet de recueillir des informations importantes sur un simple évènement produit. Elle peut améliorer les performances des systèmes de surveillance pour obtenir des résultats plus précis dans un plus court délai. Cependant, les données d'apprentissage doivent être de haute qualité pour assurer de bonnes performances de reconnaissance.

d) Méthodologie proposée pour collecter les sons de l'environnement

Le système de reconnaissance des évènements sonores est fondé sur des sons qui caractérisent les trois classes : cris des humains, bris de glace, et alarme de voitures. Une petite base de données de 150 sons a été créée afin de mettre à l'essai et d'évaluer l'approche proposée. Chaque classe contient 50 sons. Ces sons ont été téléchargés gratuitement à partir de site web :

Sounddogs. Pour notre étude, nous avons enregistré à nouveau les sons dans l'environnement de test pour deux raisons principales : (i) les enregistrements doivent être effectuer avec le même microphone que celui utilisé pour la reconnaissance en temps réel, (ii) les modèles développés doivent tenir compte de l'effet des environnements sur la performance de reconnaissance.

3.3 Schéma global de reconnaissance des sons

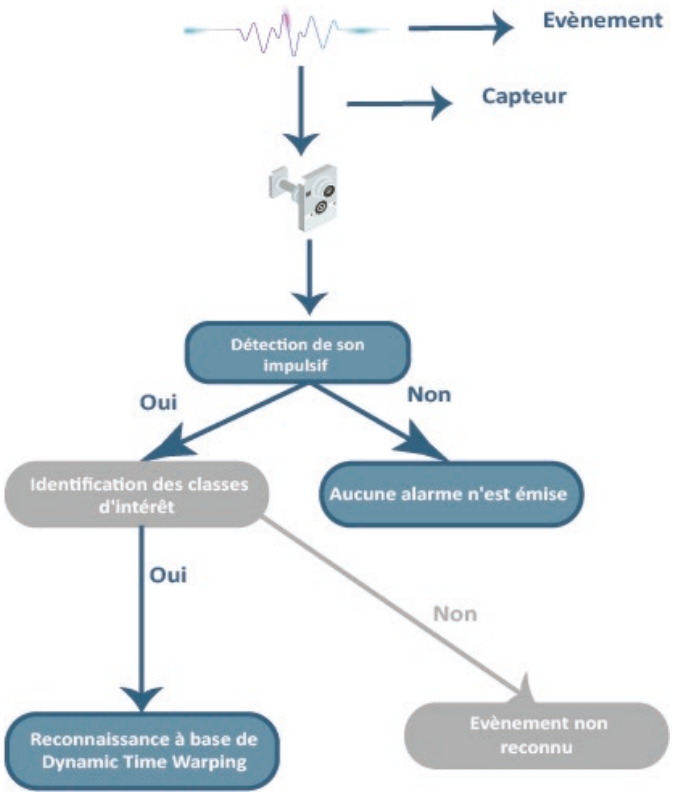

Figure 3.1: Schéma global de reconnaissance en temps réel des sons impulsifs

Notre objectif dans le cadre de ce projet de fin d'étude est de concevoir et implémenter une plateforme de reconnaissance des événements sonores en temps réel en vue de surveiller les parcs de stationnements. Elle est composée de deux étages principaux : (i)la détection des évènements sonores et(ii) de reconnaissance des sons.

La reconnaissance en temps réel des sons impulsifs est schématisée sur la Figure 3.1. Dans notre étude, le module de détection consiste à identifier les évènements acoustiques captés par le microphone. Par la suite, nous nous intéressons à la reconnaissance de trois classes d'intérêt : cris humain, alarmes de voitures et bris de glace. L'événement est classé non important dans le cas où le son impulsif n'appartient pas à ces classes. Cette tâche n'est pas considérée dans notre projet. La solution à ce problème sera proposée dans nos travaux futurs. D'autre part, la

reconnaissance des sons identifie parmi les trois catégories le type exact du son produit. Elle est basée sur deux modèles différents : le Dynamic Time Warping (DTW) et le Long short-term memory (LSTM).

3.4 Détection

Nous nous concentrons dans cette partie sur la détection des évènements acoustiques impulsifs en temps réel. La solution que nous proposons tient compte des facteurs suivants : La complexité algorithmique, la puissance de calcul des ordinateurs, l'environnement acoustique, la qualité des capteurs, l'efficacité de détection (taux d'erreurs) et le fonctionnement en temps réel.

Notre approche est basée sur les méthodes par seuillage qui consistent à comparer une caractéristique du signal audio avec un seuil. Ces méthodes sont moins exigeantes, en termes de complexité de calcul, que celles par classification [15].

La méthode que nous avons utilisée pour détecter le son impulsif est basée sur l'évolution de la variance des séquences de l'énergie à courts termes calculées à partir du signal audio. Cette méthode donne de bons résultats dans des conditions bruitées et répond le mieux aux critères énoncés précédemment.

Les performances de la méthode dépendent de :

- La sélection des paramètres de la méthode
- Le niveau de la pression acoustique des évènements sonores.

Le processus de détection est résumé comme suit :

a) Estimation de l'énergie e(k)

$$e(k) = \frac{1}{N}\sum_{n=0}^{N-1} x^2(n + kN) \quad k = 0,1 \ \dots\dots\dots\dots + \infty \tag{3.1}$$

x(n) : est un signal audio échantillonné à une fréquence Fe ;

k : est l'indice des blocs qui varie de 0 à $+\infty$;

n : est l'indice des échantillons ;

e(k) : est l'énergie du k^{iéme} bloc calculée à partir du signal ;

N : représente la longueur du k^{iéme} bloc.

b) Fenêtrage de la séquence d'énergie e_{win}(j/k)

Avec chaque nouveau block k, on obtient une nouvelle valeur d'énergie e(k). Pour

conserver les dernières valeurs L de e(k), la séquence d'énergie est enregistrée dans une fenêtre dynamique $e_{\text{win}}(\text{j}/k)$[1]. Tel que : $j = 0 \ \ldots . \ L-1, et \ k = 0 \ \ldots . \ +\infty$

La variation des indices « i » et « j » est liée aux valeurs de « k ». En fonction des valeurs « k », nous pouvons distinguer deux régimes [15]:

(i) Régime transitoire

Le but de ce régime est d'initialiser les composantes de la séquence $e_{win}(\text{j/k})$ pour les valeurs de « K<L ».

$$e_{win}(j) = e(i) \qquad \text{Tel que : i, j=0 à K-1.} \qquad (3.2)$$

Dans ce cas, la variation des indices « i et j » est simultanée. L'exemple ci-dessous présente un aperçu sur le calcul des énergies.

Exemple :

$$\text{e}_{\text{win}}(0) = \text{e}\ (0)$$
$$\text{e}_{\text{win}}(1) = \text{e}\ (1)$$
$$\text{e}_{\text{win}}(2) = \text{e}\ (2)$$
$$\text{e}_{\text{win}}(3) = \text{e}\ (3)$$
$$.$$
$$.$$
$$\text{e}_{\text{win}}(\text{L}-1) = \text{e}(\text{L}-1)$$

(ii) Régime permanent

C'est le régime dynamique, son but est de mettre à jour la séquence $e_{win}(\text{j/k})$ pour les valeurs de $K \geq L$ et de décider si la séquence captée est impulsive ou non. Pour chaque nouvel élément de k, les L composantes récentes de la séquence ewin(j/k) sont calculés de la façon suivante :

$$\text{e}_{\text{win}}(\text{j})^2 = \text{e}(\text{i}) \qquad (3.3)$$

tel que : i = k − L + 1 à k, j = 0 à L − 1

[1] La notation e(j/k) signifie que l'indice j dépend des valeurs de k

[2] L'indice k a été omis dans cette équation pour une simplicité de représentation.

L'indice « j » est un entier statique qui varie de 0 à L-1. Toutefois, l'indice « i » est un entier dynamique, car il dépend essentiellement de la variation de « k ». La modification des indices « i et j » est réalisée à la fois.

c) Normalisation de la séquence d'énergie $e_{win}(j/k)$

En fait, la clé de la méthode de détection consiste à normaliser la séquence d'énergie. Afin de standardiser le vecteur $e_{win}(j)$ et le projeter dans une plage de valeurs allant de 0 à 1, l'opération suivante est réalisée :

$$e_{norm}(j) = \frac{e_{win}(j) - \min_j(e_{win}(j=0:L-1))}{\max_j(e_{win}(j) - \min_j(e_{win}(j)))} \quad (3.4)$$

$$j = 0 : L - 1$$

d) Calcul de la variance des fenêtres normalisées

Lorsque la normalisation de la séquence d'énergie est achevée, on calcule la variance de $e_{norm}(j)$ en excluant la dernière composante[15] :

$$var(k) = \frac{1}{L-1}\sum_{j=0}^{L-2}[e_{norm}(j) - \bar{e}_{norm}(k)]^2 \quad (3.5)$$

$\bar{e}_{norm}(k)$: est la valeur moyenne du vecteur $e_{norm}(j)$.

e) Décision

Quand le signal est assez stable, à niveau constant ou légèrement variable, toutes les composantes de $e_{norm}(j)$ sont distribuées entre 0 et 1, ce qui donne une variance relativement importante. Cependant, lorsqu'une impulsion significative se produit, la dernière valeur de la fenêtre augmente à 1, tandis que les valeurs précédentes varient autour de 0. La valeur de la variance diminue.

La variance des fenêtres normalisées constitue un critère très efficace pour détecter une augmentation soudaine du niveau énergétique. Si la variance est sous un seuil « Th » donné, l'évènement détecté est impulsif, sinon aucun évènement précis n'est détecté.

- **Choix et optimisation des paramètres**

Pour améliorer l'efficacité et assurer de bonnes performances de la détection en temps réel, nous proposons d'optimiser les paramètres suivants : (i) la fréquence d'échantillonnage Fe,

(ii) la longueur de bloc N, (iii) la longueur de séquence de puissance L, et (iv) le seuil de décision Th. Nous cherchons le jeu de paramètre qui assure une détection correcte à complexité réduite. Le processus expérimental adopté dans cette étude sera discuté dans le prochain chapitre.

3.5 Reconnaissance

Dans notre projet, nous nous intéressons à la reconnaissance de trois évènements sonores : (ii) cris humains, (ii) bris de glace, et (iii) alarmes de voitures. Nous nous focalisons sur le scenario de reconnaissance en utilisant peu d'échantillons. Le schéma de classification que nous proposons est basé sur la déformation temporelle dynamique (en anglais : Dynamic Time Warping (DTW)). Nous avons choisi ce modèle vu leur efficacité pour le Small-size data. Il est bien connu que lorsque le nombre de données est trop petit, cette méthode est la plus appropriées.

Nous utilisons une technique pour l'extraction d'attributs basée sur les coefficients cepstraux (en anglais : Mel- Frequency Cepstral Coefficients « MFCCs »). La Figure 3.2 schématise notre démarche.

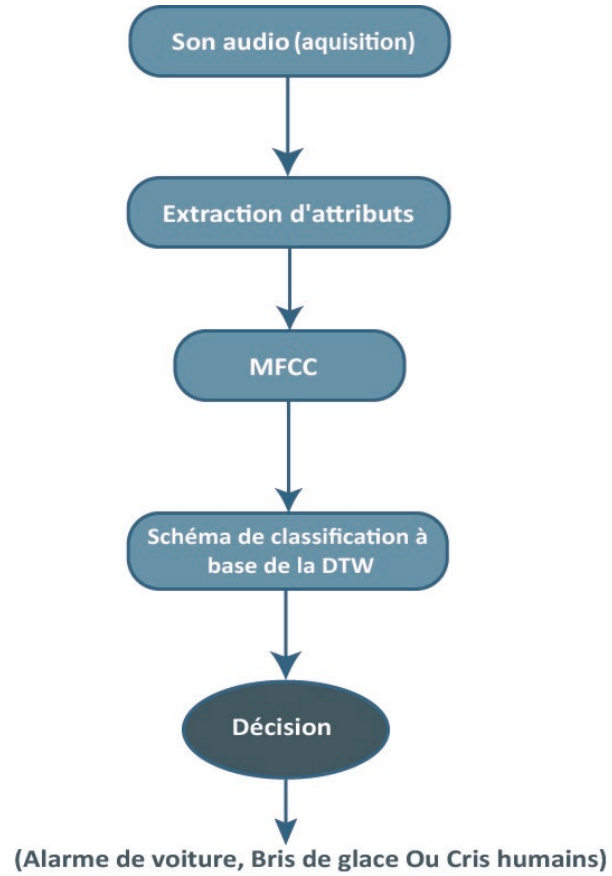

Figure 3.2: Schéma de classification globale de reconnaissance des évènements sonores

a) Extraction d'attributs

L'extraction des caractéristiques acoustiques est une étape fondamentale pour la reconnaissance des signaux. L'objectif est d'extraire une séquence de caractéristiques fournissant l'évolution du signal au cours du temps [24]. Il existe un grand nombre de techniques d'extraction de caractéristiques dans la littérature. Nous nous focalisons dans notre étude sur les MFCCs et les LPCs.

- **MFCC**

La méthode la plus populaire pour extraire les caractéristiques du signal audio est le calcul des coefficients MFCCs. C'est l'une des méthodes les plus utilisées pour la reconnaissance vocale[25]. La figure 3.3 présente le schéma fonctionnel du processus d'extraction des caractéristiques.

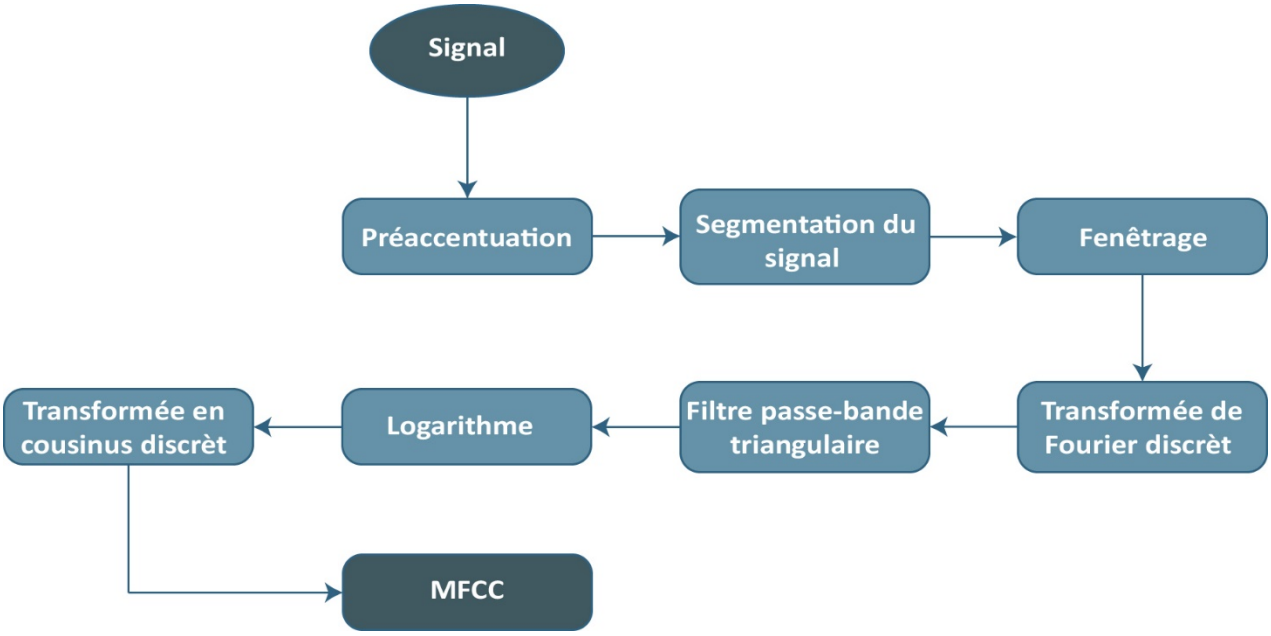

Figure 3.3: Schéma fonctionnel pour l'extraction du MFCC.

Le calcul des MFCC est effectué comme suit [26] :

1) -Préaccentuation

Cette étape consiste à augmenter l'énergie du signal à une fréquence plus élevée par un filtre passe-haut :

$$s1(n) = s(n) - a * s(n-1) \qquad (3.6)$$

S1(n) : Signal de sortie.

Le facteur de préaccentuation (a) est pris entre 0.9 et 1 (souvent 0.95).

2) Segmentation du signal

Dans cette étape, le signal est devisé en petites trames d'échantillons dont la longueur doit être comprise entre 20 ms et 30 ms pour garantir la quasi-stationnarité du signal.

3) Fenêtrage

Chaque trame doit être multipliée par une fenêtre de pondération de type Hamming pour garder la continuité des premiers et derniers points de la trame. L'opération de fenêtrage consiste à multiplier les échantillons du signal s(n), tel que n = 0, … N-1, par la fenêtre choisie.

$$s2(n) = s1(n) * w(n, a)$$

$w(n, a)$ est la fenêtre Hamming définie par l'équation ci-dessous :

$$\mathrm{W}\,(\mathrm{n}, \mathrm{a}) = \ (1 - \mathrm{a}) - \ \mathrm{a}\cos\left(\frac{2\pi \mathrm{n}}{(\mathrm{N}-1)}\right), \ \ 0 \ \leqq \ \mathrm{n} \ \leqq \ \mathrm{N} - 1 \ . \qquad (3.7)$$

$w(n, a)$: la fenêtre de Hamming.

N : nombre d'échantillons dans chaque trame.

4) Transformée de Fourier discrète

La transformée de Fourier discrète permet de convertir le signal S_2(n) du domaine temporel au domaine fréquentiel. Cette dernière est calculée par l'algorithme FFT (Fast Fourier Transform).

$$Y(f)=FFT(S2(n)) \qquad (3.8)$$

Où : Y(f) : est la transformée de Fourier discrète du signal s_2(n).

5)-Filtres passe-bande triangulaires

Un ensemble de filtres passe-bandes triangulaires est utilisé pour faire rapprocher la résolution des fréquences perçues par l'oreille humains (l'échelle de Mel). Pour passer de l'échelle fréquentielle linéaire à l'échelle de Mel, on applique l'équation suivante :

$$\mathrm{f_{mel}} = 2595 \log_{10}\left(1 + \frac{\mathrm{f}}{700}\right) \qquad (3.9)$$

6) Logarithme

Les humains n'entendent pas le son dans une échelle linéaire, donc la prochaine étape est de calculer le logarithme de l'amplitude du spectre.

7) Transformée en cosinus discrète

Il s'agit du processus de conversion du logarithme de spectre au domaine cepstral à l'aide de la DCT (Discreet Cosine Transform). Le résultat de la conversion donne l'ensemble de coefficients MFCC également appelés vecteurs acoustiques.

On peut calculer les dérivés des MFCC :

- Les dérivées premières indiquent la vitesse de variation de ces vecteurs au cours du temps,
- Les dérivées deuxièmes fournissent des informations sur l'accélération du signal.

b) Schémas de classification

Le schéma de classification que nous avons proposé est basée sur la programmation dynamique. Nous présentons dans la première sous-section des définitions sur la programmation dynamique. Par la suite, nous schématisons l'approche proposée.

○ Programmation dynamique

La première méthode de classification que nous adoptons est basée sur la déformation temporelle dynamique (en anglais : *Dynamic Time Warping* (DTW)). C'est l'une des techniques de programmation dynamique de type « Template Matching ».

La DTW est un algorithme très utilisé pour la classification des séries temporelles. Il permet de mesurer la similarité entre deux séquences temporelles qui peuvent varier au cours du temps[27], [28].

Soit P= {P_1, P_2, P_3 . . .., P_i} Q= {Q_1, Q_2, Q_3 . . .., Q_j} deux séries temporelles et la variable DTW (i, j) désigne la distance entre P_i et Q_j.

La formule de programmation dynamique pour le calcul de la DTW est donnée par l'équation suivante :

$$DTW(i,j)\begin{cases} 0 & si\ i=0\ et\ j=0, \\ \infty & si\ i=0\ ou\ j=0, et\ i\neq j \\ dis(Pi,Qj)\begin{cases} DTW(i-1,j) \\ DTW(i,j-1) \\ DTW(i-1,j-1) \end{cases} & si\ 1\leq i\leq m\ et\ 1\leq j\leq n \end{cases} \quad (3.10)$$

La variable d (P_i, Q_j) représente la distance entre P_i et Q_j.

Prenons deux séries temporelles avec une longueur différente A et B : A = {0,1, 2, 3, 5, 5, 5, 6} B = {1, 1, 2, 2, 3, 5}, La matrice sera remplie à l'aide de la formule récursive pour DTW (i, j) ci-dessus (Tableau 3.1).

Tableau 3-1: Calcul de la DTW (les cases grises représentent le chemin de l'alignement temporel dynamique)

B A		1	1	2	2	3	5
	0	∞	∞	∞	∞	∞	∞
0	∞	0	0	1	2	4	8
1	∞	1	1	0	0	1	4
2	∞	3	3	1	1	0	2
3	∞	7	7	4	4	2	0
5	∞	11	11	7	7	4	0
5	∞	15	15	10	10	6	0
5	∞	20	20	14	14	9	1

La distance DTW entre A et B est M [5,5] =1(la solution optimale).

Le but de DTW est de trouver un chemin de déformation W défini par Wk = (i, j). Le chemin global dedéformation est : $W = \{w1, w2, \ldots, wk, \ldots, wk'\}$, Où max $(m, n) \leq k \leq m + n - 2$. Les contraintes suivantes doivent être appliquées :

- Conditions aux limites : $w1$ = (1,1) et wk' = (m, n).
- Contrainte locale : pour tout nœud donné (i, j) dans le chemin, les nœuds d'éventail possibles sont limités à (i − 1,), (i, j – 1) et (i − 1, j – 1). Il garantit un chemin monotone non décroissant.

Lorsque la taille des donnés est énorme, le calcul de la distance à base de la DTW prend du temps. Pour faire face à ce problème, Itakura [29] et Sakoe-Chiba [30] ont ajouté une notion de

déformation temporelle dynamique avec fenêtre en ajoutant des contraintes de déformation supplémentaires qui permet au chemin d'être plus proche de la diagonale et d'éviter le chemin indésirable [31]. L'algorithme d'Itakura [29] se base sur la fonction de pente S, et le chemin de déformation est délimité par deux pentes S et 1/S. L'algorithme de Sakoe et Chiba utilise une diagonale de largeur fixe. Ces contraintes proposées par Sakoe et Itakura sont illustrés sur la Figure 3.4. Dans notre travail, nous avons utilisé la méthode de Sakoe-Chiba qui est basée sur une diagonale de largeur fixe.

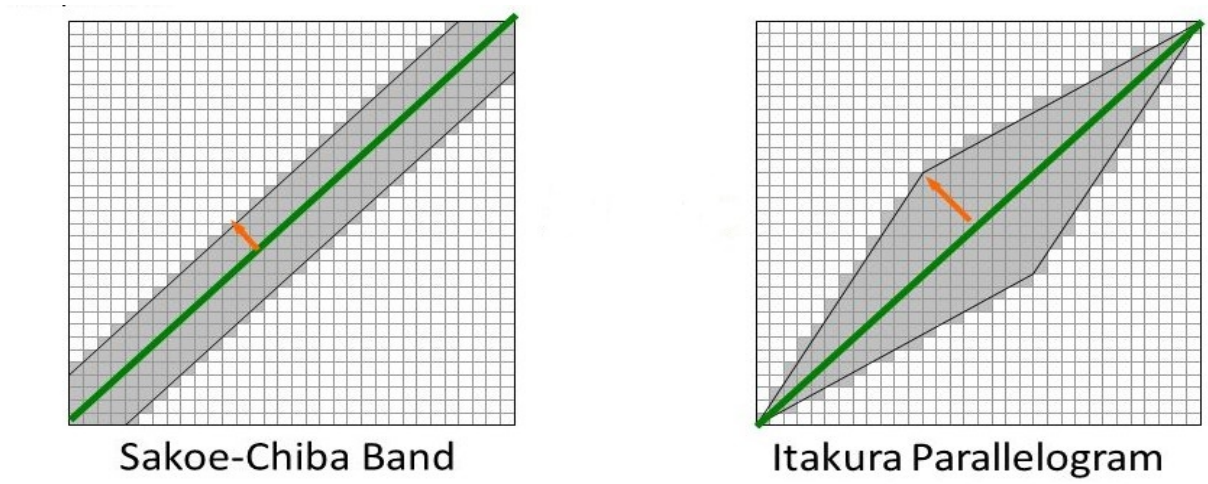

Figure 3.4 : Fenêtres de déformation de la DTW avec la bande de Sakoe-Chiba et Parallélogramme d'Itakura [32].

- **Schéma proposé**

Cette section met l'accent sur le schéma de classification proposé pour la reconnaissance des événements sonores. Notre démarche consiste à utiliser un codebook de références de 30 triplets différents pour assigner le son de test à la classe la plus probable. Chaque triplet est composé de trois matrices correspondant aux valeurs numériques d'attributs extraits à partir des trois séquences de références (cris humains, bris de glace, et alarmes de voitures). Le signal à analyser est comparé avec chacune des références et est classé en fonction de sa proximité de l'une des références du codebook.

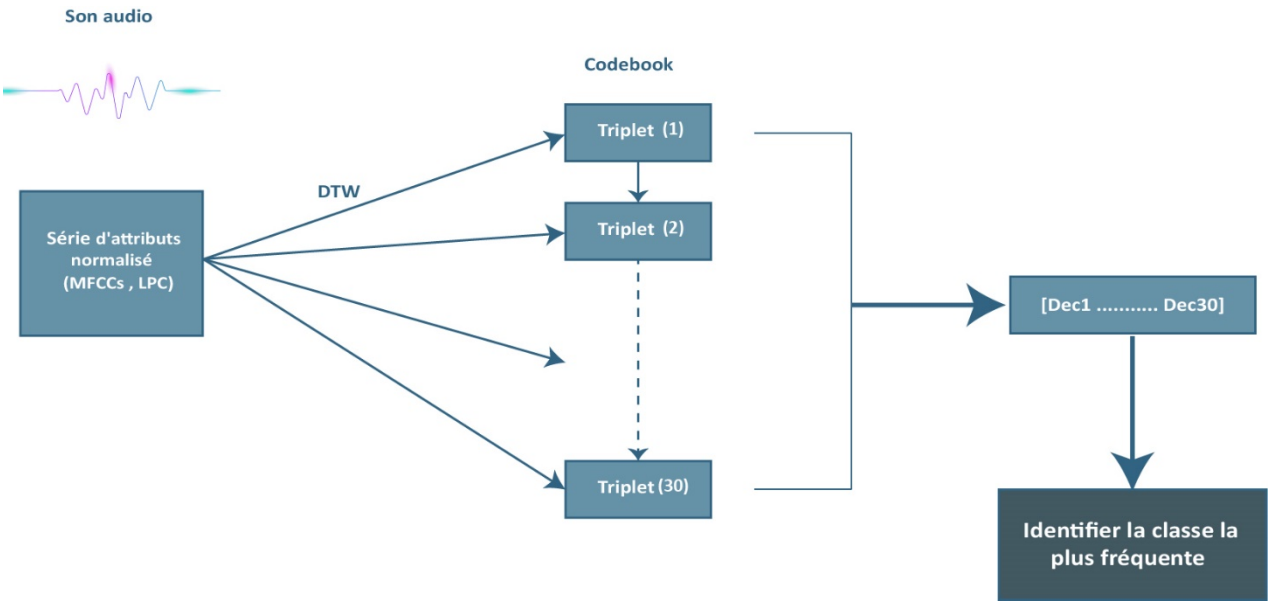

Figure 3.5: Schéma de classification à base de DTW.

La procédure de test présentée dans la Figure 3.5 se déroule comme suit : (1) la lecture de l'évènement audio en sélectionnant une partie de l'onde acoustique (de durée égale à D), (2) le choix du nombre de MFCC et le calcul des valeurs numériques (3) la construction de matrice d'attributs M qui possède le nombre de lignes correspond à la longueur de la séquence, et un nombre de colonnes correspond au nombre d'attributs,(4) calcul de la distance DTW entre la matrice M et le triplet T(i) pour i=1 jusqu'à 30,(5) Identification de la classe la plus probable.

3.6 Métriques d'évaluation

Les performances de la détection sont évaluées en utilisant les métriques suivantes [9] :

- Le taux de vrais positifs (TVP) : On peut l'évaluer à l'aide de l'équation suivante.

$$\mathrm{TVP} = \frac{\mathrm{ECD}}{\mathrm{TEP}} \tag{3.17}$$

ECD : le nombre des événements correctement détectés.

TEP : le nombre de tous les évènements acoustiques qui correspondent à des sons impulsifs.

- Le taux de faux positifs (TFP) :

$$\mathrm{TFP} = \frac{\mathrm{EDI}}{\mathrm{TED}} \tag{3.18}$$

EDI : le nombre des événements détectés d'une maniéré incorrecte.

TED : le nombre de tous les évènements détectés.

- Le taux de faux négatifs (TFN) : également appelé le taux de faux rejet, est calculé en utilisant l'équation suivante :

$$\mathrm{TFN}=1\text{-}\mathrm{TVP} \tag{3.19}$$

- Le rappel ou sensibilité :(Recall en anglais), est défini par le nombre des événements correctement détectés par rapport au nombre de tous les évènements. On peut l'estimer en utilisant l'équation suivante :

$$\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3.20}$$

- La précision : (Precision en anglais), Elle correspond au taux de prédictions correctes parmi les prédictions positives. Cette mesure est aussi appelée valeur prédictive positive [33], [34]:

$$\mathrm{Precision}\ =\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{3.21}$$

Pour comparer l'impact des représentations citées ci-dessus sur la performance de l'algorithme de détection, nous avons utilisé le F1-score qui correspond à la moyenne harmonique du taux de précision et du rappel.

$$F1-Score=2*\frac{Précision*Recall}{Précision+Recall} \tag{3.22}$$

Les méthodes de reconnaissance sont évaluées avec deux mesures de performance standard : (i) la matrice de confusion et (ii) le taux de bonne classification (TBC).

- Le taux de bonne classification (TBC) est l'une des mesures les pluscouramment utilisées pour évaluer les performances des classifieurs [34].

Cette métrique est définie comme le rapport entre les échantillons correctement classés et le nombre total d'échantillons :

$$TBC = \frac{TP + TN}{TP + FP + TN + FN}$$

- Matrice de confusion : C'est une matrice qui permet de mesurer la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée.

La matrice de confusion pour un problème de classification à trois classes (A, B et C) est schématisée sur la Figure 3.6 [34].

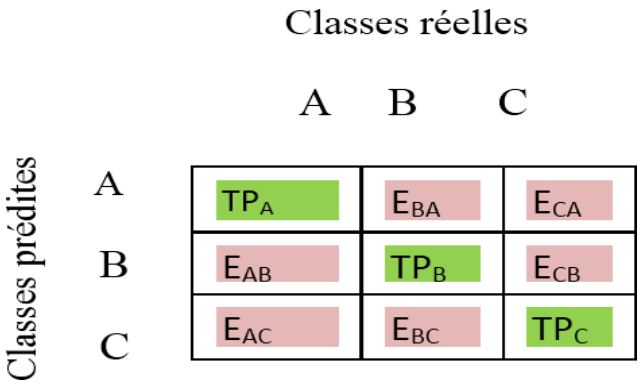

Figure 3.6: Structure de la matrice de confusion.

Dans La matrice on a :

- A, B et C sont les classes.
- La diagonal contient les échantillons correctement classifiés (TP_A, TP_B, TP_c).
- E_{AB} est le nombre d'échantillons de la classe A qui ont été incorrectement classés dans la classe B (mal classés).
- Le faux négatif dans la classe A (en anglais, False Negative noté :FN_A) est la somme de E_{AB} et E_{AC} ($FN_A = E_{AB} + E_{AC}$). Il correspond à la somme de tous les échantillons de la classe A qu'ont été incorrectement classés en classe B ou C (FN située dans une colonne).
- Le faux positif (en anglais, False Positive noté : FP), toute classe prédite qui se trouve dans une ligne, représente la somme de toutes.

3.7 Fonctionnement en temps réel des techniques implémentées

La solution que nous utilisons va être implémentée sur un PC ordinaire équipée d'un processeur Core i7 et de 8 G de RAM. Le programme fonctionne sous Windows 10. L'acquisition

de données audio est réalisée d'une maniéré sans fil en utilisant un microphone incorporé dans les smartphones. En effet, ces microphones sont de type MEMS qui possèdent des bonnes performances. Ce type possède généralement des bonnes performances. Nous avons utilisé le microphone du smartphone Samsung J4 core dans nos expériences. Le logiciel Wo mic a été exploité pour transformer notre smartphone en un microphone sans fil.

3.8 Conclusion

Dans ce chapitre, nous avons présenté l'approche d'IA centrée sur les données qu'on a adopté pour concevoir notre système de surveillance. Par la suite, nous avons présenté les méthodes proposées pour la détection et la reconnaissance des sons impulsifs. Par la suite, nous avons cité les métriques nécessaires pour évaluer les performances de ces méthodes. Dans le chapitre suivant, nous présenterons les résultats expérimentaux que nous avons obtenus.

Chapitre 4 : Implémentation et résultats expérimentaux

4.1 Introduction

Après avoir défini notre approche pour la reconnaissance des sons de l'environnement, nous présentons dans ce chapitre les résultats expérimentaux. Ce chapitre décrit aussi les outils de développement adopté et le langage de programmation utilisé. Nous terminons ce chapitre par une conclusion.

4.2 Logiciels de développement

Dans cette partie, nous allons présenter les outils de développement que nous avons utilisé pour la réalisation de notre plateforme de reconnaissance des événements sonores en temps réel.

4.2.1 Python

Python est un langage de programmation interprété à usage général, interactif, orienté objet et de haut niveau. Il a été créé par le programmeur 'Guido van Rossum' en 1991. La syntaxe élégante et le typage dynamique de Python, ainsi que sa nature interprétée, permettent au Python de devenir un langage idéal pour le développement rapide d'applications dans de nombreux domaines. Python est un langage libre placé sous licence PSFL (Python Software Foundation License), qui peut s'utiliser dans de nombreux contextes et, s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées [35], [36].

4.2.2 Anaconda

Anaconda est une distribution open-source pour python et R. Avec la disponibilité de plus de 300 bibliothèques pour la science des données, elle facilite la gestion et le déploiement des packages, livré avec une grande variété d'outils pour recueillir facilement des données de diverses sources en utilisant divers algorithmes. Anaconda aide à obtenir une configuration d'environnement facile à gérer qui peut déployer n'importe quel projet en un seul clic [37], [38].

4.2.3 Jupyter Notebook

Jupyter Notebook est une application Web Open Source permettant de créer et de partager des documents contenant du code (exécutable directement dans le document), des équations, des images et du texte. Avec cette application il est possible de faire du traitement de données, de la modélisation statistique, de la visualisation de données et du Machine Learning. Elle est disponible par défaut dans la distribution Anaconda [39].

4.2.4 Pycharm

PyCharm est un environnement de développement intégré (IDE) Python dédié qui fournit une large gamme d'outils essentiels pour les développeurs Python, étroitement intégrés pour créer un environnement pratique pour le développement productif Python , Web et science des données. Il offre une saisie de code intelligente, des inspections de code, une mise en évidence des erreurs, et des correctifs rapides [40], [41].

4.2.5 Qt Designer

Qt Designer est l'outil Qt pour concevoir et construire des interfaces utilisateur graphiques (IUG) avec Qt Widgets. Vous pouvez composer et personnaliser vos fenêtres ou boîtes de dialogue de manière à ce que vous voyez est ce que vous obtenez (WYSIWYG), et les tester en utilisant différents styles et résolutions. Les widgets et les formulaires créés avec Qt Designer s'intègrent parfaitement au code programmé, en utilisant les signaux de Qt et le mécanisme de slots, de sorte que vous pouvez facilement attribuer un comportement aux éléments graphiques. Toutes les propriétés définies dans Qt Designer peuvent être modifiées dynamiquement dans le code. En outre, des fonctionnalités comme la promotion de widget et les plugins personnalisés vous permettent d'utiliser vos propres composants avec Qt Designer [42].

4.2.6 Google colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique.

Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur [39].

4.2.7 GoldWave

GoldWave est un éditeur audio numérique professionnel, un lecteur, un enregistreur, un analyseur et un convertisseur. C'est un logiciel qui permet d'écouter, de visualiser et d'analyser des fichiers enregistrés simples développé par GoldWave Inc lancé pour la première fois en Avril 1993. Les autres fonctionnalités du programme incluent [43], [44]:

- Affichage stéréo de fichiers simples ou doubles
- Insertion de silence, suppression de bruit, réduction du bruit, inversion des données, filtres permettant de restaurer et de remasteriser l'audio
- Affichage de deux fichiers côte à côte à des fins de comparaison
- Fonctionnalités de zoom avant et arrière
- Mesure précise des informations de synchronisation
- Affichages spectraux

Figure 4.1: GoldWave [45]

4.2.8 Wo mic

WO Mic est un logiciel informatique avec deux plates-formes différentes. La première de ces plateformes est une application Windows utilisée sur ordinateur comme client (Wo Mic Client). La

seconde plateforme est une app Android utilisé sur un smartphone sous forme de serveur (serveur Wo Mic). Son rôle principal est de transformer un appareil Android en tant que microphone pour n'importe quel ordinateur. Les deux versions, Windows et Android, de WO Mic ont une interface facile à utiliser. Cela permet de facilement connecter le micro de ton appareil Android à ton PC. En plus de cela, Il y a trois façons de connecter le smartphone à l'ordinateur : Wi-Fi (fidélité sans fil), Bluetooth et USB. En ce qui nous concerne, nous avons utilisé le Wi-Fi.

4.2.9 Sonomètre

Le sonomètre est un dispositif de mesure du niveau de bruit acoustique de l'environnement nommé aussi décibel mètre. L'unité de mesure est le dB SPL (Decibel Sound Pressure Level). Dans notre projet, nous avons utilisé le KMOON GM1352[46].

Figure 4.2 : Sonomètre

4.3 Corpus de test

4.3.1 Détection

4.3.1.1 Données Clean

La méthode que nous avons proposée pour la détection des sons impulsifs a été étudié avec un corpus sonore de 200 sons impulsifs. On a choisi des armes à feu de type pistoles pour les tests. Ce corpus a été téléchargé à partir d'une base de données audio qui s'appelle Sounddogs [47], comportant environ 694395 fichiers couvrant toutes les catégories de son (rires, sons de guitares, cris mâle ou femelle, portes fermées, explosions, armes à feu, animaux, etc.). Ces sons peuvent être mono ou stéréo avec une fréquence d'échantillonnage de 11025 Hz (Fe : le nombre

d’échantillon par seconde). Les sons de cette base de données diffèrent au niveau de volume. Certains sons ont un faible volume, d'autres un moyen et certains d'entre eux ont un volume élevé. Pour cela, nous avons normalisé les amplitudes des 200 sons en utilisant l’option Dynamic Range de GoldWave.

4.3.1.2 Données environnementales

À l'aide d’un ordinateur de bureau équipé de haut-parleurs de haute qualité, les fichiers audio sont générés. Le microphone (intégré dans le smartphone) a été placé à une distance de 4 m des haut-parleurs dans l’environnement intérieur (la surface de la salle est d’environ 30 m2) et connecté sans fil à un ordinateur portable. Les fichiers audio qui sont enregistrés sur un ordinateur de bureau sont lus l’un après l’autre. La durée de silence entre chaque fichier a été fixée à 6s. Cette procédure est répétée 3 fois pour obtenir des séquences enregistrées respectivement à 70, 80 et 90 dBSPLs (Seq (1), Seq (2) et Seq (3)). La variation de la pression acoustique est réalisée en changeant le volume du haut-parleur jusqu’à l’obtention du niveau sonore souhaité. Les mesures ont été effectués en utilisant un sonomètre professionnel. Les trois séquences audio sont étiquetées manuellement pour identifier les instants de départ des événements impulsifs et enregistrés respectivement en ref (1), ref (2) et ref (3). Les références sont utilisées pour calculer les erreurs de détection [9].

4.3.2 Reconnaissance

4.3.2.1 Données Clean

La méthode proposée pour reconnaître le type des événements sonores a été testée sur un corpus de 150 sons. Tel qu'indiqué dans le chapitre précédent, nous avons sélectionné trois catégories de sons : cris des humains, bris de glace, et alarme de voitures. Chaque classe contient une cinquantaine de sons. Le corpus est téléchargé gratuitement depuis les sites Web Sounddogs et YouTube.

Le prétraitement du corpus a été réalisé grâce au logiciel GoldWave. La fréquence d'échantillonnage sonore est de 44100 Hertz. Les silences ont été éliminés manuellement. Nous avons modifié le début et la fin de chaque son et pour obtenir des sons d'amplitudes comparables, nous nous sommes servis de l'option « Dynamic Range Equalization » de GoldWave.

4.3.2.2 Données environnementales

Il s'agit du même corpus de test mais enregistré d'une autre façon dans des conditions différentes. L'enregistrement des 150 sons doit être effectué à l'aide du même microphone que nous prévoyons utiliser dans la phase de traitement en temps réel. Nous avons utilisé un PC portable équipé de haut-parleurs pour lire les fichier audio. Le microphone (intégré dans le smartphone) a été placé à une distance de 1.5m des haut-parleurs de PC dans l'environnement intérieur (la surface de la pièce est d'environ 20 m²).

Nous avons fusionné les 150 fichiers de sons de la base de données clean dans un seul audio. La durée de silence entre chaque fichier a été fixée à 6s en utilisant le logiciel GoldWave. Les formes d'ondes acoustiques, qui sont générées par des haut-parleurs, sont enregistrées simultanément à l'aide du microphone. La séquence audio obtenue est enregistrée sur le disque dur de l'ordinateur portable. Les sons ont été sauvegardés à une Fe de 16000 Hz.

4.4 Lieu de test et matériel utilisé

4.4.1 Lieu

Nos expériences se sont déroulées dans une pièce fermée d'une surface de 22 m² avec un bruit de fond de 32dB mesuré avec le sonomètre KMOON. La pièce est équipée d'un(i) climatiseur, (ii) un bureau, (iii) une chaise, (iv) et une armoire.

4.4.2 Matériel

Nous avons utilisé le matériel suivant dans notre expérience :

Pc-potable :

- Comprend l'application de notre projet,
- Génère les sons impulsifs via les haut-parleurs,

Smartphone :

- Joue le rôle d'un microphone sans fil.

Premièrement, à l'aide d'un logiciel Wo Mic, nous avons converti notre smartphone en un microphone sans fil relié à un ordinateur portable, comme l'illustre la figure 4.3.

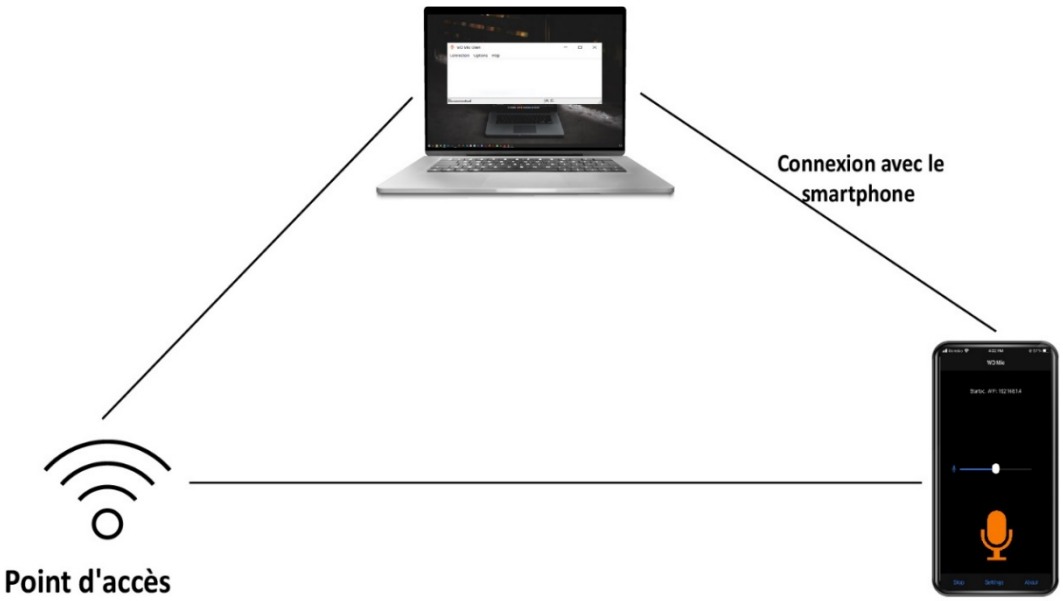

Figure 4.3 : Connexion entre le Pc portable et le Smartphone à l'aide de Wo Mic.

Figure 4.4 : Disposition du matériel de test

4.5 Résultats de la détection

4.5.1 Scénario de test

Cette partie expérimentale a pour objectif de créer un scénario d'événements impulsifs à l'intérieur d'une pièce fermée. Ces évènements sont utilisés pour la sélection des paramètres algorithmiques. Nous avons utilisé l'ensemble des sons décrit dans la section 4.3.1.2 pour tester la détection en temps réel.

Le smartphone a été placé sur l'armoire. L'ordinateur portable (source de sons impulsionnels) a été placé sur le bureau à une distance d'environ quatre mètres du smartphone (Figure 4.4). La détection automatique des bruits impulsifs, à base de la méthode mentionnée dans le chapitre précédent, a été implémentée sous PYTHON.

4.5.2 Optimisation offline des paramètres de l'algorithme

Les paramètres de l'algorithme qui doivent être optimisés pour améliorer l'efficacité de la détection sont : (i) fréquence d'échantillonnage Fe, (ii) longueur de bloc N, (iii) longueur de séquence de puissance L, et (iv) seuil de décision Th.

En règle générale, plus les fréquences d'échantillonnages sont élevées, meilleure est la qualité audio. Cependant, l'augmentation de la valeur de Fe augmente le nombre d'échantillons de la forme d'onde audio, ce qui entraîne une augmentation de la complexité de calcul de la puissance. Dans notre expérience, nous avons choisi un faible taux d'échantillonnage de 11025 Hz puisque les données audios que nous avons téléchargées depuis le site Sounddogs sont présélectionnés à cette valeur.

La valeur de N doit satisfaire les conditions suivantes : (i) elle doit être assez basse pour réduire la complexité de calcul et être appropriée pour l'exécution en temps réel, (ii) inversement, si la valeur choisie est trop basse, trop de détails inutiles dans la séquence de puissance peuvent survenir ; qui peut perturber le processus de détection. Nous avons effectué plusieurs tests empiriques pour trouver le meilleur choix en tenant compte de l'exécution en temps réel et de l'exactitude de la décision. Nous avons constaté qu'une valeur de 220 échantillons (20ms) est un choix approprié.

La longueur de la séquence d'énergie **L** doit être choisie de manière à ce que la durée de l'événement détecté est suffisamment large pour garantir une bonne performance de reconnaissance (le deuxième étage du système de surveillance audio qui vient en suite du

processus de détection). Cette valeur dépend également de la longueur du bloc. Une valeur de **L**=20 qui produira un signal impulsif d'une durée de 0.4 secondes. Cette valeur est considérée comme appropriée pour la configuration logicielle et matérielle de l'ordinateur portable que nous utilisons.

Le choix du seuil de décision **Th** est une étape très importante dans la détection des événements impulsifs. D'après notre étude expérimentale, nous avons constaté que la valeur de Th peut varier entre 5*10-5 et 0,030. La limite supérieure de cet intervalle indique la variance de la séquence de puissance lorsqu'aucun SI ne se produit. La borne inférieure désigne le seuil de blocage pour lequel aucun événement impulsif n'est détecté quelle que soit son intensité. Le choix d'un Th optimal dépend de deux critères principaux. (i) Le niveau de pressoir sonore des données audio (ii) et le type de perturbations environnementales (bruit). La satisfaction de ces critères exige l'utilisation de données audio provenant de plusieurs sources acoustiques. Le seuil de décision que nous avons trouvé dans notre expérience est Th=0.00305.

Afin d'optimiser les paramètres soulignés ci-dessus, nous avons sélectionné un ensemble de valeurs pour chaque paramètre compris dans un intervalle, et on a obtenu 1500 combinaisons. Quelques combinaisons obtenues sont représentées respectivement avec des graphe 3D dans les figures 4.5 et 4.6.

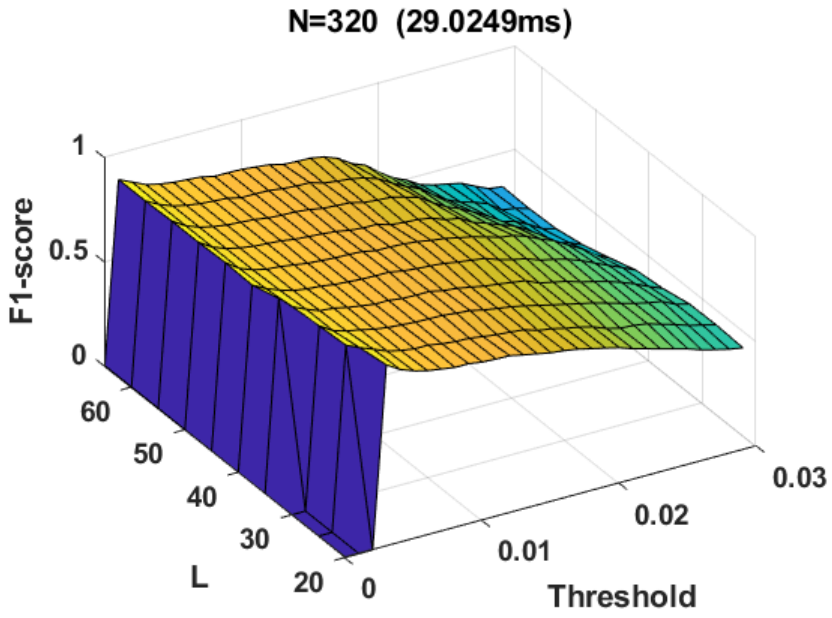

Figure 4.5 : Représentation 3D d'une des combinaisons génères avec une longueur de bloc N=320.

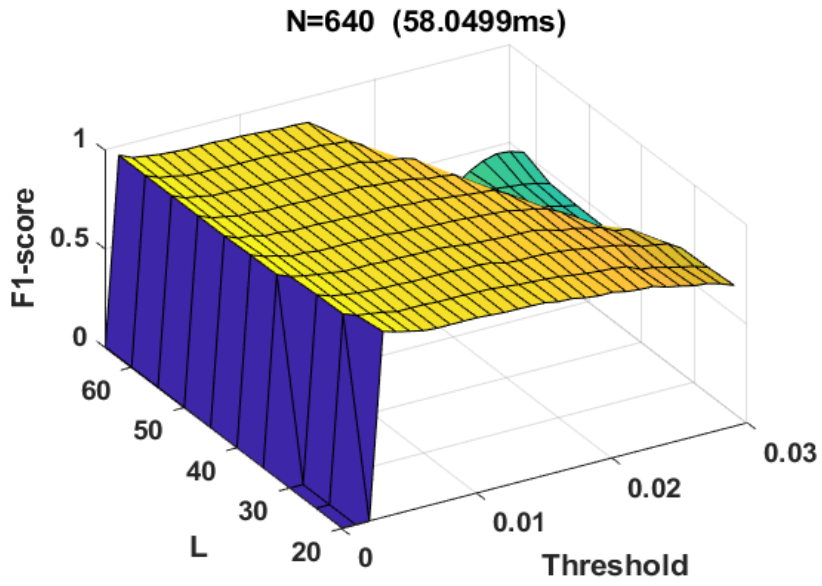

Figure 4.6 : Représentation 3D d'une des combinaisons génères avec une longueur de bloc N=640.

Les valeurs du N, L et Th utilisées dans notre expérience sont représentées dans le tableau suivant (tableau 4.1) :

Tableau 4-1 : les valeurs des paramètres sélectionnés.

Paramètres	**Valeurs utilisés**
N	[220,320,640,1200,2500]
L	[20,25,30,35,40,45,50,55,60,65,70]
Th	[0.00005,0.00105,0.00205,0.00305]

Par la suite, nous avons extrait la combinaison qui donne la meilleure valeur sur les1500 combinaisons précédemment décrites. Le choix de combinaison est effectué principalement par rapport à les métriques suivantes : le taux de vrais positifs (TVP), le taux de faux positifs (TFP) et la mesure F1-score. Nous nous sommes basés sur une procédure de sélection des paramètres optimaux pour lesquels les TVP sont maximisés et les TFP sont minimisés ce qui implique la mesure F1-scores la plus élevée, le résultat est présenté par le Tableau 4-2 et le graphe 3D de la figure1 qui illustrent la combinaison qui donne la meilleure performance du détecteur de sons impulsifs.

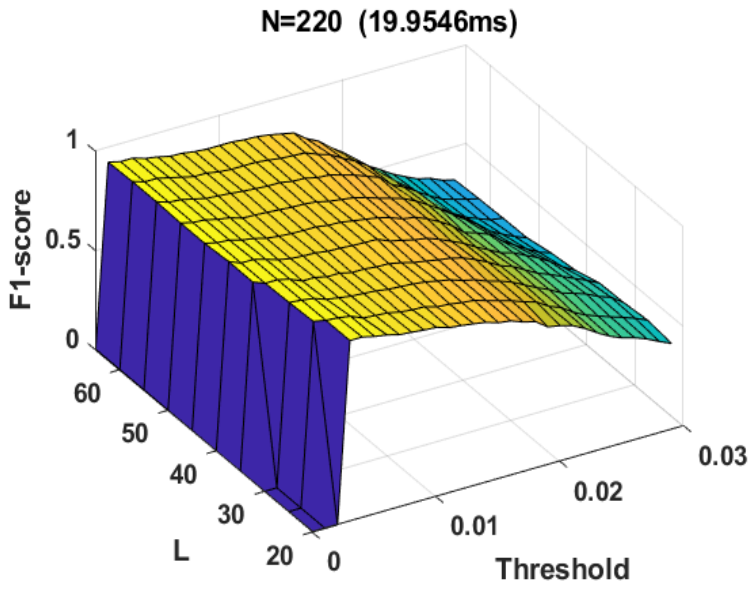

Figure 4.7 : Représentation 3D des paramètres optimaux.

Tableau 4-2 : la meilleure combinaison obtenue

N	L	Threshold	TVP	FPR	F1-score
220	20	0.00305	91٪	9٪	91٪

4.5.3 Test online de la détection

En temps réel, les paramètres optimaux obtenu offline sont utilisé. Nous avons élevé la Fe à 16000 HZ puisque le module de détection va être utilisé pour détecter une portion de signal qui va être reconnu par la suite. Cependant, l'augmentation de la valeur de Fe augmente le nombre de composantes fréquentielle, ce qui entraîne un taux de reconnaissance acceptable.

Pour mettre notre travail à la disposition des utilisateurs, nous avons développé une application en utilisant le Qt Designer. C'est un logiciel de surveillance audio en temps réel activé en permanence dans un parc de stationnement. Lorsqu'un événement impulsif se produit, le logiciel de surveillance déclenche un signal visuel et enregistre également le nombre des événements impulsifs détecté par le système. En outre, il offre la possibilité de réajuster les paramètres algorithmiques du détecteur.

Dans notre expérience, nous avons prévu un scénario simple avec des tests environmentaux en utilisant un son impulsif réel. Le logiciel a été testé dans la même pièce que celle décrite à la section

4.4.1 en milieu intérieur dans des conditions bruités (par exemple : les personnes qui parlent dans l'autre pièce). Ci-dessous, nous présentons l'interface de l'application développé.

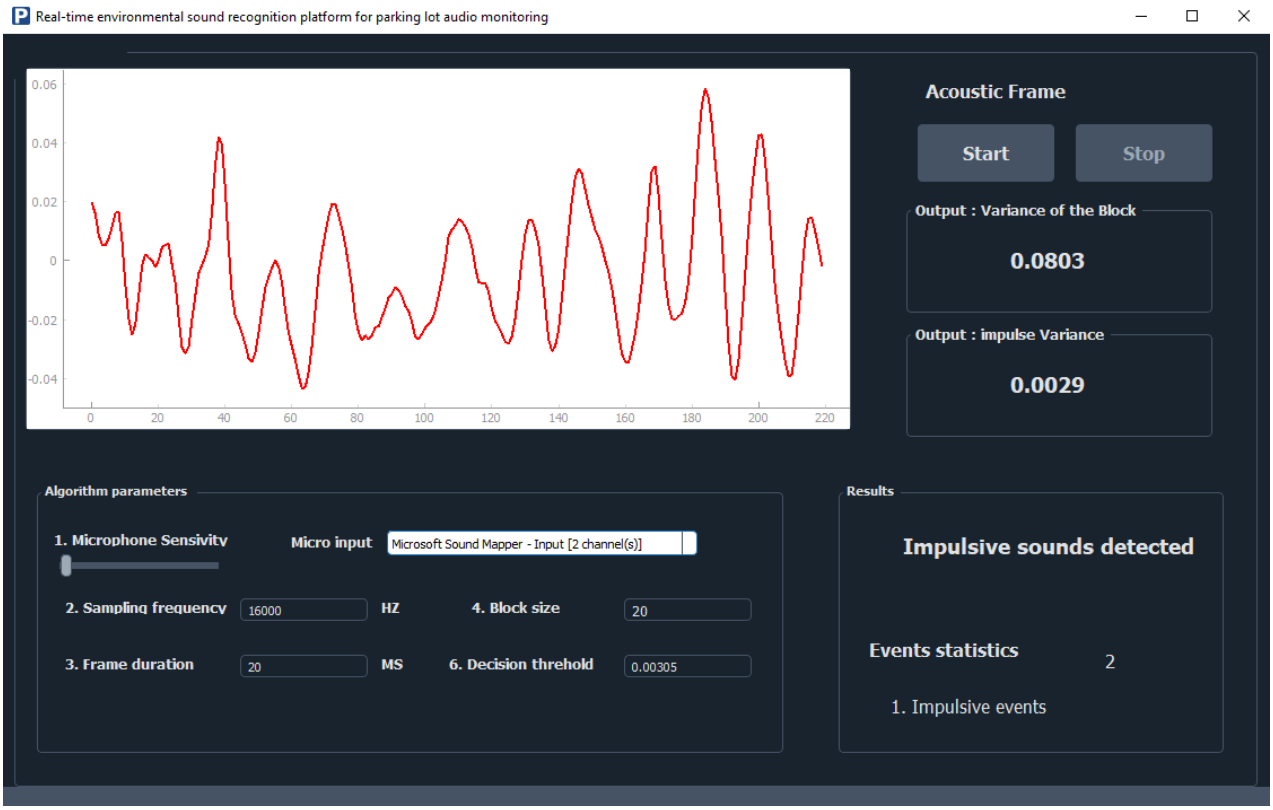

Figure 4.8 : Interface pour la détection en temps réel des sons impulsifs.

4.6 Résultats de la reconnaissance

4.6.1 Scénario de test

A L'aide de la validation de sous-échantillonnage aléatoire répété (en anglais : Repeated random subsampling validation), nous avons devisé les 50 sons de chaque classe (cris des humains, bris de glace et alarmes de voitures) en deux groupes, un groupe pour construire le codebook et l'autre groupe pour le test. Nous avons utilisé 30 sons pour construire le codebook et 20 sons pour effectuer le test. A cet effet, le codebook global, qui inclus les trois catégories, est composé de 90 sons tandis que le groupe global de test inclus 60 sons.

Dans le but de réduire la complexité des méthodes proposées pour la reconnaissance, nous devons trouver la durée minimale des sons de test qui conduit à un taux de reconnaissance élevé en utilisant le moins d'attributs possible. Pour cela, nous avons varié les durées de tous les sons de

notre corpus entre 0.1s et 0.6s avec un pas de 0.1s. Pour chaque durée imposée, nous avons varié le nombre d'attributs de 2 à 20 avec un pas de 1.

Voici un exemple d'un son de test (alarme de voiture) avant et après l'introduction de la durée du segment à traiter (les figure 4.9 et 4.10).

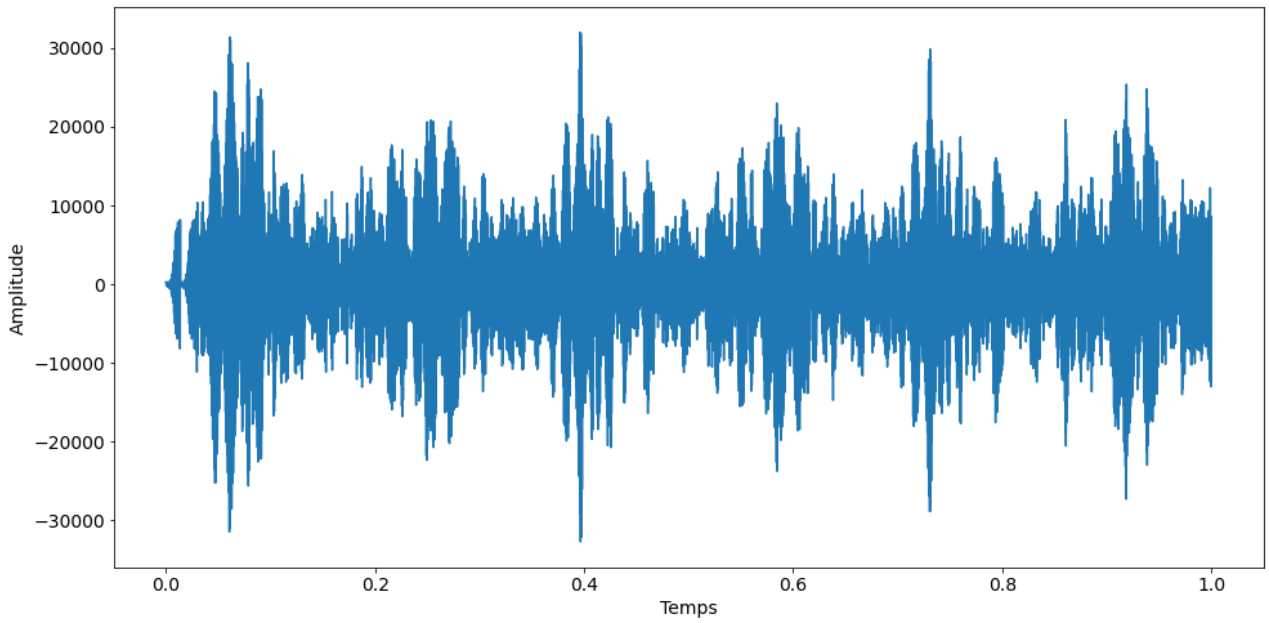

Figure 4.9 : Son de test original

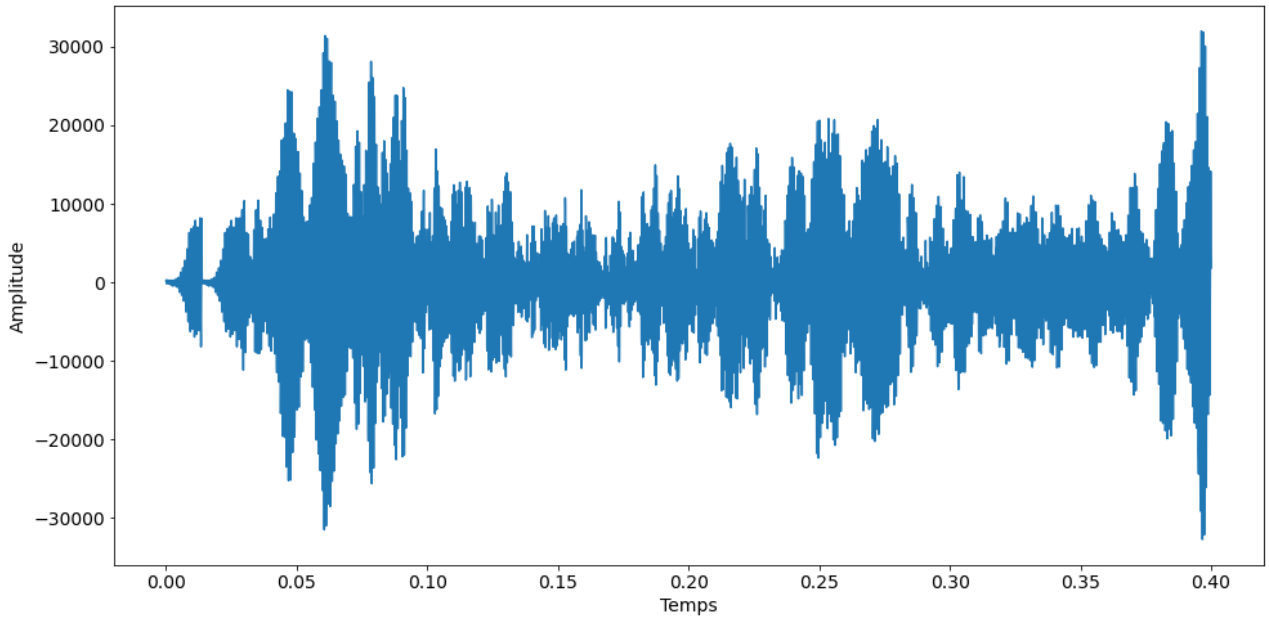

Figure 4.10 : Limitation de la durée d'un son de test à 0.4s

4.6.2 Extraction d'attributs

Pour une série temporelle, nous allons extraire les attributs MFCCs. Le calcul de ces coefficients est effectué en considérant les paramètres suivants :

- Le coefficient de préaccentuation est fixé à 0.98,
- La taille de la fenêtre d'analyse est de 20ms,
- Le recouvrement entre les fenêtres est de 10ms,
- Le nombre des bancs de filtre est de 40,

La Figure 4.11 montre un exemple de calcul de 13 MFCCs pour une alarme de voiture.

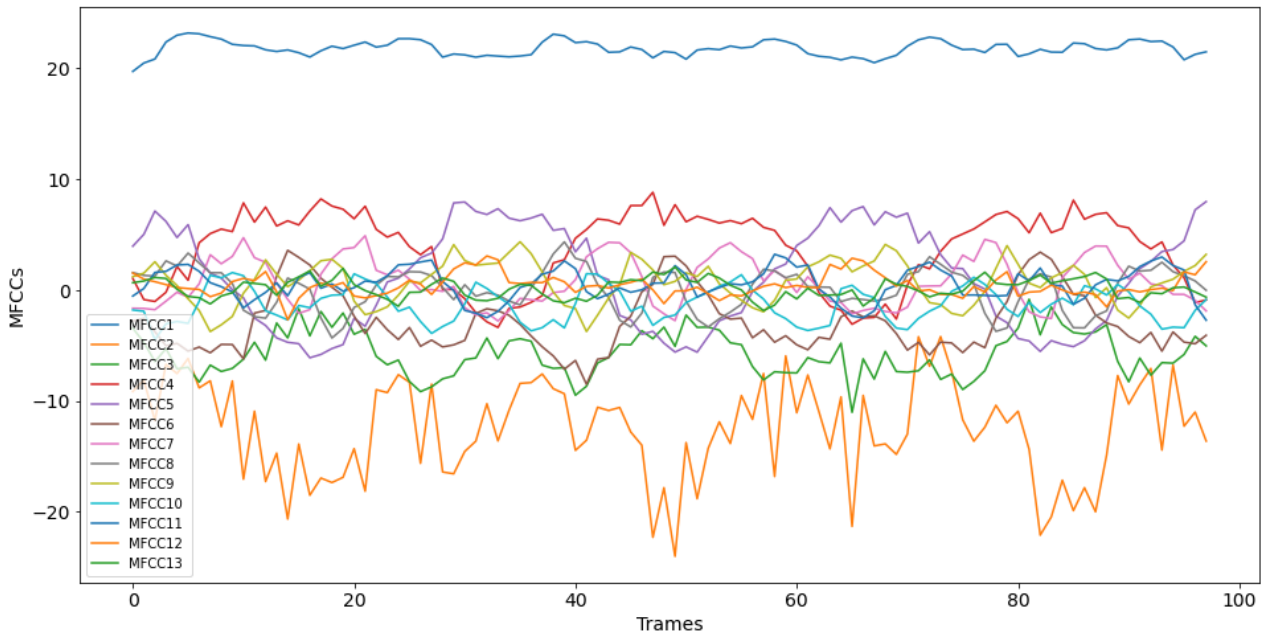

Figure 4.11 : Evolution des MFCCs d'un son en fonction des trames d'analyse.

Une deuxième étape consiste à calculer les dérivés de MFCCs. La figure 4.12 donne un exemple de calcul de 13 MFCCs en tenant compte des premier et deuxième coefficients dérivés. Le nombre total des coefficients dans ce cas est de 39 coefficients (13 MFC Cs + 13 Delta MFCCs +13 Delta Delta MFCCs).

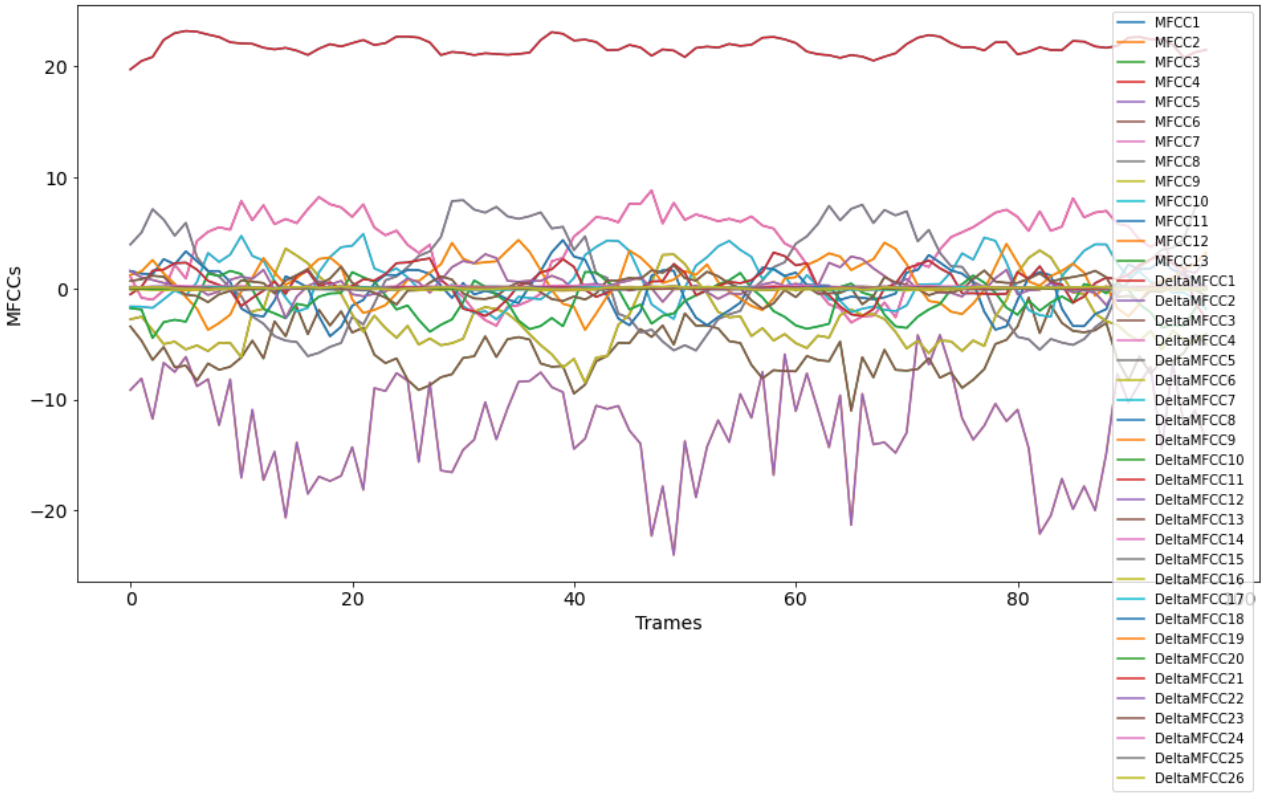

Figure 4.12 : Evolution des MFCCs d'un son en fonction des trames d'analyse (avec la première et la deuxième dérivées).

Dans cet exemple, nous calculons 5 attributs MFCCs du son utilisé précédemment. La matrice M montre les valeurs numériques d'attributs calculés. Le nombre de ligne correspond nombre de trames et le nombre de colonne correspond aux attributs(tableau4-3).

Tableau 4-3 : Matrice d'attributs MFCC pour un son de test.

21.56874681	- 15.03786289	-6.7932369	-4.40436111
21.73083671	- 14.53914447	-5.38977135	-3.71694356
21.96011466	- 11.54416719	-5.02593313	-4.27190068
21.93638171	- 13.19909927	-5.75858691	-5.19151753
21.57304082	- 8.49905172	-4.08931465	-5.99560544
21.09583254	-9.94838937	-4.72592997	-5.51529975
21.57528987	-7.77310276	-5.090482	-6.12996896
22.06795324	-7.32642037	-5.001873	-7.06609698

4.6.3 Schéma de reconnaissance à base de la DTW

Avant d'exposer les différents tests réalisés et les résultats obtenus, cette partie décris le schéma de classification à base de la DTW mentionné dans le chapitre 3. Assigner le son de test utilisé dans l'exemple précédent à la classe la plus probable est effectué comme suit :

Calcul de la distance DTW entre la matrice M et le triplet T(i) pour i=1 jusqu'à 30,

A titre d'exemple, les distances entre la matrice M et les trois matrices du triplet (1) sont : d1= 192.25139642, d2= 81.73766022 et d3= 144.28808745.

La décision préliminaire Dec1 pour ce cas est la classe 1 (Cris humains).

Constitution du vecteur des décisions finales {Dec1, Dec2, …. Dec30}

Cette étape calcul les décisions obtenues en utilisant les 30 triplets du Codebook (Figure 4.17).

(1) Alarme de voiture,

(2) Cris humains,

(3) Bris de glace,

Identification de la classe la plus probable

D'après la Figure, nous remarquons que la classe la plus fréquente qui correspond au mode de cette suite est la classe (1 alarme de voiture)

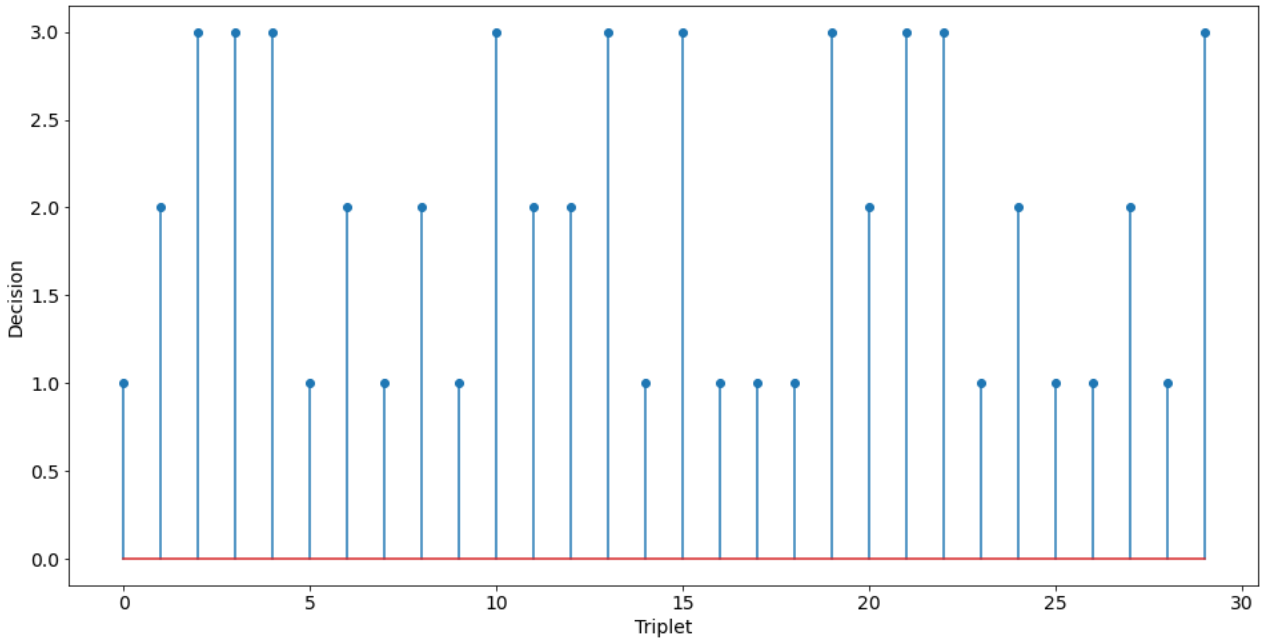

Figure 4.13 : Calcul des décisions finales

La méthode proposée est implémentée et testée sur les deux corpus clean et enviromentals. Les résultats présentés par la suite sont obtenus avec la validation croisée de sous-échantillonnage aléatoire répété 5 fois.

- Corpus de donnes clean

La Figure 4.14 présente les taux de classification (TBC) obtenus en variant le nombre des MFCCs et la durée des sons :

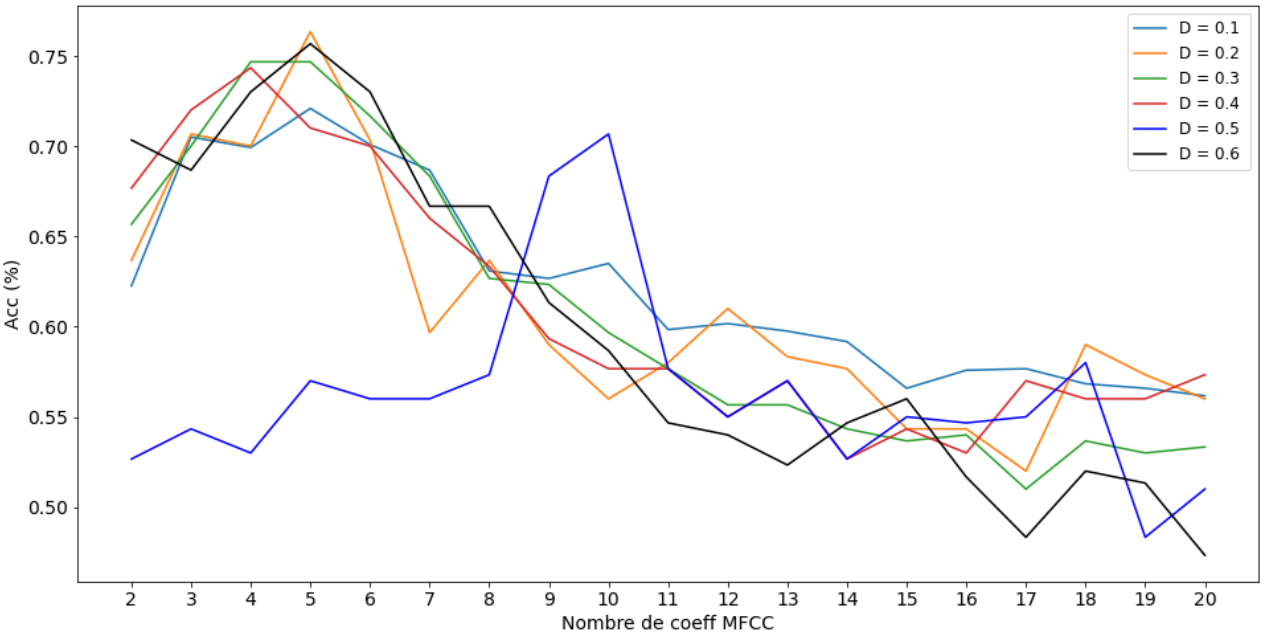

Figure 4.14 : Variation du nombre d'attributs MFCCs et de la durée des sons de test de corpus clean (sans dérivées)

D’après la Figure 4.14, nous remarquons que pour une durée de 0.2s, Le TBC a atteint une valeur maximale de **76%** en utilisant seulement 5 coefficients MFCCs. Le taux d’erreur correspondant est de : **24 %.**

- Corpus de donnée enviromentals (avec et sans dérivé)

La Figure 4.13 présente les TBC obtenus en variant le nombre des MFCCs et la durée des sons sans inclusion des dérivées.

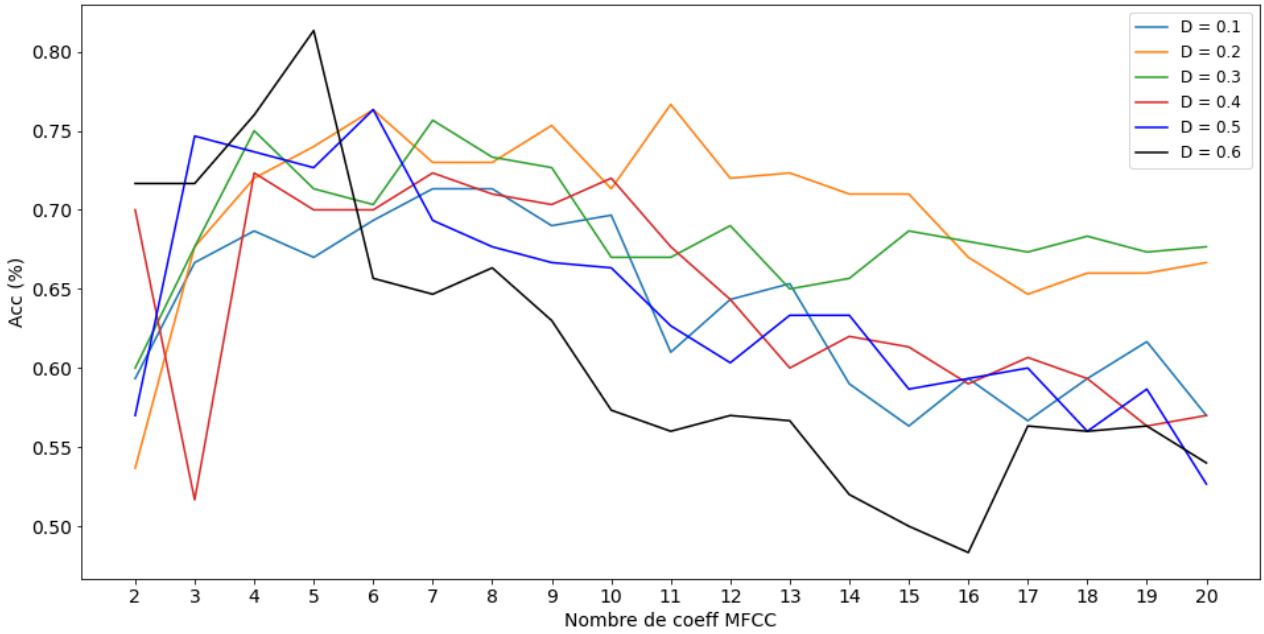

Figure 4.15 : Variation du nombre d’attributs MFCCs et de la durée des sons de test de corpus environnementale (sans dérivées).

D’après cette Figure, nous remarquons que pour une durée D=0.6s, Le TBC a atteint une valeur de 81% en utilisant seulement 5 coefficients MFCCs. Le taux d’erreur est de 19%.

En incluant les deux drivées des MFCCs, les résultats obtenus sont présentés sans la figure 4.15

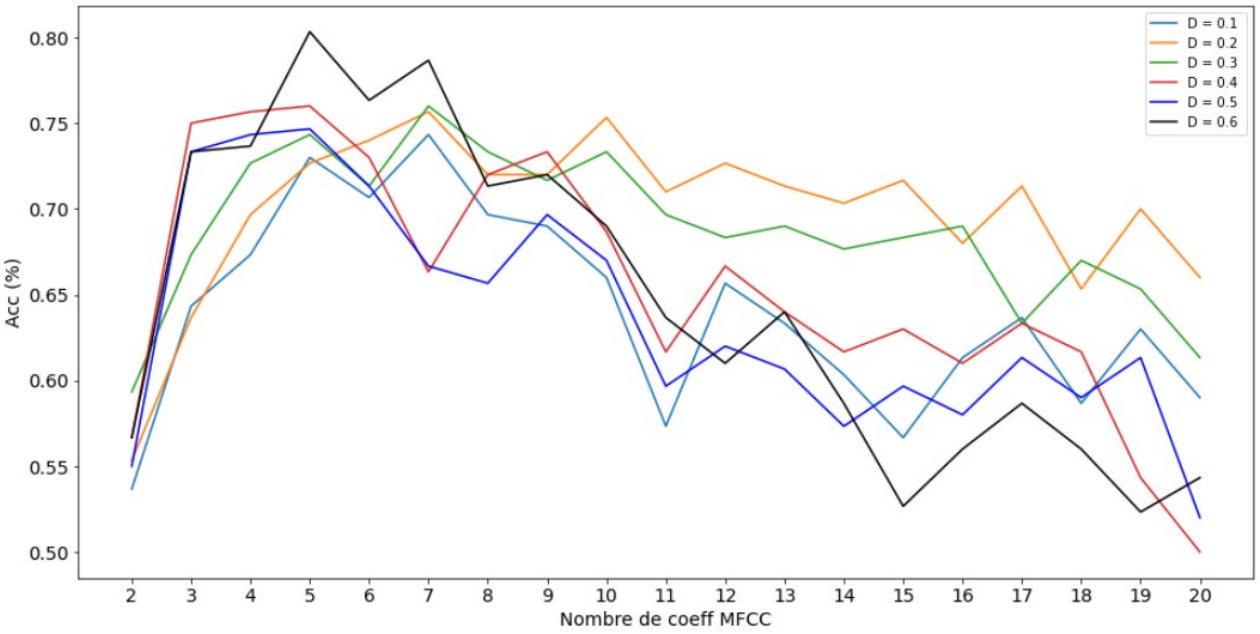

Figure 4.16 : Variation du nombre de MFCCs et de la durée des sons de test de corpus environnementale (avec inclusion des dérivées)

D'après cette Figure, nous remarquons que pour une durée D=0.6s, l'Acc a atteint une valeur de 81% en utilisant 5 coefficients MFCCs. Le taux d'erreur est de 20%.

4.6.4 Comparaison des résultats avec un modèle à base d'apprentissage

Nous avons comparé les performances de notre système à un modèle qui se base sur l'apprentissage de données. Notre choix se porte sur l'algorithme BI-LSTM vu qu'il est plus adapté pour les séquences et efficace pour le Small size data.

Les réseaux Bidirectional Long Short-Term Memory (BLSTM) sont un type spécial de LSTM-RNN, capable d'apprendre les dépendances à long terme. Se souvenir des informations pendant de longues périodes est pratiquement leur comportement par défaut, pas quelque chose qu'ils ont du mal à apprendre. Ils consistent à dédoubler une couche LSTM autrement dit c'est parcourir un signal à une dimension selon ses deux directions, l'une étant apprise pour parcourir le signal de gauche à droite, et l'autre de droite à gauche. Nous avons utilisé ce type de modèle d'apprentissage (BLSTM) pour la reconnaissance des trois classes de sons (cris humains, bris de glace, et alarmes de voitures). Vu que Les BiLSTM donnent de bien meilleures performances.

Pour se faire, nous avons aléatoirement choisis les paramètres principaux pour avoir des bons résultats. Les paramètres du modèle sont les suivantes :

- La couche d'entrée de 32 neurones et avec une fonction d'activation « relu ».

- La deuxième couche bidirectionnelle, se compose de 16 neurones et de la fonction d'activation « relu », Dropout a 0.3 comme valeur.
- La dernière couche de sortie, comprend 03 neurone et une fonction d'activation « softmax ».

Dans notre étude, nous avons utilisé le corpus environmental avec une durée de segmentation à 0.2s ainsi que le nombre de coefficient MFCCs à 4 pour comparer les deux modèles. Le tableau 4-4 présente les résultats de comparaisons :

Tableau 4-4 : Comparaison des résultats entre la DTW et le BLSTM.

Modèle / **TBC**	**DTW (%)**	**BLSTM (%)**
Taux de bonne classification	82	98

Nous avons constaté que le BLSTM donne de meilleures performances par rapport le DTW, un taux de bonne classification très satisfaisant avec peu de donnée.

4.6.5 Reconnaissance online des évènements audio

Pour valider l'efficacité du schéma de classification à base de la DTW proposé, Nous avons l'implémenté en temps réel. En se basant sur le critère de la data centric, nous avons choisis des triplets de code book représentatifs qui conduisent à un taux de reconnaissance élevé en temps réel.

D'après les résultats obtenus offline avec la validation de sous-échantillonnage aléatoire répété 5 fois sur le corpus environnementale, et parmi les 5 rééchantillonnage générées on a pu choisir les 30 triplets représentatifs du codebook. Pour une durée minimale de 0.2s et 4 attributs de MFCCs, on a pu achever un taux de reconnaissance de 82% online. Dans la matrice de confusion est donnée comme suit :

		Classes réelles		
		A	B	C
Classes prédites	A	16	2	2
	B	0	20	0
	C	3	4	13

A, B et C sont les classes à reconnaitre :

(A) Alarme de voiture,

(B) Bris de glace,

(C) Cris humains,

Pour ce cas, les taux de précision et de rappel pour les trois classes sont résumés dans le tableau 4.5.

Tableau 4-5 : Précision et rappel du modèle de reconnaissance à base la DTW.

Métriques / **Classes**	**Rappel (%)**	**Précision (%)**
Alarme de voiture	80	84
Bris de glace	100	77
Cris humains	65	87

A partir des résultats du tableau 4.5, nous avons obtenu des taux élevés du rappel et de la précision pour les trois classes. Nous pouvons conclure que les trois classes sont parfaitement gérées par ce schéma de classification proposé.

Ci-dessous, Nous exposons l'interface globale de notre système développé. Il s'agit d'une plateforme de reconnaissance d'évènements impulsifs en temps réel. Une fois un évènement impulsif est détecté, notre système affiche la classe d'événement produit dans le champ de résultat. Plus particulièrement, il identifier la le type exact du son impulsif et le catégoriser selon les trois classes d'intérêt.

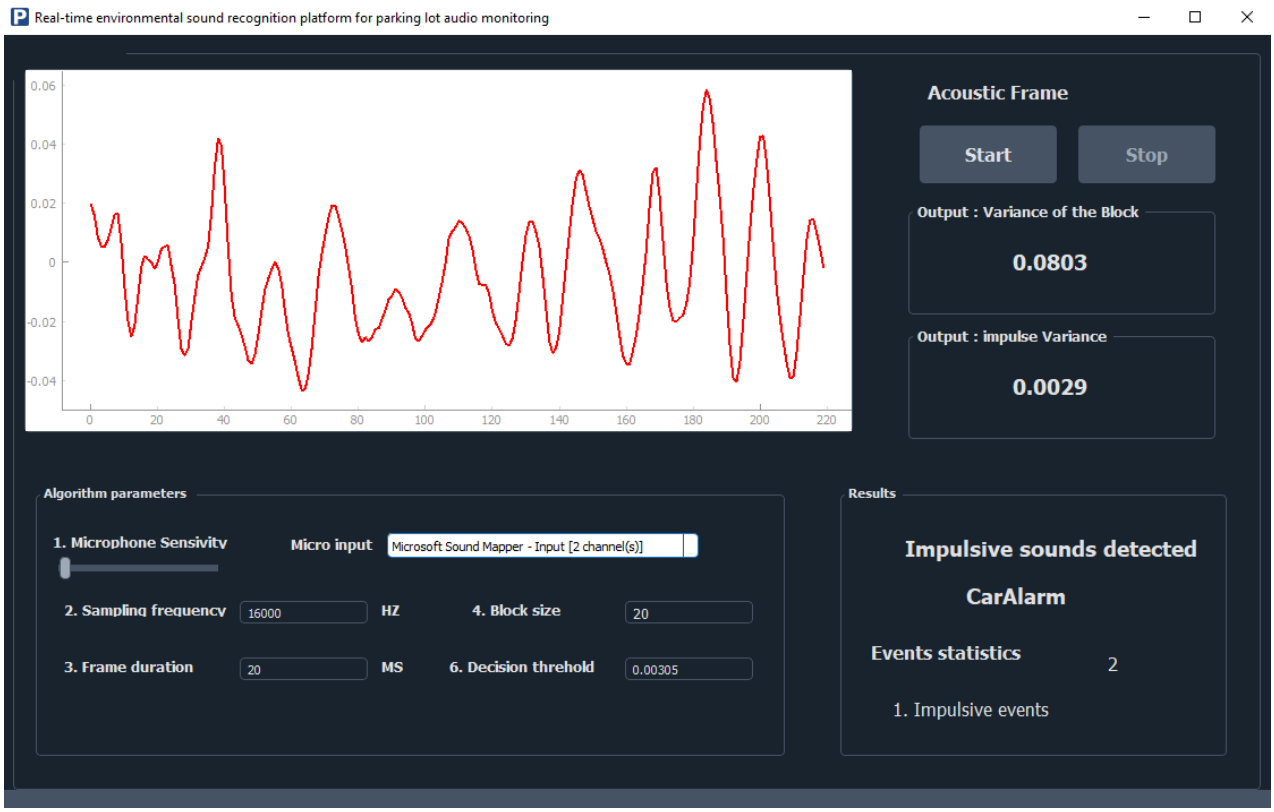

Figure 4.17 : Interface de la plateforme de la reconnaissance des évènements acoustiques en temps réel.

4.7 Conclusion

Dans ce chapitre, nous avons présenté les résultats expérimentaux de la détection et la reconnaissance des événements sonores en temps réel. Au début, nous avons fait une présentation sur les logiciels de développements et les langages de programmation utilisés. Par la suite, nous avons présenté le corpus sonore utilisé dans nos expérimentations.

Les détails concernant : le lieu de test, le matériel nécessaire sont fournis dans une section à part. Nous avons présenté les différentes étapes nécessaires pour l'implémentation des méthodes proposée.

Notre approche prouve son intérêt en donnant des bonnes solutions dans un temps assez raisonnable.

Chapitre 5 : Conclusions et travaux futures

Conclusions

L'objectif de cette étude est de mettre en œuvre une plateforme de reconnaissance des événements sonores en temps réel en vue de surveiller les parcs de stationnements. L'IA à base du Data Centric est adaptée au développement de ce système. L'approche que nous proposons est composée de deux étages principaux : la détection des évènements sonores et la reconnaissance des sons.

Nous avons présenté dans le deuxième chapitre du mémoire les généralités sur les parcs de stationnement et les systèmes de surveillance. Nous nous sommes focalisés sur l'apport de la modalité audio à ce genre de système ainsi que la description de la détection, la reconnaissance et la localisation des évènements impulsifs.

Dans le troisième chapitre, nous avons présenté l'approche d'IA centrée sur les données qu'on a adopté pour concevoir notre système de surveillance. Par la suite, nous avons présenté une description détaillée des solutions que nous avons optée pour la détection et la reconnaissance des évènements acoustiques ainsi que le fonctionnement temps réel de notre méthode. Ensuite, nous avons cité les métriques nécessaires pour évaluer les performances de ces méthodes.

Notre méthode de détection est basée sur l'analyse de la variance des énergies à courts termes d'une forme d'onde acoustique. L'optimisation des paramètres algorithmiques est faite en se basant sur une séquence audio de durée fixe. Toutefois, la méthode de reconnaissance se base sur un schéma de classification qui est :la DTW. Les MFCCs sont utilisé pour extraire les caractéristiques des attributs. Ce sont des coefficients basés sur le processus de la perception auditive humaine.

La reconnaissance à base de la DTW est fondée sur une multitude de données de références (codebook) pour le calcul des distances temporelles entre les séquences, ce qui nous a permis d'identifier la classe la plus fréquence dans notre schéma de classification. En revanche, le réseau de neurones artificiels BI-LSTM est un modèle d'apprentissage utilisé pour la comparaison des résultats de classification des événements acoustiques à base de la DTW. Nous nous sommes limités dans notre projet à trois catégories de sons : (i) cris humains, (ii) alarmes de voitures et (iii)

bris de glace. La méthode proposée a été présentée en détails dans le troisième chapitre. Notre démarche tient en compte des exigences suivantes :

- Formalisme mathématique non complexe pour la méthode de détection,
- Facilement adaptable aux conditions de l'environnement,
- La reconnaissance des sons de l'environnement doit utiliser des méthodes à complexité réduite pour faciliter son implémentation en temps réel,
- La durée du son nécessaire à la tâche de reconnaissance doit être courte,
- Le choix de références du codebook doit être représentatif afin d'obtenir un taux de reconnaissance le plus élevé possible.

Dans le quatrième chapitre, nous avons présenté les résultats expérimentaux. Nous avons utilisé : le taux de vrais positifs (TVP), le taux de faux positifs (TFP), la mesure F1-score, le taux de bonne classification, la précision, et le rappel pour évaluer les performances des méthodes proposées.

D'après les résultats obtenus, nous avons trouvé que la méthode de détection proposée a atteint un taux de vrais positifs de **91%**. En ce qui concerne la méthode de reconnaissance à base de la DTW ,on aboutit un taux de reconnaissance de **82%** en utilisant seulement 5 coefficients MFCCs pour une durée de 0.2 en choisissant les références du codebook représentatif .En comparant ce schéma de classification avec un modèle à base d'apprentissage BLSM, le taux de reconnaissance a atteint **98 % ,**il a dépassé celui obtenu avec DTW .on a conclu que Les résultats de reconnaissances pourraient éventuellement être améliorés en utilisant un modèle à base d'apprentissage en utilisant peu de donnée de haute qualité .

Les méthodes proposées pour la conception de notre plateforme possèdent l'avantage d'être :

(i) Rapide, efficace et non dépendante d'une phase d'apprentissage,

(ii) Implémentable en temps réel,

Travaux futurs

Les travaux futurs consistent à :

- Augmenter le nombre de classes et mesurer les performances de la méthode,
- Intégrer ce module de localisation dans la plateforme de reconnaissance des sons impulsifs en temps réel,

- Mesurer les performances de reconnaissance des sons de l'environnement en temps réel en utilisant d'autres technique d'extraction d'attributs basée sur la prédiction linéaire (LPCs).
- Implémenter le modèle de Bi-LSTM pour la classification des sons en temps réel

Bibliographies

[1] A. Dufaux, “Detection and recognition of impulsive sound signals,” Institute of Microtechnology, Neuchatel University, Neuchatel, 2001.

[2] DJEBRI Lamine and LAMRAOUI Mourad, “Prévision de stationnement dans un environnement IoT,” UNIVERSITE AKLI MOHAND OULHADJ-BOUIRA, BOUIRA, 2019.

[3] “PARKING FACILITIES.” https://www.parking-net.com/about-parking/parking-facilities (accessed May 05, 2022).

[4] Benoit Charette, “Installation du premier parcomètre,” Dec. 10, 2019. https://www.journaldechambly.com/16-juillet-1935-installation-du-premier-parcometre/ (accessed May 05, 2022).

[5] Shubhankar Gautam, “Traditional vs Automated Parking System,” Jul. 09, 2019. https://blog.getmyparking.com/2019/07/09/traditional-vs-automated-parking-system/ (accessed May 03, 2022).

[6] S.-F. Lin, Y.-Y. Chen, and S.-C. Liu, “A Vision-Based Parking Lot Management System,” *2006 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 2897–2902, 2006.

[7] URBANCLAP EDITORIAL, “All You Need To Know About CCTV Secured Garages ,” Sep. 07, 2017. https://www.urbancompany.com/blog/homecare/cctv/all-you-need-to-know-about-cctv-secured-garages/ (accessed Jun. 21, 2022).

[8] Yassine Benabbas, “Analyse du Comportement Humain à Partir de la Vidéo étudiant l’Orientation du Mouvement,” UNIVERSIT E DES SCIENCES ET TECHNOLOGIES DE LILLE, LILLE, 2012. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00839699

[9] Sarah AHMED HAMADA, “Détection des sons impulsifs en vue de la mise en œuvre d’un système de surveillance audio,” Université SAAD DAHLAB de BLIDA, Blida, 2017.

[10] Sébastien LECOMTE, “Classification partiellement supervisée par SVM. Application à la détection d’événements en surveillance audio,” Université de technologie Troyes, France, 2013.

[11] Mayssaa Al Najjar, Milad Ghantous, and Magdy Bayoumi, *Video Surveillance for Sensor Platforms*, Visual Sensor Nodes., vol. 114. Springer, New York, NY, 2014. doi: https://doi.org/10.1007/978-1-4614-1857-3.

[12] Yassine Benabbas, “Analyse du Comportement Humain à Partir de la Vidéo en étudiant l’Orientation du Mouvement,” Universit e des Sciences et Technologies de Lille, Lille, 2012. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00839699

[13] M. Najjar and M. Magdyybayoumi, “Lecture Notes in Electrical Engineering 114 Video Surveillance for Sensor Platforms Algorithms and Architectures.” [Online]. Available: http://www.springer.com/series/7818

[14] LYNDSAY WINKLEY, “San Diego police to continue using gunshot detection system, despite some criticism,” Oct. 07, 2017. https://www.latimes.com/local/lanow/la-me-ln-sd-gunshot-detection-20171007-story.html (accessed Jun. 22, 2022).

[15] F. Ykhlef, S. A. Hamada, F. Ykhlef, A. Derbal, and D. Bouchaffra, “Real-Time Detection of Impulsive Sounds for Audio Surveillance Systems,” Saida,Algeria, 2019.

[16] M. Aramaki, “Analyse-synthèse de sons impulsifs : approches physique et perceptive,” UNIVERSITE DE LA MEDITERRANEE, MARSEILLE, 2003. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00005491

[17] G. Ciaburro, “Sound Event Detection in Underground Parking Garage Using Convolutional Neural Network,” *Big Data and Cognitive Computing*, vol. 4, no. 3, 2020, doi: 10.3390/bdcc4030020.

[18] L. L. Ng and H. S. Chua, “Vision-based activities recognition by trajectory analysis for parking lot surveillance,” in *2012 IEEE International Conference on Circuits and Systems (ICCAS)*, 2012, pp. 137–142. doi: 10.1109/ICCircuitsAndSystems.2012.6408305.

[19] Y. and C. H. Na Keewook and Kim, “Acoustic Sensor Network-Based Parking Lot Surveillance System,” in *Wireless Sensor Networks*, 2009, pp. 247–262.

[20] P. and A. G. and D. A. Leo M. and Spagnolo, “Shape Based People Detection for Visual Surveillance Systems,” in *Audio- and Video-Based Biometric Person Authentication*, 2003, pp. 285–293.

[21] Srishti Mukherjee, “Why Andrew Ng favours data-centric systems over model-centric systems,” Feb. 15, 2022. https://analyticsindiamag.com/why-andrew-ng-favours-data-centric-systems-over-model-centric-systems/ (accessed Jun. 22, 2022).

[22] LANDING AI, “Data-Centric AI AI has Evolved. Accelerate Machine Vision Innovation with Data-Centric AI.,” 2022. https://landing.ai/data-centric-ai/ (accessed Jun. 22, 2022).

[23] ELIZA STRICKLAND, “Andrew Ng : IA sans limite,” Feb. 09, 2022. https://spectrum.ieee.org/andrew-ng-data-centric-ai (accessed Jun. 22, 2022).

[24] S. Khara, S. Singh, and D. Vir, “A Comparative Study of the Techniques for Feature Extraction and Classification in Stuttering,” in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 887–893. doi: 10.1109/ICICCT.2018.8473099.

[25] N. Dave, “Feature Extraction Methods LPC , PLP and MFCC In Speech Recognition,” 2013.

[26] S. A, A. Thomas, and D. Mathew, “Study of MFCC and IHC Feature Extraction Methods With Probabilistic Acoustic Models for Speaker Biometric Applications,” *Procedia Computer Science*, vol. 143, pp. 267–276, Jun. 2018, doi: 10.1016/j.procs.2018.10.395.

[27] S. Soheily-Khah, “Generalized k-means based clustering for temporal data under time warp,” Theses, Universite Grenoble Alpes, Grenoble , 2016. Accessed: Jun. 22, 2022. [Online]. Available: https://hal.archives-ouvertes.fr/tel-01394280

[28] Z. Djazia Souheir and S. Nacira, “Systéme automatique de reconnaissance vocale d’un locuteur basé sur la programmation dynamique (DWT),” Theses, université saad dahleb blida, Algérie,blida, 2018.

[29] F. Itakura, “Minimum prediction residual principle applied to speech recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975, doi: 10.1109/TASSP.1975.1162641.

[30] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978, doi: 10.1109/TASSP.1978.1163055.

[31] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978, doi: 10.1109/TASSP.1978.1163055.

[32] Doreen Gallagher, “Time Series I,” 2016. https://slideplayer.com/slide/6100297/ (accessed Jun. 13, 2022).

[33] “MATLAB pour l’intelligence artificiell.” https://fr.mathworks.com/discovery/artificial-intelligence.html (accessed Jun. 22, 2022).

[34] “Towards Data Science.” https://towardsdatascience.com/ (accessed May 27, 2022).

[35] “lebigdata.fr,” Jun. 05, 2022. https://www.lebigdata.fr/python-langage-definition (accessed Jun. 14, 2022).

[36] “Python,” Apr. 04, 2021. https://www.python.org/downloads/release/python-394/ (accessed Jun. 14, 2022).

[37] Mohammad Waseem, “Python Anaconda Tutorial : Everything You Need To Know,” Jul. 15, 2021. https://www.edureka.co/blog/python-anaconda-tutorial/ (accessed Jun. 11, 2022).

[38] “Anaconda.” https://www.dominodatalab.com/data-science-dictionary/anaconda (accessed Jun. 11, 2022).

[39] Henri Michel, “Google Colab : Le guide Ultime.” https://ledatascientist.com/google-colab-le-guide-ultime/ (accessed Jun. 11, 2022).

[40] “Fonctionnalités de PyCharm.” https://www.jetbrains.com/fr-fr/pycharm/features/ (accessed Jun. 11, 2022).

[41] “PyCharm.” https://www.jetbrains.com/help/pycharm/quick-start-guide.html (accessed Jun. 11, 2022).

[42] “Qt Designer Manual,” 2022. https://doc.qt.io/qt-5/qtdesigner-manual.html (accessed Jun. 11, 2022).

[43] “GoldWave Inc.,” 2022. https://www.goldwave.com/about.php (accessed Jun. 11, 2022).

[44] “Goldwave Sound File Viewing & Editing Software.” https://www.gl.com/goldwave.html (accessed Jun. 11, 2022).

[45] “Goldwave,” Jun. 2022. https://www.goldwave.com/. (accessed Jun. 20, 2022).

[46] Leo Beranek, “Acoustics,” *Acoustical Society of America*, 1993.

[47] “Sound Dogs,” May 2004. https://www.sounddogs.com/ (accessed Jun. 14, 2022).