UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

AMÉLIORATION DE LA GESTION DU BRUIT ENVIRONNEMENTAL PAR L'IDENTIFICATION DES SOURCES SONORES PRINCIPALES À L'AIDE D'UN RÉSEAU NEURONAL CONVOLUTIF SIAMOIS AVEC FONCTION DE PERTE PAR TRIPLET

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE DE LA
MAÎTRISE EN GÉNIE MÉCANIQUE

PAR
JEAN-PIERRE CÔTÉ

MARS 2024

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MAÎTRISE EN GÉNIE MÉCANIQUE (M.SC.)

Direction de recherche :

Marc-André Gaudreau	
Prénom et nom	directeur de recherche
Sousso Kelouwani	
Prénom et nom	codirecteur de recherche

Jury d'évaluation

Rachel Bouserhal	Membre du jury
Prénom et nom	Fonction du membre de jury
Mohamed Habibi	Membre du jury
Prénom et nom	Fonction du membre de jury
Sousso Kelouwani	Membre du jury
Prénom et nom	Fonction du membre de jury
Prénom et nom	Fonction du membre de jury
Prénom et nom	Fonction du membre de jury

REMERCIEMENTS

Marc-André, Josée, Henri-Victor, Esther-Bérénice, Laure-Charlotte : Merci. Pour de vrai.

RÉSUMÉ

Ce mémoire présente une preuve de concept exploratoire basée sur un modèle de réseau neuronal siamois permettant d'identifier les principales sources sonores dans des environnements industriels complexes. L'objectif est d'évaluer la faisabilité d'une méthode permettant de corréler des captations proches des sources industrielles avec une captation distante effectuée en zone résidentielle, dans le but d'améliorer la gestion des nuisances sonores. Contrairement aux approches classiques de classification, ce modèle utilise une analyse comparative fondée sur la similitude entre les signaux sonores. Cette approche vise à généraliser à des types de sons non entraînés et à s'adapter à la variabilité acoustique propre aux environnements réels. Les tests ont été réalisés en contexte simulé, avec des mélanges synthétiques contrôlés. Les résultats montrent que le modèle atteint une précision de 80 % avec un rapport signal/bruit (S/B) de 13 dB et maintient une performance de près de 60 % même lorsque le S/B est de 0 dB, démontrant sa capacité à identifier le signal dominant même dans des environnements fortement bruités. Toutefois, cette performance s'accompagne d'une diminution de la spécificité à fort S/B, ce qui traduit une tendance du modèle à attribuer à tort le rôle de source dominante à certaines sources secondaires. Cette caractéristique pourrait néanmoins être exploitée pour établir une hiérarchisation des contributions sonores plutôt qu'une identification binaire, ouvrant ainsi une nouvelle piste pour la gestion fine du bruit environnemental. Ce travail constitue une première étape vers une approche automatisée de la corrélation entre sources mobiles et mesures de bruit à distance. Des recherches futures seront nécessaires pour tester cette méthode en conditions réelles, explorer d'autres architectures (par exemple à base de Transformers), et affiner la fonction de perte afin de renforcer la robustesse et la précision du modèle.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ÉQUATIONS

CHAPITRE 1 : INTRODUCTION

Dans ce chapitre, nous présentons une problématique de nuisance environnementale par le bruit pour laquelle le port de Trois-Rivières a demandé à être accompagné dans sa recherche de solution, et nous précisons les recherches que nous avons effectuées afin de tenter d'y répondre.

1.1 Contexte et justification

1.1.1 Partenaires

Notre partenaire principal, l'Administration portuaire de Trois-Rivières (APTR), est une agence fédérale autonome responsable de la gestion des quais du parc portuaire de la ville de Trois-Rivières (Québec, Canada). Notre autre partenaire, Logistec, Services maritimes, est l'exploitant du terminal. Il offre des services de manutention des marchandises. Parmi les principales marchandises qui transitent par le port de Trois-Rivières, on retrouve l'acier, les composantes d'éoliennes et l'aluminium. Ce port, situé en zone urbaine, « accueille annuellement 55 000 camions, 11 000 wagons et plus de 250 navires marchands et de croisières provenant d'une centaine de ports situés dans plus de 40 pays à travers le monde. Il manutentionne un trafic de plus de 3,5 M de tonnes métriques. » (Aptr, 2021)

1.1.2 Nuisance par le bruit

Les quais du port de Trois-Rivières sont situés sur une bande relativement étroite entre des quartiers résidentiels et le fleuve Saint-Laurent (Figure 1.1). Cette proximité des zones habitées crée une problématique de nuisance sonore. En effet, des plaintes de citoyens sont recensées quant au bruit généré, notamment par la manipulation des éléments métalliques sur les quais à proximité. Un rapport d'impact environnemental sur les nuisances engendrées par le port de Trois-Rivières, commandé par ce dernier et livré en octobre 2020, a confirmé que les seuils légaux sont régulièrement dépassés. Ce rapport suggérait, comme mesure de mitigation, l'installation d'une barrière sonore. Cependant,

dans un effort de modernisation des liens avec la communauté, on cherche à donner accès au fleuve aux citoyens ; il est donc exclu d'installer des barrières sonores qui bloqueraient la vue du fleuve.

Figure 1.1 : Les parties des terrains de manutention du port de Trois-Rivières qui sont rapprochées des quartiers résidentiels [source : Logistec].

Figure 1.2 : Emplacement des sondes automatiques au port de Trois-Rivières (rouge : sonomètres, bleu : vibration, vert : qualité de l'air) [source : SETI].

1.1.3 Recherche de solution

En vue de gérer les dépassements de niveau sonore, l'APTR s'est muni d'un réseau de stations environnementales, dont 3 stations de monitorage du bruit (Figure 1.2). Mises en place par Seti Media, ces stations fournissent de l'information continue sur les niveaux sonores atteints.

À l'aide de ces mesures, l'APTR a amorcé une analyse de ses activités bruyantes et est à la recherche de solutions. Elle fait 2 constats :

- Il y a un lien direct entre le niveau d'activité et la nuisance sonore ;
- Les habiletés des opérateurs de machinerie varient, et elles ont une influence probable sur le bruit généré.

1.1.4 Notre proposition

Nos partenaires souhaitent que les solutions de réduction de la nuisance par le bruit qui seront adoptées aient le moins d'incidence possible sur leur niveau d'activité. À cette fin, nous souhaitons les munir d'un outil leur permettant d'agir à la source, au niveau de la manipulation des pièces lourdes, en informant en temps réel les opérateurs de machinerie lorsqu'ils provoquent une nuisance sonore en zone résidentielle. Cette information permettra d'ajuster plusieurs aspects de l'organisation du travail afin de réduire la nuisance sonore tout en minimisant l'impact sur le niveau d'activité. Ainsi, nous proposons qu'un système automatisé et capable de réagir en temps réel soit mis en place.

1.2 Problématique

Ce système devra :

1. Identifier qu'un dépassement d'un seuil prédéterminé est survenu en zone résidentielle ;

2. Obtenir les données pour chacune des sources potentielles en zone de manutention du quai ;

3. Identifier la ou les sources contributives principales ;

4. Transmettre l'information aux sources concernées.

Pour ce mémoire, nous avons choisi de nous concentrer spécifiquement sur le troisième point, à savoir la méthode d'identification des sources contributives principales. Cependant, nous avons veillé à intégrer cette méthode dans une approche globale, en tenant compte des interactions avec les autres étapes du système, ceci afin de favoriser la cohérence et la fonctionnalité du système final.

1.2.1 Pistes stratégiques

Une revue exploratoire de la littérature en lien avec notre sujet a identifié les pistes générales suivantes :

- Piste 1 : Les problématiques de pollution sonore générée par les ports maritimes situés à proximité des zones urbaines, et les stratégies de mitigation applicables ;
- Piste 2 : Les techniques d'identification des sources sonores en industrie, particulièrement en contexte de nuisance environnementale ;
- Piste 3 : L'apprentissage automatique par les réseaux de neurones profonds.

1.2.1.1 Piste 1 : La pollution sonore par les ports en zone urbaine

L'identification des sources sonores distinctes dans un environnement acoustique complexe, comme un port maritime, constitue une tâche particulièrement ardue. En effet, le bruit émis par les ports est complexifié par la présence, dans le même espace, d'un grand nombre de types de sources sonores, comme les moteurs des navires et des différents équipements de déchargement et manutention, la manipulation des marchandises elles-mêmes ou encore les sources sonores provenant de l'extérieur des installations (Čurović et al., 2021; Schenone et al., 2016). À ce propos, Curcuruto et al. (2000) écrivent que « Ces infrastructures sont très complexes et se caractérisent par de

nombreuses variables acoustiquement distinctes, correspondant aux différentes activités présentes dans l'infrastructure au cours de la journée de travail »[1].

Ainsi, du fait que les sources bruyantes, dans les ports, revêtent un caractère discontinu et impromptu, il est difficile de les distinguer, sur des mesures à long terme, du bruit de fond et/ou des sources situées dans l'environnement immédiat des sondes de monitorage des niveaux sonores en zone résidentielle (Murovec et al., 2018).

Dans le but d'aider les ports du monde à gérer leur nuisance, les autorités du Port d'Amsterdam ont publié le *Good Practice Guide on Port Area Noise Mapping and Management* (Van Breemen et al., 2008). Cet ouvrage propose, entre autres, une méthode permettant de produire une cartographie acoustique des zones portuaires. Elle fait appel à une procédure tenant compte du modèle physique de la zone, de la modélisation des sources sonores présentes ainsi que du traitement logiciel des paramètres à calculer (par exemple, les données météorologiques et les positions des sondes ayant enregistré les niveaux sonores lors d'opérations en référence). Des techniques de modélisation du même type sont par ailleurs utilisées en acoustique environnementale générale (Kragh, 2000).

La carte des niveaux sonores ainsi produite permet l'évaluation et la prédiction de la nuisance sonore. Cependant, la technique utilisée est strictement de l'ordre du modèle ; elle constitue une extrapolation à partir de mesures types et d'enregistrement des conditions pour une période en référence. Elle ne comprend aucun capteur en continu et ne peut donc faire état des actions particulières se déroulant sur le quai. En outre, elle ne permet pas de distinguer les sources ponctuelles à l'échelle d'un opérateur et son équipement.

[1] Traduction depuis l'anglais par l'auteur.

1.2.1.2 Piste 2 : L'identification des sources de bruit en industrie

La protection de la santé des travailleurs motive la recherche en identification des équipements bruyants utilisés en industrie. Diverses techniques ont été développées. Nous pouvons les regrouper en 4 catégories : l'analyse de corrélation entre 2 points de captation, la focalisation par imagerie acoustique, la cartographie acoustique et l'analyse des sources en mouvement.

1.2.1.2.1 Analyse de corrélation de phase entre 2 capteurs

L'analyse de signaux par transformée de Fourier rapide (FFT) permet la mesure de corrélation entre 2 ondes sonores, en temps réel, à l'aide d'ordinateurs courants. C'est cette technique que Wang (1978) utilise afin d'identifier les parties d'un moteur les plus significativement bruyantes : en établissant la corrélation entre chacun des capteurs placés directement en différents points du moteur et un capteur situé à quelques mètres, il identifie les parties du moteur les plus contributives.

A priori, cet outil semble approprié et directement applicable à notre problématique puisqu'il devrait permettre d'obtenir instantanément le niveau de corrélation entre la captation rapprochée des opérations/équipements et la captation effectuée à la station de monitorage de la nuisance. En plaçant des capteurs sur les équipements mobiles et en relevant lequel ou lesquels présente/nt un niveau de corrélation élevée avec les capteurs de mesure de nuisance, nous obtiendrions l'identification de la source coupable. Cependant, nous n'avons relevé aucun travail de recherche qui ait étudié l'application de cette technique en acoustique environnementale.

En effet, bien que des auteurs aient démontré des techniques fonctionnelles d'identification des sources sonores à l'aide de 2 capteurs plus ou moins distants, notre revue de la littérature a démontré que l'analyse de corrélation de phase ne donne pas de résultats probants quand le niveau de corrélation entre ces capteurs est très faible. En effet, si aucun des signaux sources n'est suffisamment dominant dans le champ sonore aérien, le système ne peut pas établir le lien mathématique puisque chaque source présente

contribue à abaisser le niveau de corrélation de toutes les autres. Dans notre cas, il n'y a, du point de vue de l'analyse de signal, aucune corrélation mesurable entre les 2 points de captation, car les distances de propagation et le nombre de sources contributives sont grands, ce qui provoque une dégradation substantielle du signal original, au point qu'il n'est plus détecté au sein du champ sonore distant. De fait, dans pareil cas, il s'agit de champ sonore diffus (Kragh, 2000), excluant l'analyse par corrélation de phase entre 2 captations éloignées l'une de l'autre. Il s'agit ici d'une limite de cette technique qui fait largement consensus auprès des acousticiens.

1.2.1.2.2 Focalisation par imagerie acoustique

Il demeure possible de mesurer la corrélation entre des signaux dont les points de capture sont rapprochés. Dans ce cas, on mesure des corrélations fortes, ce qui permet de dégager des informations d'un grand niveau de précision, à tout le moins pour les fréquences à longueur d'onde assez courte et pour des sources assez rapprochées d'un dispositif à plusieurs capteurs. C'est ainsi que des algorithmes utilisant des antennes de microphones ont été développés afin d'identifier, principalement, la direction de provenance des sources sonores. Les informations ainsi colligées sont ensuite présentées sous forme d'image ou les zones d'intensité sonores sont représentées par des taches ou des codes de couleurs. Certaines techniques font appel à des systèmes à vues multiples, c'est-à-dire utilisant simultanément des capteurs de données provenant de divers domaines (microphones et caméras vidéos) afin d'améliorer la précision et/ou l'interprétation des résultats (Essid et al., 2018; Padois et al., 2018). Le secteur de l'aéroacoustique, très actif en gestion des nuisances sonores, publie plusieurs études et persiste dans l'amélioration de techniques d'analyse dont l'imagerie acoustique est la base (Goudarzi et al., 2021).

Cette technique permet d'identifier les directions de provenance des énergies acoustiques en présence au point de mesure. Cependant, dans les cas où les sources sont éloignées de l'antenne ou pour les fréquences sonores basses ($<\approx$ 600 Hz), les différences mesurées entre les capteurs de l'antenne deviennent quasi nulles, et l'antenne ne peut alors que fournir des informations à faible résolution. On pourra alors compenser ce phénomène en

augmentant les dimensions de l'antenne, ce qui oblige à une augmentation du nombre de capteurs si on veut en conserver l'efficacité. C'est ainsi que, en pratique, les mesures par antenne microphonique seule sont peu applicables au cas de l'acoustique environnementale, car les antennes appropriées atteindraient des dimensions exagérées (Heilmann et al., 2014).

1.2.1.2.3 Cartographie acoustique

Dans le cadre de projets aux objectifs similaires à celui-ci, c'est-à-dire l'identification des sources de bruit en environnement, des systèmes intégrant plusieurs capteurs ainsi que des données prémodélisées ont été développés afin de générer des cartographies acoustiques quasi instantanées. Dans le cas du système proposé par Mackenzie et al. (2015), par exemple, les cartes acoustiques produites, pratiquement en temps réel, informent les gestionnaires de chantiers des zones potentiellement critiques. Cependant, la résolution d'analyse ne permet pas de cibler directement l'opérateur coupable, et elle n'est pas automatique ; des humains doivent être dans la boucle et afin de prendre des décisions stratégiques, en interprétant une carte, dans le but de pallier le dépassement mesuré.

En outre, l'information obtenue se limite à l'identification de la répartition de l'intensité sonore moyenne présente dans une zone donnée. Bien qu'il soit également possible de procéder à des analyses plus complètes, par exemple en isolant des sections fréquentielles ou temporelles, cette cartographie ne constitue pas une identification précise des sources ponctuelles au sein d'un champ sonore complexe.

1.2.1.2.4 Sources en mouvement

Les antennes microphoniques montrent leurs limites lorsqu'il s'agit de traiter des sources en mouvement. En effet, bien que les techniques classiques d'imagerie acoustique permettent de localiser des zones d'intensité sonore, elles perdent en efficacité lorsqu'il s'agit de suivre des sources mobiles en raison de leur difficulté à capter simultanément les variations spatiales et temporelles de ces sources sonores.

Pour répondre à cette limite, la méthode CLEANT (Cousson et al., 2019) a été développée comme une extension de la méthode CLEAN (Merino-Martínez et al., 2019), qui visait initialement à améliorer la résolution des cartes acoustiques en supprimant les interférences et les lobes secondaires associés à la présence de sources multiples. CLEAN permet ainsi une localisation plus précise dans des environnements stationnaires, mais ne prend pas en compte la dynamique temporelle des sources.

CLEANT, cependant, permet de localiser dynamiquement les sources mobiles et de quantifier leur contribution sonore au fil du temps, même lorsqu'elles se déplacent dans un environnement complexe. Fonctionnant dans le domaine temporel, cette méthode peut s'avérer pertinente dans des contextes industriels ouverts, comme les ports maritimes, où les nuisances sonores sont souvent causées par des sources mobiles, variables et intermittentes.

Toutefois, bien que CLEANT permette une analyse précise des sources mobiles dans une zone locale, ses possibilités restent limitées dans le cadre de notre recherche. En effet, CLEANT n'établit pas de corrélation directe entre les données captées localement sur les quais et les mesures prises à distance, telles que celles effectuées dans une zone résidentielle affectée par le bruit. Cette absence de lien direct avec les captations distantes réduit la capacité de cette méthode à identifier de manière exhaustive les sources responsables d'un dépassement sonore mesuré à distance.

1.2.1.3 Piste 3 : L'apprentissage automatique par les réseaux de neurones profonds

Comme nous le démontrerons dans la revue de la littérature[2], l'apprentissage automatique par les réseaux de neurones profonds figure parmi les pistes émergentes pour l'analyse des signaux sonores. Ces modèles sont déjà largement utilisés dans des domaines tels que la classification automatique des sons en environnement urbain, la détection de sources

[2] Voir 2.2.3.4

sonores nuisibles, ou encore dans la bioacoustique pour cartographier les populations animales en territoire naturel. Grâce à leur capacité à apprendre des représentations complexes directement à partir des données brutes, les réseaux de neurones profonds ont prouvé leur efficacité pour traiter des signaux acoustiques variés, tout en s'adaptant à des environnements dynamiques.

Les réseaux convolutifs (CNN[3]), en particulier, ont démontré leur capacité à extraire automatiquement des caractéristiques pertinentes à partir de données complexes et multidimensionnelles, ce qui en fait un outil privilégié pour l'analyse des signaux sonores. En exploitant la structure spatiale et temporelle des données, ces réseaux peuvent détecter des motifs récurrents, même dans des environnements bruyants ou en présence de multiples sources sonores.

Toutefois, leur efficacité dépend fortement de la qualité des signaux sonores disponibles. Dans la littérature, les requis minimaux incluent généralement un rapport signal sur bruit (S/B) raisonnable, une résolution temporelle suffisante, ainsi qu'une bande passante adaptée à la nature des sources analysées (Heittola et al., 2018; Salamon et al., 2014). Ces conditions permettent au réseau de disposer d'informations pertinentes sans être noyé dans un bruit aléatoire non structurel.

Dans notre étude, nous posons l'hypothèse que les réseaux de neurones profonds, en particulier les CNN, peuvent aider à inférer des relations causales entre des mesures acoustiques prises à différentes distances. Contrairement aux approches précédentes, qui se concentrent sur la classification ou la séparation des sources, notre objectif est d'utiliser ces modèles pour identifier les sources principales responsables des dépassements sonores, et ce, en lien avec des captations effectuées à distance. Cette capacité à corréler les niveaux de bruit mesurés en zone sensible avec les sources actives à distance

[3] Convolutional Neural Networks

représenterait une avancée notable dans la gestion du bruit environnemental, offrant un potentiel d'innovation pour des solutions plus efficaces et précises.

Cette hypothèse sera explorée dans la suite de notre travail, en mettant en œuvre des réseaux de neurones pour modéliser ces corrélations complexes, et en testant leur capacité à apporter des réponses aux problématiques spécifiques liées à la gestion du bruit en environnement portuaire urbain.

1.3 Questions de recherche

À la suite à l'analyse de notre problématique, nous constatons les défis suivants :

- Les ports maritimes peuvent être une source de pollution sonore non négligeable ;
- L'environnement sonore des ports est complexe et difficile à caractériser.

De même, nous observons que les techniques d'identification des sources sonores recensées dans les domaines de la pollution sonore par les ports ou encore l'identification des sources de bruit en industrie présentent ces caractéristiques :

- L'analyse de corrélation de phase entre 2 capteurs distants est impossible ;
- La focalisation par imagerie acoustique et la cartographie acoustique :
    - présentent les informations d'intensité et de localisation relative uniquement ;
    - présentent une résolution décroissant avec l'augmentation de la distance propagation ;
    - sont d'autant plus complexes et coûteux que la zone à couvrir est grande.
- La notion de sources mobiles dans des environnements changeants présente des défis particuliers.
- Les réseaux de neurones profonds présentent une piste de solution prometteuse, bien que nous n'ayons recensé aucun travail existant qui permette d'établir la relation entre des sources sonores et des mesures distantes.

Ainsi, aucune de ces techniques ne permet, à tout le moins à elle seule, d'effectuer automatiquement l'identification précise des sources mobiles provoquant une nuisance

mesurée à distance. La question principale à laquelle cette recherche tentera de répondre est donc :

Comment identifier, en temps réel, la contribution d'une source sonore ponctuelle mobile à un champ sonore complexe distant ?

1.4 Objectif

Cette recherche vise à faire la preuve de concept d'une méthode exploratoire permettant d'évaluer si une source sonore mobile peut être identifiée, à partir d'une captation distante dans un environnement bruité, comme responsable d'une nuisance environnementale mesurée.

1.5 Organisation du mémoire

Ce mémoire est structuré en cinq chapitres principaux, chacun visant à développer un aspect particulier de notre recherche exploratoire.

Le premier chapitre présente le contexte de la problématique de nuisance sonore au port de Trois-Rivières, ainsi que les objectifs spécifiques de notre étude. Nous y présentons également les partenaires impliqués et décrivons les pistes envisagées a priori pour répondre à cette problématique.

Le Chapitre 2, Revue de la littérature, est divisé en deux parties principales. La première couvre les fondements théoriques nécessaires à la compréhension des techniques de traitement des signaux sonores et des algorithmes d'apprentissage automatique appliqués à l'identification des sources sonores. La seconde partie explore les travaux antérieurs en matière d'identification des sources sonores nuisibles, en examinant à la fois les approches traditionnelles et les techniques basées sur l'apprentissage automatique.

Le Chapitre 3, Méthodologie, décrit en détail l'approche expérimentale utilisée pour vérifier la faisabilité de la méthode proposée. Nous y détaillons les méthodes de collecte et de traitement des données, l'architecture du modèle utilisé, ainsi que les techniques

d’entraînement et d’évaluation appliquées pour valider expérimentalement le principe étudié.

Le Chapitre 4, Résultats et discussion, présente les résultats obtenus lors de l’entraînement et de l’évaluation du modèle. Nous y analysons les performances du réseau en contexte simulé, en justifiant les choix méthodologiques et en discutant des implications de ces résultats dans le cadre de cette preuve de concept.

Enfin, le Chapitre 5, Conclusion et perspectives, récapitule les principaux apports de cette étude exploratoire et suggère des pistes pour de futures recherches visant à évaluer l’application de cette méthode en contexte réel.

* * *

CHAPITRE 2 : REVUE DE LA LITTÉRATURE

Dans ce chapitre, nous présenterons d'abord les théories et concepts fondamentaux en lien avec le traitement sonore ainsi que l'apprentissage automatique par des réseaux neuronaux, puis nous effectuerons un survol des développements et travaux antérieurs à celui-ci, en matière d'identification des sources sonores nuisibles, sur lesquels nous appuyons notre recherche.

2.1 Section 1 : Fondements théoriques

Cette première section de notre revue de littérature constitue le socle sur lequel repose notre recherche visant à améliorer la gestion du bruit environnemental par l'identification des sources sonores. Comme point de départ, nous tenons pour acquis que les techniques fondamentales de traitement du signal sonore, telles que la transformée de Fourier discrète (DFT), la transformée de Fourier rapide (FFT) et la transformée de Fourier à court terme (STFT), sont bien connues (Allen & Rabiner, 1977; Bracewell, 2000; Cooley & Tukey, 1965). Ces méthodes permettent de transformer les signaux acoustiques, initialement représentés dans le domaine temporel comme des variations de pression acoustique au fil du temps, en une représentation fréquentielle et, dans certains cas, en données de phase. Comme nous le verrons, ces représentations sont largement utilisées pour l'extraction de caractéristiques en vue de l'analyse automatique des signaux sonores (Oppenheim & Schafer, 2009).

Par ailleurs, certaines techniques d'extraction de caractéristiques exploitent directement la représentation temporelle des signaux, notamment pour des tâches où des motifs temporels spécifiques sont pertinents (Picone, 1993). Dans ce mémoire, nous nous concentrons sur les représentations qui se prêtent à l'apprentissage automatique, en soulignant leur rôle dans l'amélioration des performances des modèles.

2.1.1 Représentation des signaux sonores pour les algorithmes d'apprentissage automatique

Pour qu'un algorithme d'apprentissage automatique puisse traiter et analyser efficacement des signaux sonores, il est impératif de les lui représenter sous une forme adéquate. En effet, la représentation des signaux sonores joue un rôle crucial dans l'efficacité de l'analyse automatique, et en particulier dans les performances des modèles utilisant l'apprentissage automatique. (Purwins et al., 2019). Comme nous l'écrivions plus haut, à l'état brut, les signaux sonores sont des enregistrements de variations de pression acoustique au fil du temps, une forme qui, bien que riche en informations, est souvent trop complexe et non structurée pour une utilisation directe par des algorithmes d'apprentissage. Ainsi, une étape essentielle du prétraitement des données consiste à transformer ces signaux temporels en représentations plus structurées et compactes, qui mettent en avant les caractéristiques pertinentes tout en réduisant le bruit et les redondances.

Dans cette section, nous allons détailler ces différentes étapes de prétraitement et les techniques associées, en soulignant l'importance de chacune pour améliorer l'efficacité et la précision des algorithmes d'apprentissage automatique. En comprenant ces fondements, nous serons mieux à même d'aborder, à la suite, les techniques d'extraction de caractéristiques.

Selon Richard et al. (2013), les techniques ayant pour objectif la reconnaissance ou la classification automatique des sons reposent sur 2 fonctions principales : un module de représentation paramétrique du signal et un module d'analyse pour la classification. De façon complémentaire, Alías et al. (2016) schématisent ainsi les systèmes plus généraux d'analyse automatique des signaux sonores (traduction libre) :

Signal d'entrée → Fenêtrage → Extraction de caractéristiques → Analyse → Sortie

Nous définissons ces étapes ci-dessous.

2.1.1.1 Signal d'entrée : Prétraitement des données

Le prétraitement des signaux audio constitue une étape préparatoire importante avant l'extraction de caractéristiques. Dans la littérature, il est présenté que cette étape a pour but d'améliorer certaines propriétés du signal afin d'optimiser les performances des algorithmes d'analyse ultérieurs. Cela peut inclure la réduction du bruit, la mise en valeur des composantes pertinentes du signal, ou encore l'adaptation du format des données (Heittola et al., 2018).

Comme le soulignent Katti et Sumana (2022) :

> Tout modèle d'apprentissage automatique ou profond nécessite des données. Et toutes les données collectées doivent d'abord être analysées, nettoyées et prétraitées. La plupart des articles se concentrent davantage sur le modèle construit ou l'algorithme déployé, tandis que le prétraitement des données n'est mentionné qu'en une ou deux lignes, comme n'importe quel autre processus de la chaîne […][4].

Ainsi, bien que n'étant souvent mentionné que brièvement par les auteurs, le prétraitement des données est une étape cruciale. Dans cette section, nous structurons notre revue de littérature autour de cinq grandes étapes généralement reconnues dans le prétraitement des signaux audio :

- *Uniformisation du format* : Assurer que tous les fichiers audio sont dans un format cohérent et compatible avec les outils d'analyse utilisés.
- *Détection des parties pertinentes* : Identifier et extraire les segments pertinents du signal audio pour l'analyse, en évitant les portions non informatives.
- *Débruitage* : Atténuer les bruits de fond et, dans certains cas, séparer des sources intriquées.
- *Normalisation* : Ajuster les niveaux de volume pour permettre une analyse comparative.

[4] Traduction depuis l'anglais par l'auteur.

- *Adaptation au format d'input de l'analyse* : Convertir les données audio dans un format spécifique requis par les algorithmes d'analyse.

Ces étapes serviront de fil conducteur pour présenter les travaux existants sur le prétraitement des données audio dans les sections suivantes.

2.1.1.1.1 Uniformisation du format

Les enregistrements sonores numérisés devront être convertis vers des paramètres identiques de format, nombres de canaux, fréquence d'échantillonnage et durée.

Tableau 2.1 : Exemples de valeurs pour l'uniformisation du format en vue de l'analyse automatique des sons.

Paramètre	Valeurs en exemple	Critères
Format multicanal	monophonique, stéréophonique, binaural…	Dépend du domaine d'application et des circonstances particulières. La majorité des auteurs utilisent le format monophonique, mais des projets spécifiques avancent que les données en multicanal peuvent ajouter des facteurs discriminants (Seo et al., 2019).
Fréquence d'échantillonnage	8 ; 16 ; 22,05 ; 44,1 kHz	Il s'agit d'un compromis entre l'étendue du registre (principe de Nyquist), la résolution temporelle et la charge computationnelle (Roneel V. Sharan & Tom J. Moir, 2016). Les travaux récents utilisent majoritairement 22,05 et 44,1 kHz. Selon Aljubayri (2023), la fréquence d'échantillonnage n'a pas d'incidence significative sur les résultats de la classification automatique.
Durée	Très variable (de quelques dizaines de ms jusqu'à plusieurs secondes)	Dépend du domaine d'application et des circonstances particulières. Il pourra s'agir de la longueur des extraits enregistrés ou d'une durée imposée pouvant impliquer le *remplissage par zéros*[5]. On retrouve des extraits plus courts, par exemple, pour la détection des émotions, et plus longs pour la détection des styles musicaux.

[5] *Padding* ou *zero-padding*, en anglais, sont courants.

2.1.1.1.2 Détection des parties pertinentes

Dans le cas d'analyse de sons captés en environnements véritables, on voudra s'assurer de ne pas surcharger le système de parties inutiles. On utilisera des critères et des techniques spécifiques pour sélectionner les segments pertinents. Selon Abeßer (2020), ces techniques de détection d'événements sonores spécifiques peuvent être classées en fonction de leur développement, soit avant, soit après la transition vers l'utilisation généralisée des réseaux de neurones profonds. Barchiesi et al. (2015) proposent une liste de plusieurs des techniques plus anciennes pouvant être directement appliquées dans le domaine temporel ou sur la transformée de Fourier et qui sont donc appliquées à l'étape du prétraitement en préparation aux algorithmes d'apprentissage automatique. Ils mentionnent notamment les analyses du taux de passage par zéro, du centroïde spectral, de la décroissance spectrale ainsi que celle des caractéristiques d'énergie par bande de fréquence (énergie/fréquence). Ces techniques sont utilisées en contexte d'analyse de la parole.

Plus récemment, Lange et al. (2024) ont proposé des modèles autorégressifs linéaires pour prévoir le bruit environnemental et, ainsi, détecter les émissions acoustiques spécifiques en utilisant les résidus entre la prédiction du modèle et le signal réel. Cependant, toujours selon Abeßer (2020), les techniques récentes de détection des événements sonores sont plus couramment effectuées au sein même des algorithmes d'apprentissage automatique.

2.1.1.1.3 Débruitage

Éliminer les bruits de fond et les interférences est crucial pour améliorer la qualité du signal et la précision de l'analyse (Li et al., 2015; Wang et al., 2022). À l'étape du prétraitement, les techniques de filtrage, telles que les filtres de Wiener (Nuha & Absa, 2022) ou les filtres passe-bas ou passe-haut (Arora, 2017) sont couramment utilisées pour atténuer les bruits indésirables.

D'autre part, l'interférence des sons qui se chevauchent peut être minimisée en utilisant des techniques de séparation des sources sonores (Heittola et al., 2018). Burred (2009) ou

encore Sarkar (2024), par exemple, font état de méthodes basiques comme la capture par des techniques stéréophoniques utilisant les différences temporelles et fréquentielles entre les points de capteurs afin de distinguer les parties d'un signal[6]. Cependant, les techniques plus avancées et récentes qui sont proposées en vue de la séparation des sources en chevauchement au sein d'un signal sonore s'effectuent au sein des réseaux de neurones profonds plutôt qu'à l'étape de prétraitement (Luo et al., 2017).

2.1.1.1.4 Normalisation de l'amplitude

La normalisation consiste à ajuster les niveaux des signaux pour que toutes les données aient des amplitudes comparables. Cela peut inclure des techniques comme la normalisation par l'amplitude maximale ou la normalisation par z-score pour standardiser les données (Katti & Sumana, 2022).

2.1.1.1.5 Adaptation au format d'input

Cette dernière étape de prétraitement des données consiste à formater ces dernières conformément au type d'analyse automatique choisi. Ces formats ont évolué au fil du temps, chaque étape apportant des améliorations en précision et en capacité à traiter des données complexes. Nous proposons ci-dessous une vue sommaire des formats 1D à 4D, qui sont les plus pertinents et couramment utilisés pour l'analyse sonore automatique :

La représentation 1D des données audio correspond à la capture directe des signaux sonores sous forme de séries temporelles, où l'amplitude du signal est enregistrée à intervalles réguliers. Cette méthode a été largement utilisée dans les premières études de traitement du signal pour des tâches telles que la détection de silence, la segmentation de la parole, et la détection de pics. Les premières applications de la représentation 1D remontent aux années 1960 et 1970 alors que les systèmes de traitement numérique du signal ont commencé à être développés. Les travaux pionniers de Alan V. Oppenheim et

[6] Ces techniques sont utilisées en contexte d'analyse du signal musical.

Ronald W. Schafer dans leur livre *Digital Signal Processing* ont établi les bases du traitement du signal audio en utilisant des représentations temporelles 1D (Oppenheim & Schafer, 1975).

La représentation 2D des données audio est obtenue en appliquant des transformations temporelles et fréquentielles, telles que la transformée de Fourier, pour générer des spectrogrammes. Ces représentations montrent la façon dont le contenu fréquentiel du signal évolue au fil du temps. Utilisée pour des tâches comme la reconnaissance vocale, la classification audio et l'analyse musicale, cette méthode permet de capturer les dynamiques temporelles et fréquentielles des signaux. Les années 1980 et 1990 ont vu une adoption croissante des représentations 2D avec l'essor des techniques de reconnaissance vocale et de classification audio. Les travaux de Lawrence Rabiner et Biing-Hwang Juang, notamment dans *Fundamentals of Speech Recognition*, ont été déterminants pour l'adoption des spectrogrammes en tant que standard pour la reconnaissance vocale (Rabiner & Juang, 1993).

Avec l'avènement des modèles d'apprentissage profond, l'utilisation de tenseurs est devenue essentielle pour traiter des données multidimensionnelles. Les tenseurs sont des objets mathématiques qui généralisent les matrices à un nombre variable de dimensions (Goodfellow et al., 2016). Ils permettent de gérer des formats plus complexes et d'intégrer diverses sources d'information. Stevens et al. (2020) écrivent :

> Comparés aux tableaux NumPy, les tenseurs PyTorch possèdent quelques atouts, comme la capacité d'effectuer des opérations très rapides sur des unités de traitement graphique (GPU), de répartir les opérations sur plusieurs dispositifs ou machines, et de suivre le graphique de calculs qui les a générés. Ce sont des fonctionnalités essentielles pour la mise en œuvre d'une bibliothèque d'apprentissage profond moderne.[7]

[7] Traduction depuis l'anglais par l'auteur.

Ainsi, les tenseurs 3D ajoutent une dimension supplémentaire aux représentations 2D, permettant d'inclure des canaux multiples à l'analyse (par exemple, les enregistrements stéréo ou les matrices de microphones). Les tenseurs 3D permettent donc de modéliser les interactions entre différents canaux audio. Les années 2000 ont vu une adoption croissante des tenseurs 3D avec l'émergence des réseaux de neurones convolutionnels (3D CNN) pour l'analyse audio. Les travaux sur les architectures de CNN appliquées à l'audio, comme ceux de Abdel-Hamid et al. (2014) dans *Convolutional Neural Networks for Speech Recognition* ont été déterminants pour l'utilisation des tenseurs 3D dans la reconnaissance vocale. Les formats courants de tenseurs 3D sont [num_channels, height, width], ou [batch_size, height, width].

Les tenseurs 4D, quant à eux, structurent les données selon le format [batch_size, num_channels, height, width], facilitant l'utilisation de CNN pour analyser les caractéristiques spatiales et temporelles des signaux audio. Largement utilisés dans les cadriciels[8] d'apprentissage profond comme PyTorch, TensorFlow/Keras, MXNet et CNTK, les tenseurs 4D sont essentiels pour les applications nécessitant une analyse avancée des signaux audio et vidéo.

2.1.1.2 Fenêtrage

Après le prétraitement des données, la seconde étape système d'un système d'analyse automatique des signaux sonores consiste à diviser le signal en échantillons de longueur finie afin de convertir le signal audio typiquement non stationnaire en signal considéré quasi stationnaire au sein de chaque trame. La longueur des fenêtres ainsi que leurs durée et type de chevauchement feront l'objet de réglages adaptés aux applications spécifiques. Dans le cadre de la présente étude, cette action de fenêtrage est réalisée au sein de la STFT.

---

[8] *Frameworks*, en anglais, est plus courant.

R. V. Sharan et T. J. Moir (2016) résument ainsi ces ajustements et leurs réglages courants dans le domaine de la reconnaissance automatique des sons :

> Le prétraitement du signal vise à préparer le signal sonore pour l'extraction de caractéristiques. En général, un signal est divisé en segments plus petits, souvent de 10 à 30 ms, et une fonction de fenêtrage est appliquée pour lisser le signal en vue d'une analyse ultérieure. La fenêtre de Hamming semble être le choix préféré dans la plupart des systèmes de reconnaissance automatique des sons. […] En fonction de la fréquence d'échantillonnage du signal, une taille de trame de 256, 512 ou 1024 échantillons est habituellement choisie, avec un certain chevauchement entre les trames adjacentes, tel que 25 % ou 50 %, pour éviter la perte d'information aux bords de la fenêtre.[9]

2.1.1.3 Extraction de caractéristiques

L'objectif de cette 3e étape est d'obtenir une représentation compacte des caractéristiques des signaux sonores qui seront le plus à même d'optimiser l'efficacité de l'analyse subséquente. Notons que l'inclusion des changements temporels du signal sonore à cette représentation pourra provoquer une très grande dimensionnalité des vecteurs de caractéristiques. Ainsi, on pourra appliquer des procédés de réduction de la dimensionnalité des données afin de compacter ces vecteurs. Nous nous attarderons plus loin aux différentes techniques d'extraction de caractéristiques (2.1.2), car celles-ci sont variées et font l'objet de nombreuses expérimentations en recherche sur l'analyse automatique des signaux sonores.

2.1.1.4 Analyse

Dans le cas de l'analyse automatique des signaux sonores, il s'agira de l'algorithme d'apprentissage automatique ou de prédiction. Nous présentons cette étape ultérieurement dans le texte car il s'agit du cœur de ce travail de recherche (2.1.3).

---

[9] Traduction depuis l'anglais par l'auteur.

2.1.1.5 Sortie

Cette dernière étape d'un système d'analyse automatique des signaux sonores est cruciale car elle constitue le point de convergence des résultats obtenus à partir des étapes précédentes. La sortie d'un tel système peut prendre diverses formes en fonction de l'application spécifique et des objectifs du traitement du signal. Les exemples suivants illustrent la diversité et l'importance des applications possibles de ces systèmes, démontrant leur polyvalence et leur impact sur divers domaines. Nous présentons ces applications selon un niveau croissant de complexité :

- *Classification des genres musicaux* (Tzanetakis & Cook, 2002) : Les systèmes peuvent analyser des morceaux de musique et les catégoriser automatiquement en genres spécifiques tels que le jazz, le rock, la musique classique, etc. Cette capacité est particulièrement utile pour les services de diffusion musicale en continu et les bibliothèques de musique en ligne. Les algorithmes sont relativement simples, comme les k-plus proches voisins (k-NN) ou les machines à vecteurs de support (SVM)[10].
- *Indexation et recherche audio* (Foote, 1997) : Essentielle pour la gestion des grandes bases de données audio, cette application permet de générer des métadonnées descriptives pour des fichiers audio, facilitant une recherche efficace et une récupération rapide des informations pertinentes. Elle est exploitée dans les archives numériques, les médias sociaux et les plateformes de partage de contenu. Elle implique l'extraction de caractéristiques et la génération de métadonnées, souvent via des techniques de traitement du signal et de bases de données.
- *Identification de locuteurs* (Kinnunen & Li, 2010) : Utilisée pour reconnaître et distinguer différentes voix humaines, cette technologie est essentielle dans les domaines de la sécurité et de la biométrie, notamment pour les systèmes de reconnaissance vocale pour le contrôle d'accès et les services d'authentification par la voix. Utilise des techniques de reconnaissance vocale (ASR[11]) et des modèles de correspondance de caractéristiques, avec des défis modérés en termes de variabilité de la voix ainsi que de bruit.
- *Détection de défaillances mécaniques* (Lei et al., 2020; Randall, 2021) : Dans les industries, les systèmes d'analyse sonore sont utilisés pour détecter les défaillances

---

[10] k-NN : *k-nearest neighbors* ; SVM : *Support Vector Machines*

[11] *Automatic Speech Recognition*

des machines en analysant les vibrations qu'elles produisent. Cela permet une maintenance prédictive et réduit les temps d'arrêt. Cette discipline implique l'analyse de signaux vibratoires, souvent en utilisant des algorithmes de traitement du signal avancés et de l'apprentissage automatique.

- *Sécurité publique* (Crocco et al., 2016; Valenzise et al., 2007) : Les systèmes d'analyse sonore peuvent détecter des sons spécifiques associés à des événements dangereux, comme les coups de feu ou les explosions, et sont utilisés dans les systèmes de surveillance et les dispositifs de sécurité publique. Nécessite la détection de sons spécifiques dans des environnements bruités, souvent en temps réel, avec des défis importants en matière de précision et de rapidité.
- *Reconnaissance des sons environnementaux (ESC[12])* (Barchiesi et al., 2015) : Cette application permet aux systèmes d'identifier et de classifier divers sons de l'environnement, comme les bruits de la nature, les sons urbains, ou les alertes sonores. Ces systèmes sont utilisés dans des applications telles que la surveillance de la pollution sonore et la surveillance de populations animales. Implique la classification d'un large éventail de sons variés, nécessitant des algorithmes robustes pour gérer la diversité et le bruit.
- *Interaction homme-machine* (Amodei et al., 2016; Deng & Li, 2013) : L'analyse des signaux sonores joue un rôle crucial dans le développement des interfaces vocales et des assistants personnels intelligents, améliorant l'interaction naturelle entre les utilisateurs et les machines. Utilise des modèles de traitement du langage naturel (NLP[13]) et de reconnaissance vocale avancés, nécessitant une intégration fluide pour une interaction naturelle.
- *Analyse de la scène auditive computationnelle (CASA[14])* (Wang & Brown, 2006) : Cette technologie permet aux systèmes de modéliser les environnements acoustiques complexes, incluant la détection et la localisation des sources sonores multiples, la séparation des sons et la reconstruction des scènes auditives. Elle est appliquée dans les systèmes de réalité virtuelle et augmentée, les prothèses auditives intelligentes et les dispositifs de communication assistée. Il s'agit d'une technologie complexe impliquant la séparation des sources sonores et la modélisation d'environnements acoustiques, avec des algorithmes sophistiqués pour la localisation et l'analyse simultanée de multiples sources sonores.
- *Surveillance de la santé* (Li et al., 2017) : Les systèmes d'analyse sonore sont utilisés pour surveiller la santé à travers la détection des sons physiologiques tels

---

[12] *Environemental Sound Classification*

[13] *Natural Language Processing*

[14] *Computational Auditory Scene Analysis*

que la toux, la respiration et les battements cardiaques. Ces applications sont cruciales dans les dispositifs de santé portables et les systèmes de surveillance des patients. Nécessite l'intégration de capteurs spécialisés et l'analyse de signaux physiologiques, souvent en temps réel, avec des exigences élevées en matière de précision et de fiabilité pour des applications critiques de santé.

2.1.2 Techniques d'extraction de caractéristiques[15]

Comme nous l'avons mentionné plus haut, l'analyse automatique du signal sonore repose sur la capacité à extraire des informations pertinentes à partir des données brutes. Les techniques d'extraction de caractéristiques sont donc cruciales, car elles déterminent la qualité et l'efficacité de l'analyse subséquente. Les caractéristiques extraites doivent être à la fois représentatives des propriétés essentielles des signaux sonores et suffisamment compactes pour permettre une manipulation et une interprétation efficaces par les algorithmes d'apprentissage automatique. Compte tenu de la diversité des signaux sonores et des contextes d'application, il est nécessaire de s'attarder sur les différentes techniques d'extraction de caractéristiques pour identifier les méthodes les plus adaptées à chaque type de signal et à chaque objectif d'analyse. Dans cette section, nous explorerons les principales techniques utilisées, en soulignant leurs avantages, leurs limitations et les critères de choix en fonction des besoins spécifiques de l'analyse automatique du signal sonore.

La littérature scientifique fait état d'une variété de formats de vecteurs de caractéristiques pour l'analyse des signaux sonores ; nous mentionnerons ici ceux que notre recherche a permis d'identifier comme étant les plus significatifs.

Selon G. Sharma et al. (2020), les techniques d'extraction de caractéristiques peuvent être classées selon un continuum en augmentation de complexité suivant la chronologie de leur développement :

---

[15] *Feature extraction*, en anglais, est usuel.

Domaine temporel → fréquentiel → temps-fréquence → descripteurs profonds

2.1.2.1 Caractéristiques temporelles

Développées jusqu'à la fin des années 1950, ces techniques extraient les caractéristiques directement du signal audio dans le domaine temporel. Elles incluent des mesures comme l'énergie du signal, la puissance, le taux de passage par zéro (ZCR[16]), la détection de l'enveloppe ADSR[17], ou le centroïde temporel. Bien que moins performantes en contexte d'analyse des sons par réseaux de neurones, ces caractéristiques sont encore utilisées en prétraitement des données avant d'être fournies en entrée à des techniques d'extraction plus complexes, en particulier pour des tâches où les signaux audio doivent être filtrés selon des critères simples[18].

2.1.2.2 Caractéristiques fréquentielles

Développées principalement pendant les décennies 1950-60, ces techniques extraient l'information fréquentielle du signal à l'aide de la transformée de Fourier ou de l'analyse autorégressive. Des exemples de ces techniques sont le centroïde spectral, l'extraction des coefficients LPC[19] ou encore les coefficients cepstraux en fréquences mel (MFCC[20]). Nous présenterons les MFCC en détail plus bas dans ce texte.

---

[16] *Zero-crossing rate*

[17] ADSR signifie *Attack*, *Decay*, *Sustain*, *Release*. Il fait référence aux 4 phases principales de l'enveloppe sonore utilisée en acoustique, synthèse sonore, et traitement du signal pour caractériser l'évolution temporelle de l'amplitude d'un son.

[18] On pourra, par exemple, utiliser le ZRC afin de détecter les parties d'un flux sonore qui contiennent des sons à contenu tonal.

[19] *Linear Predictive Coding*

[20] *Mel-Frequency Cepstral Coefficients*

Caractéristiques *temps-fréquence* : Ces caractéristiques, développées au courant des décennies 1960 à 90, combinent des informations temporelles et fréquentielles. Elles utilisent des techniques qui maintiennent la résolution dans les deux domaines, notamment par l'application de fenêtres temporelles glissantes. Les méthodes typiques incluent la STFT, les spectrogrammes et la transformée en ondelettes.

La distinction entre ces deux derniers types d'extraction de caractéristiques réside dans leur approche de l'analyse du signal. En effet, les caractéristiques strictement en domaine fréquentiel, bien que pouvant s'appliquer sur des fenêtres temporelles s'enchaînant, examinent le signal de chaque échantillon dans son ensemble sans prendre en compte les variations temporelles spécifiques, alors que les caractéristiques temps-fréquence permettent une analyse plus détaillée en prenant en compte l'évolution des fréquences au cours de l'échantillon.

2.1.2.3 Descripteurs profonds[21]

Toujours selon ce classement par G. Sharma et al. (2020), les techniques d'extraction de caractéristiques par des méthodes profondes sont une évolution des précédentes. En effet, elles impliquent l'utilisation de réseaux de neurones profonds, comme les CNN ou les réseaux de neurones récurrents (RNN[22]), pour apprendre automatiquement des représentations complexes des signaux audio. Ces réseaux peuvent ingérer divers formats de signaux sonores, y compris la forme d'onde brute, les spectrogrammes, les spectrogrammes mel et les MFCC, pour extraire des caractéristiques pertinentes pour des tâches spécifiques. Ils peuvent ainsi utiliser plusieurs formats et « choisir » les représentations les plus efficaces (Deng & Yu, 2014). Contrairement aux méthodes traditionnelles, où les caractéristiques doivent être définies manuellement, les réseaux

[21] *Deep Features* dans l'article original.

[22] *Recurrent Neural Network*

profonds peuvent découvrir automatiquement des représentations hiérarchiques des données.

Tableau 2.2 : Sommaire du classement des techniques d'extraction de caractéristiques conformément à G. Sharma et al. (2020)

Catégorie	Techniques d'extraction de caractéristiques	Exemples
Temporelles	Caractéristiques extraites directement du signal dans le domaine temporel	- ZCR - Détection de l'enveloppe ADSR - Temps d'attaque logarithmique - Calcul de l'énergie à court terme - Centroïde temporel - Autocorrélation - …
Fréquentielles	Caractéristiques obtenues après transformation du signal en domaine fréquentiel	- Centroïde spectral - Extraction des coefficients LPC - Décroissance spectrale (*Spectral Rolloff*) - Flux spectral - MFCC - …
Temps-fréquence	Caractéristiques capturant à la fois des informations temporelles et fréquentielles	- STFT - Spectrogramme - Transformée en ondelettes - Représentation en temps-fréquence (CQT[23]) - …
Méthodes profondes	Caractéristiques apprises automatiquement par des réseaux de neurones profonds	- CNN - RNN - Autoencodeurs - Réseaux de neurones convolutifs récurrents (CRNN[24]) - Transformers Formats de signaux sonores : forme d'onde brute, spectrogrammes, spectrogrammes mel, MFCC, représentations en ondelettes.

[23] *Constant-Q Transform*

[24] *Convolutional Recurrent Neural Network*

D'autre part, selon Mitrović et al. (2010), les formats d'extraction de caractéristiques peuvent aussi être classés en 2 grandes familles, soit les techniques *physiques* et les techniques *perceptuelles* :

2.1.2.4 Techniques Physiques :

Ces techniques se concentrent sur les propriétés physiques mesurables des signaux audio. Elles incluent des mesures basées sur l'énergie, la fréquence et d'autres paramètres directement dérivés des signaux bruts. Nous proposons ici quelques exemples de techniques physiques significatives dans la littérature récente en lien avec l'analyse automatique des signaux sonores :

- *Spectrogrammes* : Représentation visuelle des fréquences présentes dans un signal audio au fil du temps, obtenue en appliquant la transformée de Fourier à des segments successifs du signal. Utilisés comme entrée pour les CNN, ce qui en fait une technique populaire pour l'analyse audio récente. Les spectrogrammes sont utilisés pour la reconnaissance vocale, l'analyse musicale et la classification des sons (Bianco et al., 2019).
- *Codage prédictif linéaire (LPC)* : Technique utilisée pour modéliser le spectre d'un signal audio en prédisant les échantillons futurs basés sur les échantillons passés. Permet une représentation compacte et efficace des signaux en réduisant la redondance dans le signal. Le LPC est largement adopté pour l'extraction des caractéristiques vocales (Prabakaran & Shyamala, 2019).
- *Transformée en ondelettes* : Technique de décomposition multirésolution utilisée pour l'analyse des signaux à différentes échelles temporelles et fréquentielles. La transformée en ondelettes est très polyvalente et fondamentale pour l'analyse fréquentielle (Lee & Kwak, 2023). Elle est notamment utilisée dans le domaine de l'empreinte audio (Huang, 2024).
- *Analyse cepstrale* : Le cepstre est le résultat de l'application du logarithme sur le spectre de fréquence suivi de la transformée de Fourier inverse de ce spectre logarithmique. Ceci transforme le spectre logarithmique en domaine temporel, révélant des informations sur les structures périodiques du signal original. D'abord développée dans le cadre du traitement de la parole, elle est utilisée également pour le traitement des autres signaux audio, l'analyse des signaux biomédicaux et le diagnostic des machines industrielles. (Ibarra-Zarate et al., 2019).

2.1.2.5 Techniques perceptuelles :

Ces techniques se basent sur des modèles de perception humaine du son. Elles sont conçues pour capturer les aspects du signal audio qui correspondent à la façon dont les humains perçoivent les sons, souvent en utilisant des échelles non linéaires ou des modèles auditifs. Nous proposons ces exemples de techniques perceptuelles significatives dans la littérature récente en lien avec l'analyse automatique des signaux sonores :

- *MFCC* : Il s'agit ici d'une analyse cepstrale basée sur une échelle de fréquences mel, qui tâche de modéliser la sensibilité de l'oreille humaine aux différentes fréquences. Les MFCC sont largement utilisés dans les systèmes de reconnaissance de la parole (Abayomi-Alli et al., 2022). Cette technique est également très présente dans la littérature récente pour ses performances robustes et sa pertinence dans les tâches d'analyse audio, notamment dans le domaine de l'ESC (Al-Hattab et al., 2021; Su et al., 2019). Comme la technique d'extraction de caractéristiques que nous avons choisi d'utiliser dans le cadre de cette recherche est celle des MFCC, nous nous y attarderons plus bas dans ce texte (voir 2.1.2.7).
- *Spectrogrammes mel* : Les spectrogrammes mel utilisent une échelle de fréquences mel pour transformer le spectre de fréquences d'un signal audio en une représentation plus proche de la façon dont l'oreille humaine perçoit les sons. Il s'agit, de fait, du même traitement que dans le cas des MFCC, mais sans l'étape finale de Transformée de cosinus discrète (DCT) (voir 2.1.2.7.1). Les spectrogrammes mel continuent d'être utilisés en analyse automatique des signaux sonores, souvent en comparaison des performances d'autres techniques dans le contexte de réseaux CNN (Al-Hattab et al., 2021; Hafiz et al., 2023).
- *Coefficients perceptuels de prédiction linéaire* (PLP[25]) : Caractéristiques perceptuelles dérivées du codage prédictif linéaire (LPC). Les PLP améliorent le LPC par l'ajout de variables dans les domaines dynamique (compression non linéaire) et fréquentiel (ex. : échelle de Bark). Les coefficients PLP sont utilisés pour capturer les caractéristiques perceptuelles de la parole (ex. : Zaidi et al., 2021).
- *Caractéristiques chromatiques*[26] : Distribution de l'énergie spectrale dans les 12 classes de hauteur musicale, correspondant au système tonal de la musique occidentale. Les caractéristiques chromatiques sont donc efficaces pour capturer

---

[25] *Perceptual Linear Predictive Coefficients*

[26] *Chroma Features*, en anglais, est plus usuel.

les informations harmoniques et tonales dans la musique. Elles sont utilisées dans la classification des genres musicaux et d'autres tâches d'analyse musicale (Zalkow & Müller, 2021).

2.1.2.6 Choix de format d'extraction de caractéristiques

Alías et al. (2016), dans le cadre de leur revue des techniques d'extraction de caractéristiques physiques et perceptuelles pour la parole, la musique et les sons environnementaux, soulignent les différences significatives entre ces domaines en matière de complexité et de nature des signaux sonores. Les signaux vocaux et musicaux présentent des structures temporelles et harmoniques bien définies, tandis que les sons environnementaux sont souvent plus variés et bruités, rendant leur analyse plus complexe. Cette distinction est fondamentale à l'heure de choisir des techniques d'extraction de caractéristiques adaptées à chaque domaine. Dans le cadre de cette recherche, nous concentrons donc notre revue au domaine de l'analyse automatique des environnements sonores puisqu'il est le plus proche de notre sujet de recherche. Voici les principales variables à considérer dans ce contexte :

2.1.2.6.1 Temporalité et dynamique

Les sons environnementaux sont non structurés et présentent souvent des variations temporelles importantes. Les caractéristiques extraites doivent donc pouvoir représenter cette diversité. Karol J. Piczak (2015) a montré que l'utilisation de caractéristiques spectrotemporelles telles que les MFCC[27] permet de mieux capturer les variations dans les environnements sonores complexes, cependant que d'autres descripteurs

[27] Le cas des MFCC en tant que technique temps-fréquence est particulier : À l'instar du spectrogramme, ils utilisent la STFT comme première étape de transformation et pourraient donc être classés comme technique temps-fréquence. Toutefois, les étapes subséquentes du traitement, visant à créer des coefficients cepstraux à l'aide de DCT, créent des coefficients strictement du domaine fréquentiel. Cependant, l'utilisation d'un fenêtrage temporel paramétrable permet d'observer les changements de contenu fréquentiel selon les besoins particuliers à chaque situation.

spectrotemporels comme les coefficients d'ondelettes ou les flux chromatiques peuvent également être utilisés. L'ajout de caractéristiques temporelles comme les coefficients delta et delta-delta (1re et 2e dérivée) aux MFCC peut également améliorer la capacité à observer les changements rapides dans les signaux audio puisque les caractéristiques delta estiment le taux de changement instantané des valeurs des caractéristiques, et les delta-delta estiment leur accélération (Benetos et al., 2018; Heittola et al., 2018).

2.1.2.6.2 Dimensionnalité et représentativité

Il est crucial de trouver un équilibre entre la dimensionnalité des caractéristiques et leur capacité à représenter fidèlement les sons. Une dimensionnalité trop élevée peut entraîner des coûts computationnels importants et des problèmes de surapprentissage, tandis qu'une dimensionnalité trop faible peut ne pas capturer suffisamment d'informations. Les MFCC, souvent mentionnés dans notre recherche en raison de leur représentation compacte, offrent un bon compromis (Heittola et al., 2018; Sharan et al., 2021). Toutefois, des techniques avancées comme les autoencodeurs peuvent également être utilisées pour réduire la dimensionnalité tout en conservant les informations pertinentes (Bengio et al., 2013). Ces approches permettent d'optimiser la représentation des signaux sonores, en préservant un équilibre entre la simplicité computationnelle et la richesse informationnelle des caractéristiques extraites.

2.1.2.6.3 Discrimination et invariance

Les caractéristiques doivent permettre une discrimination efficace entre différents signaux sonores tout en étant invariantes aux variations non pertinentes (par exemple, les variations d'intensité ou de provenance). Les méthodes d'extraction doivent donc être choisies en fonction de leur capacité à minimiser la variance entre les instances à associer, tout en maximisant la distance entre les instances à dissocier. Les MFCC, lorsqu'ils sont combinés avec les techniques de normalisation décrites plus haut, peuvent offrir un bon équilibre entre discrimination et invariance, comme le démontre Karol J. Piczak (2015).

En outre, l'auteur démontre que les coefficients cepstraux en fréquence linéaire (LFCC[28]) et les descripteurs d'énergie spectrale peuvent également offrir de bonnes capacités de discrimination pour certains types de sons environnementaux.

2.1.2.7 Coefficients cepstraux en fréquences mel (MFCC)

Nous constatons que les MFCC sont une technique d'extraction de caractéristiques largement reconnue et utilisée dans le domaine de l'analyse sonore en raison de sa capacité à capturer les caractéristiques spectrales pertinentes des signaux audio tout en assurant une dimensionnalité raisonnable. Les MFCC se montrent efficaces pour représenter les caractéristiques phonétiques des sons, ce qui en fait un choix privilégié pour de nombreuses applications, notamment la reconnaissance automatique de la parole et la classification audio. Dans le cadre de notre étude, nous avons choisi d'utiliser les MFCC pour notre réseau de neurones convolutifs siamois avec fonction de perte par triplet, car nos recherches ont démontré que ce format est parmi les plus appropriés pour optimiser les performances de ce type de modèle. Cette section se propose d'explorer les fondements théoriques des MFCC, leur processus de calcul et les raisons de leur efficacité, en justifiant leur choix comme méthode d'extraction de caractéristiques dans notre contexte spécifique.

2.1.2.7.1 Fondements théoriques et processus de calcul

Davis et Mermelstein (1980) sont les auteurs de l'article *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*, souvent cité comme un des points de départ du développement de techniques de reconnaissance vocale automatique. Dans cette étude portant sur la reconnaissance automatique des mots à partir de la reconnaissance des syllabes, ils présentent la méthode

[28] *Linear Frequency Cepstral Coefficients*

des *Coefficients Cepstraux en fréquences mel* et démontrent qu'elle est la plus performante parmi les méthodes utilisées à l'époque[29].

Le MFCC est obtenu selon ces étapes :

1. Calcul de la transformée de Fourier du signal sonore ;
2. Pondération du spectre de puissance selon l'échelle de fréquences mel ;
3. Calcul de la transformée en cosinus discrète (DCT)[30] du log-mel-spectre.

La 1re étape consiste à appliquer une STFT afin d'obtenir le spectre de puissance du signal sonore sur des fenêtres temporelles fixes.

La 2e étape consiste à diviser le spectre fréquentiel selon une échelle inspirée de la façon dont les humains perçoivent les fréquences sonores. Ainsi, l'échelle des fréquences mel a pour objectif de correspondre à la perception des hauteurs sonores, pour une onde pure, en tenant en compte l'élargissement des octaves avec l'augmentation de la fréquence. La formule de conversion des Hz en mel est la suivante :

Équation 2-1 : Conversion Hz → mel

$$\boldsymbol{m = 1127 \cdot ln\left(1 + \frac{f}{700}\right)}$$

Les filtres mel sont donc espacés linéairement dans le registre des basses fréquences jusqu'à environ 1 kHz, et logarithmiquement dans le registre supérieur (Figure 2.1).

La 3e étape consiste à appliquer une DCT au log-mel-spectre obtenu précédemment. Cette opération a pour but de compresser les informations spectrales tout en réduisant leur corrélation, en produisant un ensemble de coefficients cepstraux. La DCT agit comme une opération de regroupement, où seuls les premiers coefficients, qui capturent les

---

[29] Le LPC et le *Linear Prediction Cepstral Coefficients* (LPCCs) sont les principales méthodes d'extraction qui précèdent le MFCC.

[30] *Discrete Cosine Transform*

composantes les plus significatives (essentiellement liées à l'enveloppe spectrale globale), sont retenus. Ces coefficients forment les vecteurs caractéristiques utilisés comme entrée pour des algorithmes d'apprentissage automatique.

Dans cette même publication, les auteurs présentent cette forme computationnelle pour le MFCC :

Équation 2-2 : Algorithme des MFCC

$$MFCC_i = \sum_{k=1}^{20} X_k \cos\left[i\left(k - \frac{1}{2}\right)\frac{\pi}{20}\right], i = 1, 2, \cdots, M,$$

Où M est le nombre de coefficients de cepstres, et X_k, k = 1,2,… 20 représente le log-énergie de l'ordre du filtre k.[31]

On obtient une séquence de vecteurs acoustiques qui seront utilisés lors des étapes subséquentes de l'analyse.

Notons que la validité psychoacoustique de l'échelle mel est controversée et a fait l'objet de propositions alternatives depuis sa publication (Moore & Glasberg, 1996; Prabakaran & Shyamala, 2019; Zwicker & Terhardt, 1980). Cependant, son utilisation au sein des MFCC et la grande popularité de cette méthode en font une technique largement prédominante en analyse automatique du signal sonore, comme l'affirment Heittola et al. (2018) : « Les caractéristiques acoustiques les plus couramment utilisées pour représenter le contenu spectral des signaux audio sont les énergies des bandes mel et les coefficients cepstraux en fréquence mel (MFCC) »[32].

---

[31] Cette version pour 20 filtres mel. La valeur *20* est remplacée par une variable dans les versions plus récentes de l'équation, puisque le nombre de filtres mel peut varier.

[32] Traduction depuis l'anglais par l'auteur.

Malgré leur conception initiale pour l'ASR, les MFCC ont également démontré une performance solide pour l'ESC. Par exemple, une étude par Mu et al. (2021) a montré que les MFCC, lorsqu'ils sont combinés avec des algorithmes d'apprentissage automatique, peuvent efficacement discriminer divers types de sons environnementaux, tels que les bruits urbains et les sons naturels. De plus, Virtanen et al. (2018) ont comparé différentes techniques de traitement des signaux et ont confirmé que les MFCC restent compétitifs en comparaison aux autres méthodes. Plus précisément, Serizel et al. (2018), au chapitre 4, écrivent :

> En particulier, les MFCC […] restent, même aujourd'hui, l'une des caractéristiques les plus largement utilisées dans la classification des sons depuis leur utilisation initiale pour le traitement de la musique par Logan (2000). Cela est surprenant, car les MFCC ont été conçus à l'origine pour traiter les signaux de parole et, en particulier, pour la reconnaissance de la parole (Davis & Mermelstein, 1980). En fait, les MFCC intègrent certaines propriétés de perception et, en référence au modèle classique source-filtre de production de la parole, éliminent principalement la partie source, rendant ainsi les MFCC relativement indépendants de la hauteur tonale. Une application directe des MFCC pour l'analyse de la musique et des sons environnementaux est surprenante, car (1) l'étendue des hauteurs tonales est bien plus large dans les sons en général que dans la parole ; (2) pour les hautes fréquences, la propriété de déconvolution des MFCC ne s'applique plus (les MFCC deviennent donc dépendants de la hauteur) et (3) les MFCC ne sont pas fortement corrélés avec les dimensions perceptuelles du « timbre polyphonique » dans les signaux musicaux, malgré leur utilisation répandue en tant que prédicteurs de similarité perçue du timbre (Alluri & Toiviainen, 2010; Mesaros & Virtanen, 2010; Richard et al., 2013). Il semble cependant que la capacité des MFCC à capturer les propriétés de l'enveloppe spectrale « globale » soit la principale raison de leur succès dans les tâches de classification des sons.[33]

Ceci est appuyé par plusieurs auteurs, dont Fang et al. (2021), Bonet-Solà et Alsina-Pagès (2021), ou encore Al-Hattab et al. (2021) dans leur article *Rethinking environmental sound*

[33] Traduction depuis l'anglais par l'auteur.

classification using convolutional neural networks : optimized parameter tuning of single feature extraction,[34] qui nous a servi de source pour le modèle de CNN choisi pour cette étude. En effet, dans cet article, les auteurs s'inscrivent en faux devant la tendance des chercheurs à suggérer des méthodes à plusieurs flux d'analyse et apprentissage automatique très profond, exigeant de plus en plus de ressources, afin d'augmenter la précision de la reconnaissance automatique. Ils suggèrent que l'on s'attarde davantage au paramétrage des algorithmes effectuant la représentation paramétrique du signal. La méthode qu'ils retiennent en tant que plus performante fait appel à un seul flux MFCC et à 3 couches de CNN[35].

2.1.2.7.2 Réglages MFCC

Outre les réglages de la fonction STFT, l'utilisation des MFCC requiert (1) l'ajustement des paramètres du nombre de filtres mel ainsi que (2) du nombre de coefficients cepstraux, en plus d'offrir (3) la possibilité d'exclure le 1er coefficient cepstral (C_0) ainsi que (4) d'extraire également les 1re et 2e dérivées des coefficients. Nous présentons ces réglages ci-dessous.

(1) Nombre de filtres mel

Le nombre de filtres mel doit être ajusté selon la bande passante, elle-même déterminée par la fréquence d'échantillonnage utilisée, selon le principe de Nyquist. Ainsi, le registre sera divisé selon le nombre de filtres choisi en élargissant ceux-ci de concert avec l'augmentation du registre. À titre d'exemple (Figure 2.1, 1er et 3e graphique), 24 filtres sous 10 kHz présentent la même résolution fréquentielle que 38 filtres sous 44,1 kHz. Ainsi, une augmentation de la fréquence d'échantillonnage sans ajustement du nombre de

---

[34] *Repenser la classification des sons environnementaux à l'aide de réseaux de neurones convolutionnels : optimisation des paramètres pour une extraction de caractéristiques unique*. (traduction par l'auteur)

[35] Voir 2.2.3.4.2.

filtres provoquera une perte de précision de l'information fréquentielle, particulièrement notable dans le registre grave. Cet ajustement pourra donc s'avérer nécessaire pour les études plus récentes utilisant des fréquences d'échantillonnage plus grandes[36].

De même, une augmentation du nombre de filtres, pour un registre donné, offrira à l'analyse subséquente une séparation fréquentielle plus fine, comme le démontrent les 2e, 3e et 4e gtaphiques de la Figure 2.1. Cette augmentation pourra s'avérer souhaitable en ESC puisque la technique originale des MFCC, développée pour l'ASR, cherchait strictement à représenter les formants des voyelles, ce qui exigeait un discernement fréquentiel moins fin (O'shaughnessy, 1987).

[36] Lors des années de développement embryonnaire de l'ASR, une f.é. = 10 kHz (jumellée à un filtre passe-bas à 5 kHz) était correcte puisqu'on cherchait à identifier les voyelles à l'aide de leurs formants les plus marqués, lesquels ne dépassaient pas 5 kHz. Cependant, les travaux plus récents, particulièrement en ESC, utilisent fréquemment une f.é. = 20,05, voire 44,1 kHz.

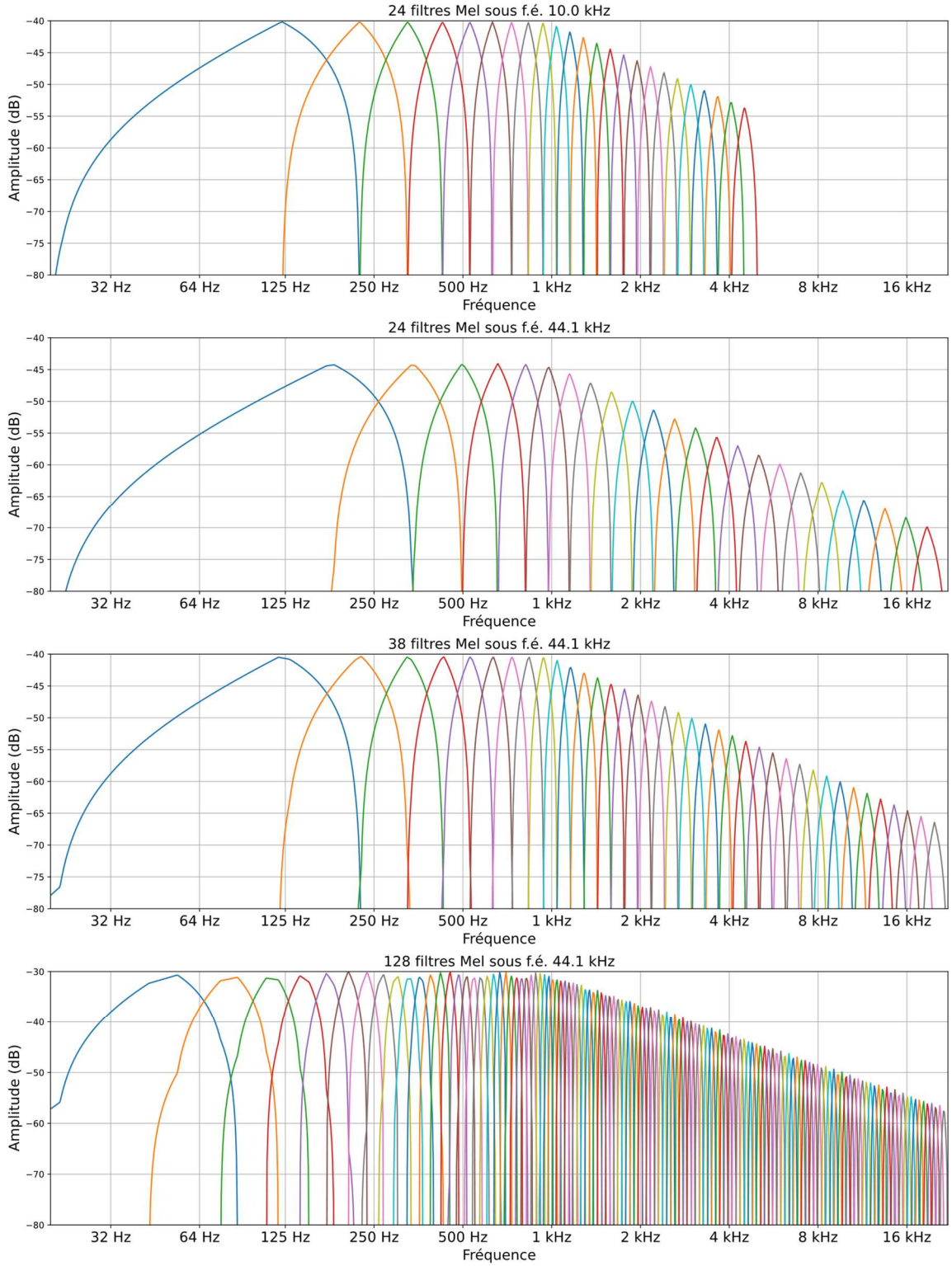


Figure 2.1 : Comparaison des filtres mel.

(2) Nombre de coefficients

Comme présenté précédemment, après la pondération du spectre de puissance selon l'échelle de fréquences mel, les coefficients de cepstre sont obtenus en appliquant la DCT au log-mel-spectre, conformément à l'Équation 2-2. En plus du nombre de bandes mel, il est nécessaire de spécifier le nombre de coefficients à extraire.

Les coefficients cepstraux reflètent les différences d'intensité dans les registres fréquentiels en indiquant les variations entre les différentes parties du spectre, chaque coefficient en représentant une division spécifique. Par exemple, le premier coefficient cepstral (C_0) témoigne de la différence entre la totalité du spectre et une absence de signal, alors que le deuxième coefficient représente la distribution du contenu fréquentiel basé sur le centroïde spectral, informant sur l'équilibre général entre les basses et les hautes fréquences. Ainsi de suite, chaque nouveau coefficient présente des variations plus fines entre les poids des parties fréquentielles du signal. Conséquemment, le 128e coefficient cepstral, par exemple, représente les différences d'intensité à une résolution très fine, concentrée sur les variations dans les sections très étroites du spectre fréquentiel (O'shaughnessy, 1987).

Le nombre de coefficients cepstraux à produire devra être adapté à la représentation nécessaire pour la tâche spécifique. En effet, une résolution fréquentielle plus fine n'est pas garante d'une meilleure performance (Zheng et al., 2001), notamment puisqu'elle pourra entraîner du surapprentissage ou de la confusion par des bruits polluants (Goodfellow et al., 2016).

Une autre considération est le fait que le nombre de coefficients cepstraux choisi ne doit pas dépasser celui des filtres mel. En effet, un nombre élevé de coefficients cepstraux est inutile si des coefficients capturent des différences au sein d'une même bande mel, lesquelles seront nulles. Une règle empirique courante bien que non documentée directement est que le nombre de bandes mel doit être au moins égal au nombre de

coefficients cepstraux plus 2. Par exemple, pour 13 coefficients cepstraux, on devrait utiliser au moins 15 bandes mel.

(3) Premier coefficient (C_0)

D'autre part, comme l'écrivent Zheng et al. (2001) dans *Comparison of Different Implementations of MFCC,* il est possible d'ignorer le premier coefficient cepstral :

> (…) dans de nombreux systèmes ASR, le 0e coefficient du cepstre MFCC est ignoré en raison de son manque de fiabilité [(Picone, 1993)]. En fait, le 0e coefficient peut être considéré comme une somme des énergies moyennes de toutes les bandes de fréquences du signal en cours d'analyse.[37]

Ce 1er coefficient, souvent appelé le coefficient de l'énergie moyenne (ou coefficient zéro [C_0]), représente principalement l'énergie globale du signal ; il pourra donc ne pas fournir d'information pertinente sur les caractéristiques de la forme spectrale. C_0 sera donc utile dans les cas spécifiques où cette distinction entre des sons basée sur l'amplitude est importante, comme dans le cas de la séparation entre le bruit de fond et un signal ciblé. Cependant, si les amplitudes sont normalisées puisqu'on cherche à distinguer des sons indépendamment de leur puissance, éliminer ce premier coefficient peut améliorer le contraste au sein de la matrice MFCC (Figure 2.2, 4e graphique). Ce sera le cas, par exemple, si on cherche à classifier des sons du même type mais à des distances variées du point de captation sonore.

Mentionnons ici que, bien que cette technique consistant à éliminer C_0 soit mentionnée dans les ouvrages moins récents, portant principalement sur le ASR, nous n'avons pas retrouvé cette précision dans les travaux de ces dernières années. Comme il n'y a pas, non plus, de raison vérifiée à l'effet du contraire, nous concluons que la pratique est tombée en oubli lors de l'application des fonctions standardisées des très populaires bibliothèques

[37] Traduction depuis l'anglais par l'auteur.

Librosa, Torchaudio et *TensorFlow*, lesquelles n'offrent pas directement ces paramètres sous forme d'arguments nommés pour la fonction MFCC.

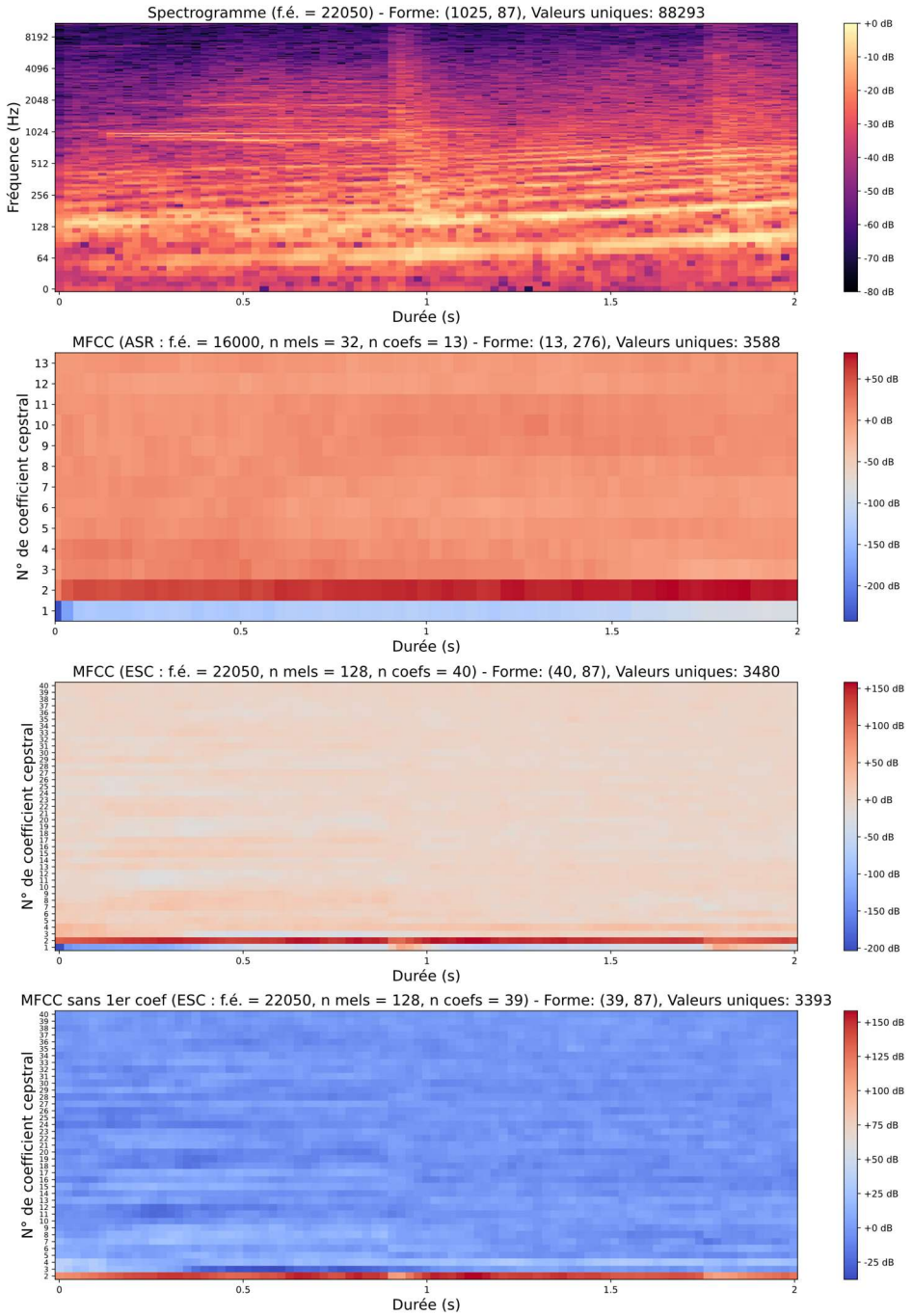


Figure 2.2 : Comparaison des représentations MFCC. A : Spectrogramme ; B : MFCC en réglages ASR ; C : MFCC en réglages ESC ; et D : MFCC en réglages ESC sans le 1er coefficient cepstral.

La Figure 2.2 présente des matrices MFCC en comparaison d'un spectrogramme. On y observe que :

- Les MFCC présentent une réduction de dimensionnalité substantielle en comparaison aux spectrogrammes (96 % de réduction) ;
- Les réglages ESC, pour les MFCC, ne présentent aucune augmentation de dimensionnalité depuis les réglages ASR, malgré le passage de 13 à 128 coefficients, ce qui augmente la finesse de la résolution fréquentielle. Ceci est dû à des réglages temporels moins serrés du côté de l'ESC[38], ce qui correspond à la proposition de Davis et Mermelstein (1980) décrite plus loin (2.1.2.7.3).
- L'élimination du 1er coefficient permet de dégager de meilleurs contrastes dans le reste de la matrice, sans augmentation de la dimension.

(4) Delta (1re dérivée) et delta-delta (2de dérivée)

Mentionnons finalement qu'il est également possible d'extraire les 1re et 2de dérivées des coefficients cepstraux. Comme l'expliquent Sidhu et al. (2024) :

> La première dérivée des MFCC est appelée delta, Δ, et est également connue sous le nom de *coefficient différentiel*. Le delta représente la variation d'un coefficient d'une trame à l'autre. La seconde dérivée des MFCC est appelée delta-delta, ΔΔ, et est également connue sous le nom de coefficient d'accélération. En définissant une caractéristique de f_k à un instant k, les coefficients delta et delta-delta peuvent être décrits dans les Équations 10 et 11 ci-dessous.
>
> $$\Delta_k = f_k - f_{k-1} \qquad (10)$$
>
> $$\Delta\Delta_k = \Delta_k - \Delta_{k-1} \qquad (11)$$
>
> Ces deux caractéristiques pourraient offrir une meilleure compréhension de la dynamique du spectre de puissance.[39]

[38] Ici, les réglages sont, pour ASR : FFT = 512, Hop = 160; et pour ESC : FFT = 2048, Hop = 512. Il s'agit donc d'une résolution temporelle meilleure pour l'ASR, au prix, cependant, d'une résolution fréquentielle inférieure.

[39] Traduction depuis l'anglais par l'auteur.

Les delta et delta-delta permettent donc d'obtenir la représentation des changements temporels dans le signal sonore, lesquels ne sont pas directement représentés par les coefficients cepstraux. Nous observons l'utilisation de ces dérivées principalement dans les travaux du domaine des sciences médicales (Boualoulou et al., 2022; Shahin et al., 2021; Sidhu et al., 2024). Notons que ces dérivées sont intégrées directement à la populaire librairie Librosa, les rendant assez simples à appliquer à l'heure de comparer les performances d'un système d'analyse selon leur usage ou pas.

2.1.2.7.3 Valeurs recensées dans la littérature

Selon nos recherches, les auteurs se montrent peu loquaces quant aux réglages MFCC qu'ils utilisent. Davis et Mermelstein (1980), dans leur article original, avaient toutefois présenté les principes généraux pouvant guider les choix de réglages. Par exemple, dans cet extrait, ils présentent le rapport entre la charge computationnelle et les paramètres à privilégier :

> Étant donné que les taux de reconnaissance pour six coefficients cepstraux et un espacement de trames de 6,4 ms sont assez comparables à ceux pour dix coefficients et un espacement de trames de 12,8 ms, augmenter le nombre de coefficients tout en maintenant une résolution temporelle légèrement plus grossière est plus avantageux d'un point de vue computationnel que d'utiliser moins de coefficients avec une fréquence plus élevée.[40]

À titre d'exemple, le Tableau 2.3 présente les réglages de paramètres MFCC que contiennent quelques publications du domaine de l'ESC que nous avons recensées. Nous

[40] Traduction depuis l'anglais par l'auteur.

constatons que plusieurs informations sont manquantes. En outre, dans ces articles, aucune discussion n'argumente les réglages choisis[41].

Tableau 2.3 : Paramétrages STFT et MFCC de quelques publications en exemples

Application observée	Échantillonnage PCM (kHz)	Fenêtre FFT (points)	Saut (points)	N. de bandes mel	N. de coefs	1er coef	Delta et Delta-delta
Boddapati et al. (2017)	32	Hamming 1024	512	32 ?	13 ?	?	Non ?
Su et al. (2019)	22,05	? 512	256	?	20	Oui ?	Oui
J. Sharma et al. (2020)	32	Hamming 1024	512	?	?	?	?
Al-Hattab et al. (2021)	44,1	Hamming/Hanning 2048	2048	128	128 ?	?	?
Hafiz et al. (2023)	16	? 2048	512 (?)	? (128?)	13	?	?

Originalement, Davis et Mermelstein (1980) proposaient les réglages suivants :

- Fréquence d'échantillonnage : 10 kHz
- Taille de la fenêtre FFT : 256 points (Hamming)
- Longueur du saut : 128 ou 64 points
- Nombre de filtres mel : 20
- Nombre de coefficients : 10

[41] Al-Hattab et al.(2021) comparent toutefois les performances de Hamming vs Hanning sous 2048/2048, ainsi que vs 1024/512 sous Hamming. Hamming 2048/2048 s'avère la combinaison la plus performante dans le cas spécifique de leur recherche.

Ces réglages représentaient un compromis efficace pour l'époque. Cependant, plus récemment, apparemment sans en vérifier spécifiquement les valeurs efficaces[42], les réglages généralement reconnus comme *habituels*[43] par les chercheurs des domaines de la reconnaissance automatique de la parole et de la classification des sons environnementaux sont présentés au Tableau 2.4. Nous énumérons également les valeurs appliquées par défaut par les versions courantes des bibliothèques les plus populaires, sous Python, pour l'analyse automatique des signaux sonores[44] :

Tableau 2.4 : Paramétrages STFT et MFCC communs

Paramètre	ASR	ESC	Librosa	PyTorch (Torchaudio)	TensorFlow
Fréquence d'échantillonnage (kHz)	16	22,05 - 44,1	22,05	16	Non spécifié (format de l'input)
Taille de la fenêtre FFT (points)	512	1024 - 2048	2048	400	2048
Longueur du saut (points)	160	512	Aucun	400 (Taille FFT)	512 (n_fft // 4)
Nombre de filtres mel	23 - 40	40 - 128	128	128	128
Nombre de coefficients	12 - 13	20 - 40	20	40	128 (N mel)

[42] Nous n'avons pu trouver de recherches vérifiant spécifiquement l'effet de ces paramètres sur l'apprentissage automatique. Il semble que ces études reprennent les réglages des travaux antérieurs afin de poursuivre le développement des algorithmes en aval ou en parallèle des MFCC.

[43] Nous retrouvons ces réglages dans diverses publications procédurales, non documentées, sur le Web.

[44] Il nous semble probable que les valeurs suggérées par les algorithmes des outils communs soient, de fait, celles qui sont utilisées dans les études publiées sans mention particulière des paramètres choisis.

2.1.2.7.4 Dimensions de la matrice

Les dimensions de la matrice MFCC dépendent de deux facteurs : le nombre de coefficients cepstraux (1re dimension, *x*) et le nombre de tranches temporelles (2e dimension, *y*).

1re dimension

Les MFCC permettent une très grande réduction de la première dimension depuis le spectrogramme FFT. Dans l'exemple de la Figure 2.2, ils permettent de passer de 1025 à seulement 40 valeurs sur cette dimension, voire 39 si C_0 est ignoré. À cela, nous ajoutons 40 (39) valeurs dans le cas de l'ajout des delta, et 40 (39) supplémentaires si les delta-delta sont également extraits, pour un total d'une dimension x = 120 pour un signal sous 40 coefficients cepstraux + delta et delta-delta.

2e dimension

Le nombre de tranches temporelles constitue la 2e dimension. L'extraction des MFCC est indépendante de ce paramètre puisqu'elle est appliquée à chacune des tranches temporelles du fenêtrage FFT, lequel dépend de la durée de l'extrait sonore et de sa fréquence d'échantillonnage, ainsi que des réglages de dimension et chevauchement de la fenêtre FFT. Ainsi, par exemple, le signal représenté par la Figure 2.2 comporte, à strictement parler, 83 tranches. Le calcul est le suivant (Oppenheim & Schafer, 2009) :

Équation 2-3 : Nombre de tranches temporelles en FFT

$$num_{frames} = \left\lceil \frac{N - n_{fft}}{hop_{length}} \right\rceil + 1$$

où :

- num_{frames} = nombre de tranches temporelles
- N = nombres d'échantillons (durée x f.é.) [Ici : 2 sec x 22 050 Hz = 44 100]
- n_{fft} = nombres de points de la fenêtre FFT [ici : 2048 points]

- hop_{len} = chevauchement FFT [ici : 512 points]

Selon cette application stricte, cependant, seules les tranches complètes de la fenêtre FFT sont considérées, ce qui peut laisser des segments partiels du signal non analysés, surtout aux extrémités de l'extrait. Il est donc courant d'utiliser un remplissage par zéros[45] afin d'ajouter des valeurs supplémentaires au signal ; ceci permettra d'inclure aux calculs ces segments partiels, évitant les artefacts de bordure (Lyons, 1997).

Dans le cas des bibliothèques de traitement du signal mentionnées plus haut, le calcul est le suivant (McFee et al., 2015b) :

Équation 2-4 : Nombre de tranches temporelles avec padding

$$num_{frames} = \left\lceil \frac{N + hop_{length} - 1}{hop_{length}} \right\rceil$$

Le résultat, dans le cas de notre exemple, est de 87 tranches, comme les graphiques présentés en font foi.

2.1.2.7.5 Résolutions temporelle vs fréquentielle dans le cas des STFT pour les MFCC

Lors de l'extraction des STFT préalable aux MFCC il est essentiel de trouver un compromis entre la résolution temporelle et la résolution fréquentielle. En effet, une fréquence d'échantillonnage élevée permet de capturer des détails temporels fins, cependant que, pour un n_fft donné, une augmentation de la fréquence d'échantillonnage réduit la durée de chaque fenêtre d'analyse, ce qui augmente le pas fréquentiel et diminue donc la résolution fréquentielle. En effet, le pas fréquentiel est inversement proportionnel à la durée de la fenêtre d'analyse : plus la fenêtre est courte, plus le pas fréquentiel est grand, et moins la résolution fréquentielle est précise.

[45] *Padding* ou *zero-padding*, en anglais, sont courants.

Il s'agit d'une notion fondamentale pour toutes les applications des FFT. En contexte d'extraction des MFCC, cependant, ces paramètres doivent être ajustés afin d'assurer leur concordance avec la largeur minimale des filtres mel. On voudra en effet assurer que la résolution fréquentielle de la STFT soit égale ou plus fine que la largeur de bande mel la plus étroite, ceci afin d'éviter la redondance dans les données MFCC.

À titre d'exemple, le réglage typique de longueur de fenêtre STFT, en ASR, est entre 10 et 40 ms (Jamal et al., 2017), ce qui permet de capter les changements suffisamment rapidement pour qualifier les consonnes, tout en permettant une résolution fréquentielle suffisante pour identifier les formants des voyelles (O'shaughnessy, 1987). Le Tableau 2.5 compare les valeurs de résolution de réglages STFT en lien avec ceux des bandes mel. On y constate qu'une n_FFT de 512 points, sous f.é. = 16 kHz, présente une résolution temporelle suffisante pour l'ASR (32 ms), et que la résolution fréquentielle (31,2 Hz) est plus fine que l'analyse MFCC sous 29 filtres mel (min 67,76 Hz), ce qui est correct. Cependant, si on cherche à augmenter la représentation fréquentielle, par exemple à l'aide d'une analyse MFCC resserrée à 128 filtres mel mais sous les mêmes réglages STFT, on verra tout le registre ≨ 900 Hz est représenté par des bandes mel plus fines (min 14,32 Hz) que la résolution fréquentielle de la STFT, ce qui produira des données redondantes et diminuera la validité de l'analyse, ou, à tout le moins, son efficacité.

On cherchera, afin d'optimiser le système, à ce qu'il y ait une concordance approximative entre la résolution fréquentielle de la STFT et la bande mel la plus étroite. Selon les données du tableau, on constate que les réglages 1024 points FFT et 128 bandes mel répondent à cette condition mais en privilégiant la résolution fréquentielle, alors que 512 points et 64 bandes mel présente un meilleur compromis afin d'améliorer la résolution temporelle sans surcharger le système inutilement.

Tableau 2.5 : Concordance des réglages STFT et mel

Fréquence d'échantillonnage STFT :	16 kHz		
Fenêtre FFT :	512 points	1024 points	2048 points
Résolution temporelle STFT :	32 ms	64 ms	128 ms
Résolution fréquentielle STFT :	31,2 Hz	15,6 Hz	7,8 Hz

Nombre de filtres mel :	29		128		64	
N° du filtre mel	*f* centrale (Hz)	Largeur de bande (Hz)	*f* centrale (Hz)	Largeur de bande (Hz)	*f* centrale (Hz)	Largeur de bande (Hz)
0	82,36	67,76	34,04	14,32	48,14	29,24
…						
27	6668,40	638,16	536,59	24,12	1406,23	82,31
…						
40	—	—	889,58	31,00	2766,91	135,49
…						
62	—	—	1731,32	47,42	7357,89	314,90
…						
126	—	—	7670,30	163,26	—	—

En résumé, on pourra être tenté d'augmenter les valeurs de résolution afin de tirer profit des processeurs modernes plus puissants, mais cette pratique pourra créer des systèmes peu efficaces. Dans tous les cas, cependant, on devra choisir de privilégier la résolution temporelle ou fréquentielle, selon le domaine d'application, ceci en assurant la concordance entre les paramètres STFT et MFCC afin d'optimiser la charge computationnelle.

2.1.3 Réseaux de neurones artificiels pour l'analyse automatique des signaux sonores

Les parties précédentes ont présenté les fondements théoriques du traitement des signaux sonores. Nous avons examiné la préparation des fichiers sonores et nous nous sommes attardés sur les MFCC en tant que technique d'extraction de caractéristiques. Ces étapes

sont cruciales pour transformer les signaux sonores bruts en représentations analytiques pertinentes et exploitables par des algorithmes d'apprentissage automatique.

Nous nous concentrons maintenant sur les réseaux de neurones artificiels (ANN[46]), une famille d'algorithmes d'apprentissage automatique particulièrement performants pour l'analyse automatique des signaux sonores. Les ANN, et plus spécifiquement les réseaux neuronaux convolutifs (CNN[47]), offrent des capacités remarquables pour la reconnaissance et la classification des motifs dans les données sonores. En intégrant ces techniques, nous croyons pouvoir améliorer la précision et l'efficacité de l'identification des sources sonores, contribuant ainsi à une gestion plus efficace du bruit environnemental.

Nous aborderons d'abord les concepts fondamentaux des ANN, leur structure et leur fonctionnement, avant de nous diriger vers des architectures plus sophistiquées comme les réseaux neuronaux convolutifs siamois avec fonction de perte par triplet. Cette progression permettra de comprendre la façon dont chaque composant et technique s'imbriquent afin de former un système efficace pour l'analyse des signaux sonores. Par cette exploration, nous chercherons à justifier l'utilisation des CNN siamois avec fonction de perte par triplet comme une solution possible et fonctionnelle pour identifier les sources sonores principales.

2.1.3.1 Introduction aux réseaux de neurones artificiels (ANN)

Les ANN ont été inspirés par le fonctionnement des réseaux de neurones biologiques. L'idée de base est de créer un système capable d'apprendre et de prendre des décisions de manière similaire à un cerveau humain. Le concept des ANN a été présenté pour la première fois en 1943 par McCulloch et Pitts, qui ont proposé un modèle mathématique

---

[46] *Artificial Neural Network*

[47] *Convolutional Neural Network*

des neurones. Depuis lors, les ANN ont évolué et se sont complexifiés, permettant des avancées significatives dans divers domaines, y compris l'analyse des signaux sonores.

Le développement des ANN a traversé plusieurs phases, commençant par des modèles simples tels que le perceptron, proposé par Rosenblatt en 1958. Le perceptron est un algorithme de classification binaire qui prend une entrée et la transforme par une fonction d'activation pour produire une sortie. Cependant, les perceptrons ont des limitations, notamment leur incapacité à résoudre des problèmes non linéaires, comme démontré par Minsky et Papert (1969).

Pour surmonter ces limitations, les chercheurs ont développé des architectures plus complexes constituées de multiples couches (LeCun et al., 2015; Rumelhart et al., 1986). Ainsi, un ANN est constitué de neurones artificiels organisés en couches : une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Chaque neurone dans ces couches reçoit des signaux d'entrée, les pondère, et passe le résultat à travers une fonction d'activation afin d'introduire de la non-linéarité dans le modèle et produire une sortie. Ce processus, appelé propagation de l'information, se fait de la couche d'entrée vers la couche de sortie. Les connexions entre les neurones sont représentées par des poids, qui sont ajustés durant l'apprentissage du réseau.

L'apprentissage, ou entraînement, ajuste ces poids pour minimiser l'erreur entre les sorties prévues et les sorties réelles, améliorant ainsi les performances du réseau sur de nouvelles données. Cet apprentissage se fait généralement par rétropropagation de l'erreur, une méthode introduite par Rumelhart et al. (1986). Cette rétropropagation permet au réseau d'apprendre à partir des données d'entraînement et d'améliorer ses performances sur de nouvelles données.

2.1.3.2 Types de réseaux de neurones

2.1.3.2.1 Réseaux neuronaux simples

Le perceptron à une seule couche (SLP[48]) est le modèle le plus élémentaire de réseau de neurones. Ce modèle se compose d'une seule couche de neurones connectée directement aux neurones d'entrée (Rosenblatt, 1958). Dans un SLP, chaque neurone effectue une somme pondérée des entrées et applique une fonction d'activation pour produire une sortie binaire. Ce type de réseau est capable de résoudre des problèmes de classification linéairement séparables (**Erreur ! Source du renvoi introuvable.**, A). Cependant, les SLP ne peuvent pas résoudre des problèmes non linéaires, limitant ainsi leur applicabilité à des tâches plus complexes (**Erreur ! Source du renvoi introuvable.**, B).

2.1.3.2.2 Réseaux neuronaux multicouches

Pour surmonter les limitations des SLP, les chercheurs ont développé les réseaux neuronaux multicouches (MLP)[49]. Un MLP est composé de plusieurs couches : une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Chaque neurone dans ces couches cachées applique une fonction d'activation non linéaire permettant au réseau de modéliser des relations complexes et non linéaires dans les données (Rumelhart et al., 1986). Le processus d'apprentissage des MLP utilise l'algorithme de rétropropagation de l'erreur, ce qui rend ces réseaux adaptés à des tâches telles que l'analyse des signaux sonores (Deng & Li, 2013).

Les MLP présentent plusieurs avantages notables pour l'analyse des signaux sonores. En effet, leur capacité à modéliser des relations non linéaires permet de capturer des caractéristiques complexes des signaux sonores (Bengio et al., 2013). De plus, ils peuvent être entraînés sur de grands ensembles de données, ce qui améliore leur robustesse et leur

---

[48] *Single-Layer Perceptron*

[49] *Multi-Layer Perceptron*

précision (Deng & Li, 2013). Enfin, l'utilisation de multiples couches cachées permet de créer des représentations hiérarchiques des données sonores, facilitant ainsi l'extraction de caractéristiques pertinentes.

Cependant, les MLP ont aussi des limitations. Leur architecture dense et entièrement connectée peut entraîner un nombre élevé de paramètres, nécessitant des ressources computationnelles importantes pour l'entraînement et l'inférence (LeCun et al., 2015). De plus, les MLP peuvent être sujets à des problèmes de surapprentissage (overfitting) si les données d'entraînement ne sont pas suffisamment diversifiées ou si le modèle est trop complexe pour la tâche à accomplir (Srivastava et al., 2014). Enfin, malgré leur capacité à capturer des relations non linéaires, les MLP ne sont pas toujours optimisés pour les données structurées comme les signaux sonores, comme nous le verrons maintenant.

2.1.3.3 Réseaux neuronaux pour l'analyse des signaux sonores

Les limitations des MLP en matière de complexité, de surapprentissage et d'optimisation pour les données structurées, comme les signaux sonores, ont conduit à l'exploration d'architectures de réseaux neuronaux plus spécialisées. Voici un aperçu des types de réseaux neuronaux significatifs pour l'analyse automatique des signaux sonores :

2.1.3.3.1 Réseaux de Neurones Récurrents (RNN)

Un RNN typique est composé de plusieurs cellules récurrentes, où chaque cellule reçoit une entrée à un instant t, ainsi que l'état caché de la cellule précédente (h_{t-1}). La cellule récurrente combine ces deux informations pour produire une sortie (y_t) et un nouvel état caché (h_t), qui sera utilisé lors de l'étape suivante. Ce mécanisme permet au réseau de conserver une trace des informations passées et de les utiliser pour influencer les décisions futures. Ainsi, les RNN peuvent modéliser les dépendances temporelles dans les données sonores, ce qui les rend appropriés pour des applications telles que la reconnaissance vocale et la prédiction de séries temporelles (Graves & Schmidhuber, 2005).

Cependant, les RNN classiques présentent certaines limitations, telles que la difficulté de capturer des dépendances à long terme en raison du problème de la disparition du gradient (Bengio et al., 1994). Pour surmonter ces limitations, des variantes comme les LSTM et les GRU ont été développées, offrant une meilleure capacité de modélisation des dépendances temporelles longues (voir ci-dessous).

2.1.3.3.2 Réseau neuronal à mémoire long court terme (LSTM[50])

Les LSTM sont une variante des RNN conçue pour capturer les dépendances à long terme dans les données séquentielles. Ils surmontent le problème de disparition du gradient en utilisant des portes qui régulent le flux d'informations, et en maintenant un état de cellule séparé qui peut préserver l'information sur de longues séquences. Les LSTM sont donc efficaces pour les tâches comme la reconnaissance vocale continue et l'analyse musicale (Hochreiter & Schmidhuber, 1997). Ils sont cependant plus complexes que les RNN simples, ce qui les rend plus gourmands en ressources computationnelles.

2.1.3.3.3 Réseaux de Neurones Gated Recurrent Unit (GRU)

Les GRU sont une autre variante des RNN, conçue pour capturer les dépendances temporelles dans les données séquentielles. Ils simplifient la structure des LSTM en combinant certaines des portes, ce qui permet de réduire le nombre de paramètres tout en maintenant des performances comparables. Les GRU ont été présentés par une équipe dirigée par Yoshua Bengio, au MILA[51] (Cho et al., 2014), dans le cadre d'une étude appliquée en traduction automatique de l'anglais vers le français. Notons que le terme GRU n'est pas directement utilisé dans l'article en référence, mais il a été adopté par la communauté scientifique et est devenu courant par la suite.

---

[50] *Long Short-Term Memory*

[51] Montréal Institute for Learning Algorithms

2.1.3.3.4 Réseaux neuronaux convolutifs (CNN)

Les CNN, popularisés dans les années 1990 et 2000, sont particulièrement efficaces pour traiter des données structurées en grille, comme les images ou encore les signaux sonores sous forme de spectrogrammes ou de MFCC. Ils utilisent des couches convolutionnelles pour extraire des caractéristiques locales et des couches de sous-échantillonnage pour réduire la dimensionnalité, ce qui les rend très performants pour la classification et la reconnaissance de motifs complexes dans les données sonores (Purwins et al., 2019).

Comme les CNN sont le type de ANN que nous avons choisi dans le cadre de la présente recherche, nous nous y attardons plus en détail dans la partie suivante de ce mémoire (2.1.3.4).

2.1.3.3.5 Réseaux de Neurones Transformers

Les Transformers, développés par Google (Vaswani et al., 2017), sont à la base des modèles de langage de grande taille (LLM[52]). Contrairement aux RNN, LSTM et GRU, qui traitent les séquences de manière séquentielle et souffrent de limitations liées aux dépendances à long terme, les Transformers utilisent des mécanismes d'attention qui permettent de traiter les éléments d'une séquence en parallèle. En calculant une série de poids d'attention, les Transformers indiquent l'importance relative de chaque élément de la séquence d'entrée pour chaque élément de la séquence de sortie. De plus, le mécanisme d'attention multitêtes des Transformers, appliqué plusieurs fois indépendamment et combinant les résultats, permet au modèle de pondérer l'importance de différentes parties de la séquence d'entrée, capturant ainsi des dépendances complexes sur de longues séquences sans être limité par la position des éléments dans la séquence.

Les Transformers sont devenus la norme pour les tâches de NLP. Leur efficacité a été démontrée dans des applications de traduction automatique avec des modèles comme

---

[52] *Large Language Model*

BERT(Devlin, 2018), GPT (incluant des versions comme GPT-3) et XLNet (Yang, 2019), ainsi que dans la génération de texte, où GPT-3[53] a produit d'excellents résultats en matière de cohérence et de pertinence contextuelle (Topal et al., 2021).

Les Transformers ont été largement adoptés dans diverses tâches de classification, notamment la reconnaissance d'événements sonores. Par exemple, l'*Audio Spectrogram Transformer* (Gong et al., 2021) illustre leur efficacité dans ce domaine. De plus, les Transformers ont également démontré leur potentiel dans la classification d'images grâce au modèle *Vision Transformer* (ViT) (Dosovitskiy et al., 2020), qui segmente les images en fragments traités comme des séquences. Cette approche a été étendue à la classification audio, où le ViT a été appliqué aux spectrogrammes, révélant ainsi la polyvalence des Transformers pour traiter des données audio structurées sous forme d'images spectrales (Koutini et al., 2021). Cependant, ces travaux soulignent que les Transformers sont nettement plus gourmands en ressources computationnelles que les CNN, avec une charge multipliée par quatre lorsque la longueur de la séquence d'entrée est doublée. Les auteurs ont néanmoins proposé une architecture optimisée qui surpasse les CNN en performance pour diverses tâches de classification de signaux sonores.

2.1.3.4 Les réseaux neuronaux convolutifs : Approfondissement et applications

Comme nous le présentions brièvement en 2.1.3.3.4, les CNN sont une classe de réseaux de neurones particulièrement adaptés au traitement des données structurées, telles que les images et les signaux sonores. Dans cette partie, nous explorons la motivation derrière les CNN, leur structure, les principes de base des convolutions et du sous-échantillonnage[54], ainsi que leur architecture typique.

---

[53] À l'heure d'écrire ce mémoire, GPT-4o, une amélioration de GPT-3, est courant. Cependant, l'article en référence précède cette version.

[54] *Pooling*, en anglais, est usuel.

2.1.3.4.1 Introduction aux CNN : motivation et structure

Les CNN ont été initialement développés pour surmonter les limitations des réseaux neuronaux traditionnels dans le traitement des données structurées. Contrairement aux MLP, où chaque neurone d'une couche est connecté à tous les neurones de la couche suivante, les CNN utilisent des connexions locales et partagent les poids entre les neurones pour capter les motifs locaux. Ceci permet une réduction significative du nombre de paramètres nécessaires. Par exemple, pour une matrice MFCC de 32 x 32, la première couche fully connected d'un MLP pourrait nécessiter des millions de connexions, contre seulement 320 pour un CNN avec 32 filtres de taille 3 x 3. De plus, les CNN conservent la structure spatiale ou temporelle des données grâce à leurs filtres de convolution, tandis que les MLP traitent les données d'entrée comme des vecteurs 1D « aplatis », ce qui peut entraîner une perte d'information structurelle importante où les motifs locaux et les relations spatiales ou temporelles sont souvent perdus. En conséquence, les MLP ne tiennent pas compte de la structure inhérente des données, alors que l'architecture des CNN leur permet d'exploiter les structures spatiales ou temporelles d'une matrice (Krizhevsky et al., 2012; LeCun et al., 2015; LeCun et al., 1998).

2.1.3.4.2 Convolutions, activation et sous-échantillonnage : principes de base

Les couches convolutionnelles sont au cœur des CNN. Elles appliquent des filtres de convolution aux entrées pour extraire des caractéristiques locales. Chaque filtre est glissé sur l'entrée et effectue une opération de convolution, produisant une carte de caractéristiques qui met en évidence des motifs spécifiques tels que des bords ou des textures dans le cas des images, mais aussi les motifs représentés par les spectrographes ou encore les MFCC dans le cas des sons. La taille des filtres est réglée par le paramètre

noyau[55], et la distance par laquelle la fenêtre de convolution (ou le noyau) se déplace à chaque étape est réglé par le *pas*[56].

Cette opération est suivie par une fonction d'activation permettant d'introduire de la non-linéarité dans le modèle en transformant les valeurs linéaires des neurones en sorties non linéaires (Nair & Hinton, 2010). Il s'agit d'une étape essentielle, car les ANN sans fonctions d'activation seraient équivalents à un seul modèle linéaire, quelle que soit leur profondeur. Les principaux types de fonctions d'activation sont :

- Sigmoïde : Sortie entre 0 et 1, interprétable comme une probabilité.

Équation 2-5 : Sigmoïde

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

- Tangente hyperbolique (tanh) : Sortie entre -1 et 1

Équation 2-6 : Tanh

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

- ReLU (Rectified Linear Unit) : Si l'entrée est positive ou nulle ($x \geq 0$), la sortie est égale à l'entrée ($ReLU(x) = x$). Si l'entrée est négative ($x < 0$), la sortie est nulle ($ReLU(x) = 0$).

Équation 2-7 : ReLU

$$f(x) = \max(0, x)$$

ReLU est l'une des fonctions d'activation les plus utilisées dans les architectures de réseaux de neurones, par exemple dans les populaires VGG, ResNet, et Inception, en

---

[55] *Kernel*, en anglais, est plus courant

[56] *Stride*, en anglais, est plus courant

raison de sa simplicité et de son efficacité à briser la linéarité tout en étant une méthode computationnelle efficace.

Le sous-échantillonnage est une technique utilisée pour réduire la dimensionnalité des cartes de caractéristiques tout en conservant les valeurs les plus significatives, ce qui permet au réseau de conserver les caractéristiques importantes des motifs repérés même si leur position change légèrement ou s'ils subissent des déformations mineures. Les opérations de sous-échantillonnage les plus courantes sont le sous-échantillonnage maximal et le sous-échantillonnage moyen[57], qui prennent respectivement le maximum ou la moyenne des valeurs dans une région de la carte de caractéristiques (Scherer et al., 2010). À titre d'exemple, la Figure 2.3 schématise le max sous-échantillonnage en noyau = 2 x 2 et pas = 2 (aucun chevauchement). Chaque grille de 2 x 2 est réduite à sa valeur maximale, diminuant de 1/4 le nombre de valeurs.

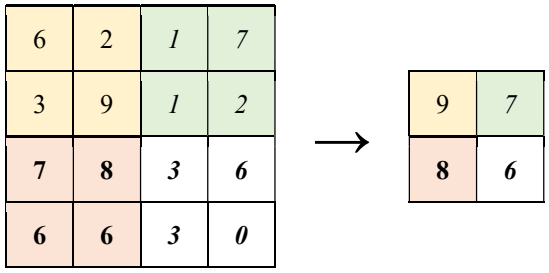


Figure 2.3 : Sous-échantillonnage

2.1.3.4.3 Architecture typique des CNN : couches convolutionnelles, couches de sous-échantillonnage, et couches fully connected

Une architecture typique de CNN (Figure 2.4) se compose de plusieurs couches convolutionnelles et de sous-échantillonnage, suivies de couches entièrement connectées, aussi appelées couches denses[58]. Les couches convolutionnelles et de sous-

[57] *Max-pooling* et *average-pooling*.

[58] *Fully connected*, en anglais, est également usuel.

échantillonnage sont souvent organisées en blocs, où plusieurs couches convolutionnelles sont suivies d'une couche de sous-échantillonnage. Ces blocs sont empilés pour créer des représentations hiérarchiques des données, avec des couches plus profondes capturant des caractéristiques de plus en plus abstraites (Krizhevsky et al., 2012).

Après les blocs de convolution et de sous-échantillonnage, les sorties sont aplaties[59] afin de les préparer, en les transformant en vecteurs unidimentionnels, à passer à travers une ou plusieurs couches entièrement connectées. Ces couches fonctionnent comme dans les MLP, où chaque neurone est connecté à tous les neurones de la couche précédente. Elles combinent les caractéristiques extraites par les couches précédentes pour produire la sortie finale, souvent via une couche softmax dans le cas des tâches de classification (LeCun et al., 2015)

[59] *Flattening*.

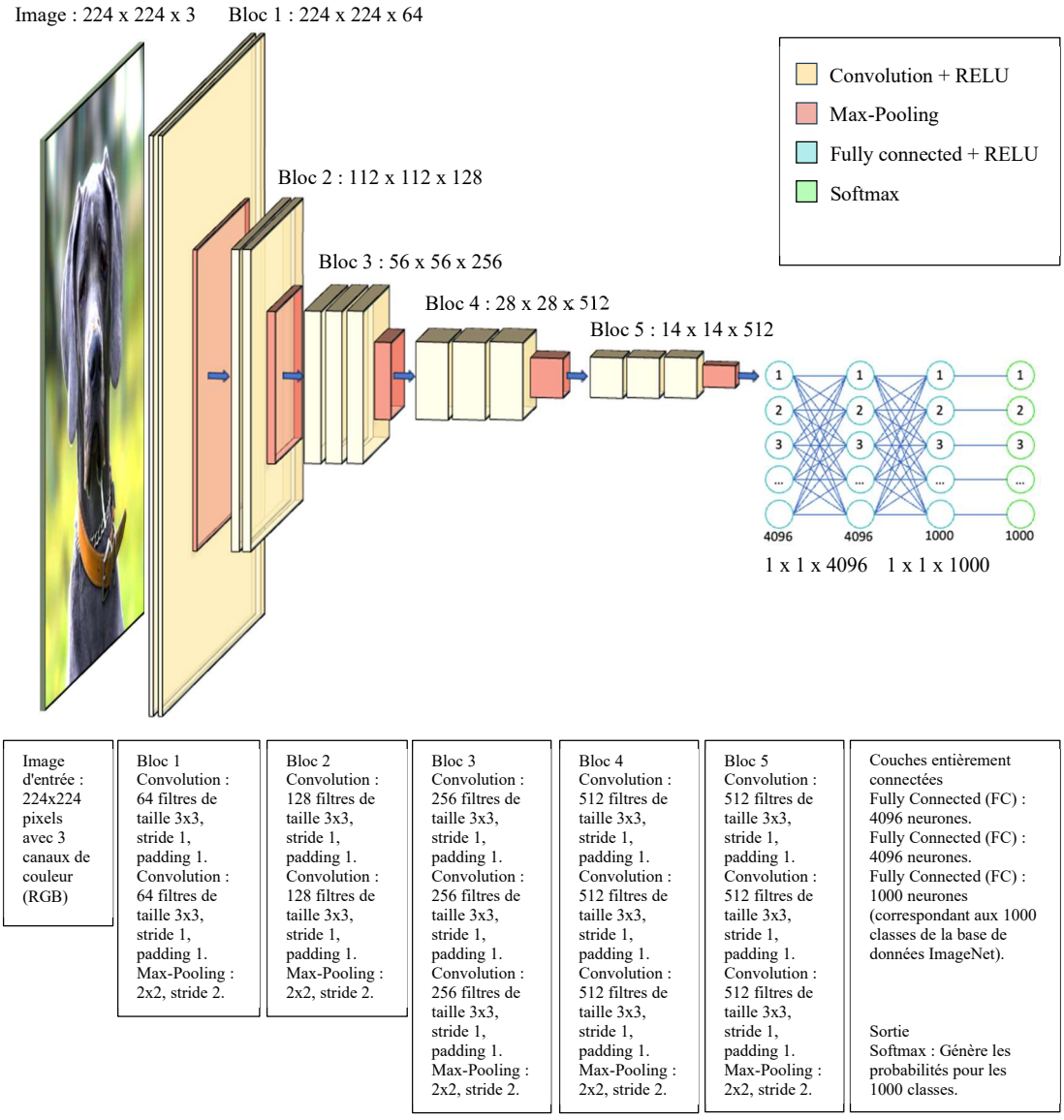


Figure 2.4 : Shématisation d'un CNN pour la classification des images. Ici, le modèle VGG-16 (Simonyan & Zisserman, 2014, in Mascarenhas & Agarwal, 2021).

Une couche softmax transforme les scores *logits*[60] d'entrée en probabilités normalisées, permettant d'interpréter la sortie du réseau comme des probabilités d'appartenance à différentes classes. Ces probabilités sont obtenues en divisant les valeurs après exponentiation par la somme de toutes les valeurs exponentielles des autres classes (Équation 2-8)(LeCun et al., 2015). Le résultat est un vecteur de probabilités dont la somme est égale à 1, ce qui facilite l'interprétation des résultats puisque la valeur de probabilité d'appartenance de l'entrée à une classe spécifique peut être directement lue en pourcentage.

Équation 2-8 : Softmax

$$softmax(z_i) = \frac{e^{z_i}}{\sum_j^C e^{z_j}}$$

2.1.3.5 Les CNN pour l'identification des sources sonores[61]

Les CNN ont initialement été développés pour la reconnaissance d'images et ont connu un grand succès dans ce domaine grâce à leur capacité à capturer les motifs dans les données visuelles. Des architectures de CNN comme LeNet-5 (LeCun et al., 1998), AlexNet (Krizhevsky et al., 2012), VGG (Simonyan & Zisserman, 2014), et ResNet (He et al., 2016) ont établi des niveaux de performance élevés en classification d'images, démontrant leur capacité à traiter des données visuelles complexes.

À l'aide de quelques adaptations spécifiques, les CNN ont par la suite été adaptés à l'analyse des signaux sonores. Les différences entre les applications de CNN pour les images et les sons résident dans la nature des données et les types de transformations nécessaires pour préparer ces données pour les réseaux convolutifs. Cependant, les données présentées au réseau après l'extraction de caractéristiques (voir 2.1.2) sont des

---

[60] Dans le domaine des ANN, les scores logits sont les vecteurs de scores non normalisés.

[61] Plusieurs des modèles répétoriés dans cette partie sont présentés plus en détail en 2.2.3.4.2

matrices bidimensionnelles (domaine temporel [I, T] ou domaine fréquentiel [f, T]) similaires à celles des images (O'shea & Nash, 2015). Toutefois, le design des CNN doit être ajusté pour les tâches spécifiques de détection des motifs sonores. Cela inclut le choix des tailles de filtres, des couches de pooling et des fonctions d'activation adaptées à la capture des caractéristiques sonores pertinentes (Hershey et al., 2017).

Voici quelques domaines d'application des CNN dans l'analyse des signaux sonores, avec quelques références de travaux phares : Reconnaissance de la parole (Amodei et al., 2016; Han et al., 2020; Sainath et al., 2015), analyse et détection d'émotions dans la voix (El Ayadi et al., 2011; Nigar, 2024), détection des sources nuisibles en environnement urbain (Bonet-Solà et al., 2023a, 2023b; Claudi Socoró et al., 2017; Ginovart-Panisello et al., 2021), identification des animaux (Salamon et al., 2017; Stowell & Plumbley, 2010; Vidaña-Vila, Navarro, Alsina-Pagès, et al., 2020), reconnaissance de la musique et des instruments (Ashraf et al., 2023; Pons et al., 2017; Van den Oord et al., 2013).

Nous nous attarderons en 2.2.3.4 aux travaux significatifs en identification automatique des sons qui utilisent des réseaux neuronaux profonds (DNN[62]), dont les CNN.

2.1.3.6 Réseaux neuronaux siamois (SNN)[63]

Les champs d'application présentés ci-dessus font appel à des techniques de classification des sons. En effet, il s'agit de filtrer les fichiers sonores afin de déterminer à quelle catégorie ils appartiennent, qu'il s'agisse d'un mot, d'un type de son urbain ou encore d'une émotion, par exemple.

Or, contrairement aux méthodes de classification traditionnelle, notre recherche ne vise pas à catégoriser les signaux sonores ; elle se concentre plutôt sur la comparaison de

---

[62] *Deep Neural Network*

[63] *Siamese Neural Networks*

différents signaux rapprochés à un signal complexe éloigné, afin d'y distinguer et identifier le constituant dominant. Dans cette partie, nous explorons les réseaux neuronaux siamois, une classe de réseaux de neurones artificiels particulièrement adaptés aux tâches de comparaison de signaux.

2.1.3.6.1 Concept et structure des réseaux neuronaux siamois

Les réseaux neuronaux siamois sont constitués de deux (ou plus) sous-réseaux jumeaux qui partagent les mêmes paramètres et poids. Cette architecture permet aux réseaux de traiter des entrées simultanément et d'apprendre à évaluer leur similarité. Les sorties des sous-réseaux sont ensuite comparées à l'aide d'une fonction de distance, telle que la distance euclidienne, pour déterminer la similitude entre les entrées.

Cette approche se montre particulièrement efficace dans des domaines tels que la reconnaissance de visages, la vérification de signatures ou encore l'identification de locuteurs, où l'objectif est de distinguer entre des entités similaires, mais distinctes. Bien que d'abord utilisés en contexte d'analyse d'images, les réseaux siamois sont notamment également utilisés dans les contextes d'analyse des signaux sonores suivants :

2.1.3.6.2 Reconnaissance et vérification du locuteur

Dans ces applications, les réseaux siamois comparent des segments de voix pour déterminer si deux segments proviennent du même locuteur. Par exemple, les travaux de Chen et Salman (2011) ont servi de point de départ pour l'utilisation des CNN siamois (SCNN[64]) pour plusieurs travaux ultérieurs. De plus, la détection d'attaques de relectures

[64] *Siamese Convolutional Neural Network*

audio[65] peut également faire appel à des SNN, comme l'ont démontré von Platen et al. (2020).

2.1.3.6.3 Identification d'événements sonores et apprentissage par peu d'exemples

Les réseaux siamois ont été utilisés pour l'identification et la classification d'événements sonores dans divers contextes, tels que les environnements urbains ou les habitats naturels. Par exemple, Zhong et al. (2021) ont utilisé un SNN avec le modèle CNN *DenseNet-201* (Huang et al., 2017) et une fonction de perte par triplet pour détecter, classifier et compter les appels des baleines bleues. Les auteurs ont démontré que les réseaux siamois surpassent les CNN traditionnels pour cette tâche spécifique. Ils mentionnent notamment le fait que les SNN peuvent être entraînés à partir de très peu de données, ce qui est également proposé par plusieurs auteurs dans le domaine de la détection d'événements bioacoustiques, en utilisant l'*apprentissage par peu d'exemples* (few-shot learning) (Droghini et al., 2018; Fedele et al., 2022). En effet, selon ces auteurs, les SNN, grâce à l'entraînement par comparaison, sont une solution intéressante dans les cas où peu de données sont disponibles pour l'entraînement du modèle. Ils sont donc mieux en mesure de s'adapter à des contextes non prévus et présentent davantage de robustesse aux variations intraclasses et interclasses. Dans le même ordre d'idée et dans le domaine de la reconnaissance faciale, Chopra et al. (2005) écrivent :

> The method can be used for recognition or verification applications where the number of categories is very large and not known during training, and where the number of training samples for a single category is very small.

[65] Les attaques de relecture sont des types d'attaques où un enregistrement audio d'une voix légitime est capturé puis rejoué pour tromper un système de reconnaissance vocale ou de vérification vocale.

2.1.3.7 Couches partagées, distance et fonction de perte

2.1.3.7.1 Couches partagées

Dans un SNN, les couches convolutionnelles des sous-réseaux jumeaux sont identiques en matière de structure et de paramètres. Cette conception partagée signifie que lorsque l'un des sous-réseaux est mis à jour au cours de l'apprentissage, les jumeaux le sont également de manière identique. Cette stratégie améliore l'efficacité de l'apprentissage et garantit que les transformations appliquées aux différentes entrées sont cohérentes et comparables (Bromley et al., 1993).

2.1.3.7.2 Fonction de distance

Après le passage des entrées à travers les sous-réseaux jumeaux, les sorties résultantes sont comparées à l'aide d'une fonction de distance. La distance euclidienne est couramment utilisée pour cette comparaison. La distance euclidienne entre deux vecteurs u et v est définie comme :

Équation 2-9 : Distance euclidienne

$$d(u, v) = \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}$$

Cette formule calcule la racine carrée de la somme des carrés des différences entre les coordonnées correspondantes des deux points. La distance euclidienne est largement utilisée en raison de sa simplicité et de son interprétation géométrique intuitive.

2.1.3.7.3 Fonction de perte

La fonction de perte utilise les distances calculées par la fonction décrite ci-dessus pour ajuster les poids des réseaux afin de minimiser la dissemblance entre les exemples similaires et maximiser la dissemblance entre les exemples différents. Voici deux fonctions de perte couramment utilisées :

2.1.3.7.3.1 Fonction de perte par contraste :

Équation 2-10 : Fonction de perte par contraste (Contrastive Loss)

$$\mathcal{L} = \frac{1}{2} y d(u,v)^2 + \frac{1}{2}(1-y) max\big(0, m - d(u,v)\big)^2$$

où $d(u,v)$ est la distance euclidienne entre les vecteurs u et v, et m est une marge qui sépare les paires dissemblables. La fonction de perte par contraste est définie pour une paire d'exemples (x_1, x_2) et leur label $y \in \{0,1\}$ indiquant si les deux exemples appartiennent à la même classe $(y = 1)$ ou à des classes différentes $(y = 0)$ (Chopra et al., 2005).

2.1.3.7.3.2 Fonction de perte par triplet :

Équation 2-11 : Fonction de perte par triplet (Triplet Loss)

$$\mathcal{L} = max(0, d(a,p) - d(a,n) + \alpha)$$

où d est la distance euclidienne et α est une marge qui vise à garantir que l'ancrage est plus proche du positif que du négatif d'au moins α. La fonction de perte par triplet est utilisée pour comparer un ancrage a avec un exemple positif p (de la même classe) et un exemple négatif n (d'une classe différente) (Schroff et al., 2015)[66].

2.1.3.8 Réseaux neuronaux convolutifs siamois avec fonction de perte par triplet

La Figure 2.5 présente l'intégration des éléments décrits précédemment dans le cas de l'analyse des données sonores à l'aide d'un réseau neuronal convolutif siamois avec fonction de perte par triplet.

[66] Nous détaillons la présentation de la fonction de perte par triplet, avec la version utilisée dans le cadre de notre étude, à la partie 0.

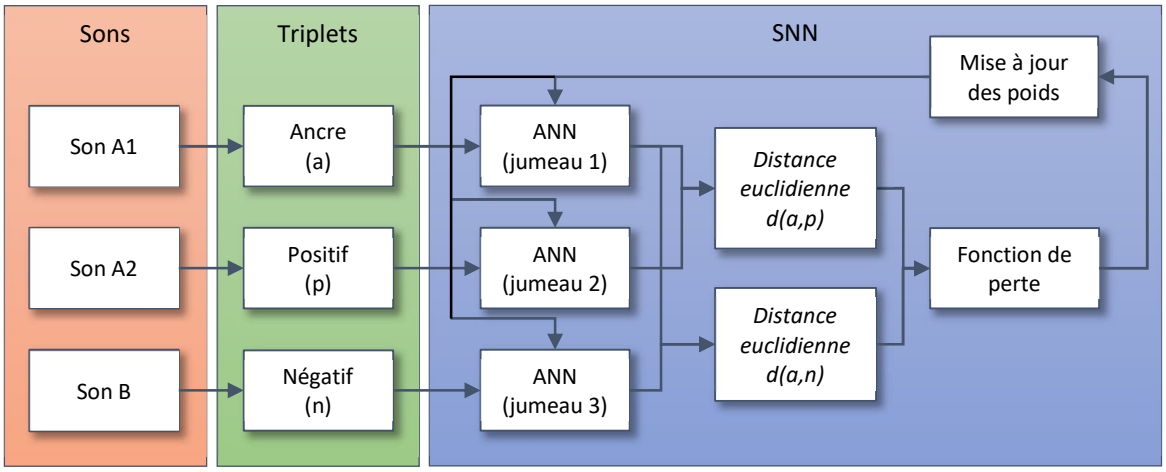


Figure 2.5 : Principe des SNN avec fonction de perte par triplet pour l'analyse des sons

2.1.3.8.1 Principe général

L'entraînement d'un SNN avec une fonction de perte par triplet repose sur la comparaison de triplets d'exemples (ancre, positif, négatif). L'objectif est d'apprendre une représentation des données qui minimise la distance entre des exemples similaires (ancre et positif) tout en maximisant la distance entre des exemples dissemblables (ancre et négatif).

2.1.3.8.2 Étapes de l'entraînement

1- Construction des triplets :

- Ancre (a) : Exemple de référence.
- Positif (p) : Exemple de la même classe que l'ancre.
- Négatif (n) : Exemple d'une classe différente de celle de l'ancre.

2- Passage des entrées dans les sous-réseaux :

- Chaque triplet est passé, simultanément, à un des 3 sous-réseaux jumeaux du SNN.
- Chaque sous-réseau extrait un vecteur de caractéristiques à partir de ses entrées respectives (ancre, positif, négatif).

3- Calcul des distances :

- La distance euclidienne entre les vecteurs de caractéristiques de l'ancre et du positif $[d(a,p)]$.
- La distance euclidienne entre les vecteurs de caractéristiques de l'ancre et du négatif $[d(a,n)]$.

4- Calcul de la fonction de perte par triplet :

- Comparaison des distances euclidiennes entre un ancrage (a) et un positif (p) avec celles entre l'ancrage (a) et un négatif (n), en s'assurant que la distance entre l'ancrage et le positif est inférieure à celle entre l'ancrage et le négatif d'au moins une marge α.
- α est un hyperparamètre qui détermine le degré de séparation souhaité entre les exemples similaires et dissemblables.

5- Mise à jour des poids :

- Les gradients de la fonction de perte sont calculés.
- Les poids des sous-réseaux jumeaux sont mis à jour pour minimiser la perte.

6- Répétition du processus :

- Le processus est répété sur plusieurs itérations avec différents triplets pour affiner les représentations des données.

2.2 Section 2 : Travaux antérieurs en identification des sources sonores nuisibles

Dans cette section, nous passerons en revue les travaux antérieurs qui ont contribué à l'identification des sources sonores nuisibles. Cette analyse rétrospective est essentielle pour situer notre recherche dans le contexte plus large des études existantes et pour identifier les lacunes et les opportunités pour des contributions nouvelles. Nous examinerons les approches méthodologiques, les technologies employées, les résultats obtenus, et les limites des études précédentes. Ces éléments nous permettront de justifier les choix méthodologiques de notre recherche, ainsi que de démontrer de quelle façon elle s'inscrit dans le continuum des efforts pour améliorer la gestion du bruit environnemental.

2.2.1 Historique de l'identification des sources sonores avant l'apprentissage automatique

2.2.1.1 Premières études et approches traditionnelles

Les premières tentatives d'identification des sources sonores nuisibles se sont principalement concentrées sur des méthodes classiques d'analyse de signaux, sur des techniques de traitement du signal ainsi que sur des algorithmes basés sur des modèles physiques, dont voici deux des principales méthodes, soit l'analyse spectrale et la fonction de transfert.

2.2.1.1.1 Transformée de Fourier et analyse spectrale

Comme nous l'avons brièvement mentionné précédemment, la transformée de Fourier est une méthode mathématique qui transforme un signal temporel en un signal fréquentiel, permettant d'analyser le contenu fréquentiel d'un signal sonore. Cette technique a été utilisée dans les premières études pour l'analyse des bruits environnementaux et industriels en identifiant les sources à l'aide de leur signature fréquentielle. Quelques exemples d'applications :

- Pollution sonore : Un exemple de travail dans ce domaine est celui de Moukas et al. (1982) sur la pollution sonore par les avions et les hélicoptères. Le système fait appel aux analyses spectrale et temporelle, combinées au codage prédictif linéaire, pour différencier les sources sonores dans divers environnements. La méthode a démontré une très bonne précision de reconnaissance pour différencier des sources sonores structurellement dissemblables, comme les avions à voilure fixe et les hélicoptères. Cependant, la précision peut être affectée par les conditions acoustiques variables et les bruits de fond imprévus. En outre, les auteurs suggèrent que des travaux supplémentaires sont requis afin d'identifier des sources de même source structurelle.
- Reconnaissance de signatures sonores de véhicules[67] : Wu et al. (1998) ainsi que Wieczorkowska et al. (2018) ont utilisé l'analyse spectrale et l'analyse en

[67] Voir Xiong (2019) pour une revue des principales techniques pour l'identfication des sources sonores d'un véhicule.

composantes principales (PCA[68]) pour la reconnaissance des véhicules. Leurs techniques traitent les vecteurs de fréquence des sons des véhicules comme des vecteurs dans un espace de caractéristiques de haute dimension, permettant l'identification des types de véhicules en fonction de leurs signatures sonores. Les performances de ces techniques se sont cependant montrées limitées par les bruits de fond et les interférences acoustiques, la précision des résultats dépendant de la qualité des données de formation et de l'environnement d'application. Plus récemment, une autre étude notable dans ce domaine est celle de Wang (2022) sur la reconnaissance et la détection des signaux de bruit et de vibration des véhicules. Wang a utilisé un algorithme LMS[69] à taille de pas variable pour améliorer la précision de l'identification des sources sonores dans des environnements complexes. Les résultats montrent que l'algorithme propose une bonne performance de détection et de séparation des signaux, avec une amélioration significative du taux de reconnaissance comparativement aux méthodes précédentes. Cependant, la performance de l'algorithme peut être influencée par les variations des conditions acoustiques, nécessitant une adaptation spécifique aux différents contextes d'application.

- Qualité des environnements urbains : Defréville (2005) a utilisé une approche combinant des méthodes physiques et perceptives pour caractériser et identifier les sources sonores dans un environnement urbain. L'analyse du signal est effectuée par le système Extractor Discovery System (EDS), développé par Sony CSL Paris, qui est conçu pour automatiser le processus de découverte de caractéristiques pertinentes dans les signaux audio. Cet effort de collaboration entre perception et mesure s'est montré limité par la subjectivité des perceptions humaines ainsi que par la complexité des environnements sonores urbains.
- Diagnostic de machines et équipements : Dans le domaine industriel, l'analyse spectrale est utilisée pour diagnostiquer les machines et équipements en détectant les anomalies fréquentielles qui peuvent indiquer des défauts ou des pannes imminentes. Les travaux de Sawalhi et Randall (2004), par exemple, ont présenté de bons résultats en permettant une identification précoce des défauts potentiels. Cependant, la précision s'est avérée limitée dans des environnements très bruyants ou complexes, en plus de souvent nécessiter une calibration spécifique aux machines et aux contextes industriels particuliers.

---

[68] *Principal Component Analysis*

[69] L'algorithme LMS (*Least Mean Squares*) est une technique qui ajuste les coefficients d'un filtre numérique, à l'aide d'une méthode de descente de gradient, afin de minimiser l'erreur entre la sortie désirée et la sortie réelle du filtre.

2.2.1.1.2 Méthodes par Fonction de transfert

Les méthodes par fonction de transfert utilisent des modèles mathématiques pour caractériser la relation entre une source sonore et le champ sonore résultant. Ces techniques se sont montrées utiles pour identifier les sources sonores en mesurant la réponse du système à des excitations connues et en utilisant ces informations pour localiser les sources de bruit. Voici des exemples qui utilisent cette méthode, pertinents dans le cadre de notre recherche :

- Wang et Crocker (1983) ont utilisé des systèmes à entrées multiples afin d'identifier les sources de bruit, à l'aide de techniques de cohérence, dans des environnements clos. Principalement orientée vers l'industrie, la technique vérifie la fonction de transfert entre un point de contrôle et de multiples capteurs près des sources afin d'établir la contribution de chacune. La méthode a donné de bons résultats avec des sources peu cohérentes et donc une contamination modérée, mais échoue dans le cas de forte contamination. Précisons que le système est tributaire des distances impliquées, puisque la corrélation de phase doit pouvoir être établie. En outre, la méthode se montrait exigeante en matière de calcul, limitant son application en temps réel avec les appareils de l'époque.
- Nelson et Yoon (2000) ont développé une méthode basée sur des techniques d'inversion discrète à partir des mesures de fonctions de transfert. En mesurant la réponse du système à des excitations connues, ils ont pu localiser et identifier les sources de bruit dans divers environnements. Leur méthode, efficace en conditions de bruit stationnaire, a été appliquée avec succès dans des domaines tels que l'aérospatiale et l'automobile. Cependant, la précision de cette technique dépend fortement de la qualité des mesures et des modèles utilisés, ainsi que de la capacité à contrôler les conditions expérimentales pour minimiser les erreurs de mesure et les interférences.
- Une étude de Noël (2004) propose une méthode d'identification des sources sonores bruyantes en utilisant des fonctions de transfert temporelles. Cette approche permet d'identifier les sources de bruit de manière plus précise et rapide comparativement aux méthodes précédentes. La méthode, qui s'appuie sur les signatures temporelles spécifiques de chaque source, a démontré une capacité à isoler les sources de bruit même en présence de multiples sources concurrentes. Les limitations, cependant, incluent la dépendance à des conditions spécifiques de l'environnement industriel, comme la configuration des machines et des structures environnantes. De plus, la méthode peut être limitée par la complexité du traitement des données et la nécessité de calibrations précises.

2.2.1.2 Techniques spécifiques et avancées en identification des sources sonores

Les limites des méthodes telles que l'analyse spectrale et la fonction de transfert ont mis en évidence la nécessité de techniques plus sophistiquées pour l'identification des sources sonores. En effet, ces approches classiques sont limitées par leur sensibilité aux bruits de fond, leur dépendance à des calibrations spécifiques et leur difficulté à gérer des environnements acoustiquement complexes. Pour surmonter ces défis, des techniques spécifiques et avancées ont été développées, exploitant les progrès récents en matière de traitement du signal et d'acoustique. Ces méthodes offrent une précision accrue, une meilleure robustesse face aux interférences acoustiques et une capacité améliorée à localiser et identifier les sources sonores dans des environnements variés et dynamiques. Voici un aperçu des principales techniques présentées ou utilisées plus récemment sans toutefois faire appel à l'apprentissage automatique, que nous commenterons en 2.2.3 :

2.2.1.2.1 Focalisation par imagerie acoustique, et variantes

La focalisation par imagerie acoustique, ou beamforming, utilise des réseaux de microphones pour former des faisceaux directionnels en combinant les signaux reçus afin de maximiser la sensibilité dans une direction spécifique. Cette méthode permet l'identification de la direction d'arrivée des sources sonores et une atténuation des interférences provenant d'autres directions. Van Trees (2002) a décrit les principes fondamentaux de cette technique et ses applications variées, incluant la surveillance acoustique.

Les progrès récents dans la technologie des réseaux de microphones et l'efficacité des algorithmes d'imagerie acoustique ont considérablement amélioré la performance de cette technique. Hou et al. (2022) ont souligné que les avancées en matière d'algorithmes d'imagerie acoustique ont permis de surmonter certaines limitations de cette technique. Toutefois, des défis subsistent, notamment en matière de résolution spatiale, de traitement des sources de bruit corrélées et de réponse en fréquence. Pour répondre à ces défis et

améliorer les performances dans des contextes spécifiques, plusieurs variantes du beamforming ont été développées :

2.2.1.2.1.1 Identification des sources de bruit (NSI[70]) :

Ginn et Haddad (2012) définissent ainsi cette spécialisation du beamforming :

> Les techniques d'identification des sources de bruit (NSI) sont utilisées pour optimiser l'émission de bruit d'une large gamme de produits, y compris les véhicules, les appareils ménagers et les éoliennes. L'objectif des techniques NSI est d'identifier les sous-sources les plus importantes d'un objet en matière de position, de contenu fréquentiel et de puissance acoustique. Le classement des sous-sources peut ensuite servir à déterminer où des modifications de conception amélioreront le plus efficacement l'émission sonore globale.[71]

Le NSI utilise des réseaux de microphones et des algorithmes d'analyse avancée pour localiser et identifier les sources de bruit dans des environnements complexes. L'un des objectifs principaux est d'améliorer la résolution spatiale afin d'identifier précisément la position des sources dans des environnements comportant plusieurs sources de bruit ; elles nécessitent donc des approches robustes pour la séparation des sources et l'analyse de leur impact. Voici quelques références significatives en NSI :

- Pereira et al. (2012) comparent des méthodes théoriques et expérimentales pour l'identification des sources de bruit utilisant des réseaux de microphones et des algorithmes avancés, tels que l'*equivalent source method* (ESM) ainsi qu'une approche bayésienne. Cette comparaison met en avant l'efficacité de ces techniques dans des environnements complexes.
- Gade et al. (2013) précisent que, dans le cas de la technique de *formation de faisceau planaire par retard et somme (DAS)*[72] : « La résolution spatiale est proportionnelle à la distance entre le réseau et la source, et inversement

---

[70] *Noise Source Identification*

[71] Traduction depuis l'anglais par l'auteur.

[72] *Delay-and-sum planar beamforming*

proportionnelle à la longueur d'onde, avec comme conséquence que la résolution est bonne uniquement pour les fréquences moyennes à élevées[73] ». En réponse à cette limitation, les auteurs proposent des algorithmes améliorés utilisant des techniques de déconvolution itérative afin d'obtenir une résolution spatiale améliorée.

- Hou et al. (2022) publient une revue exhaustive des méthodes d'identification des sources de bruit basées sur des réseaux de microphones, offrant une vue d'ensemble actualisée des techniques avancées et de leurs applications pratiques.

2.2.1.2.1.2 Holographie acoustique :

L'holographie acoustique utilise des principes de la transformée de Fourier inverse pour reconstruire les champs sonores tridimensionnels à partir des mesures prises par des réseaux de microphones. Cette technique permet de visualiser la distribution spatiale des sources sonores et leur propagation dans un environnement donné. Les techniques d'holographie acoustique incluent des méthodes avancées de traitement du signal comme l'holographie acoustique en champ proche (NAH[74]). Cette méthode est largement utilisée dans l'aéroacoustique et l'analyse structurelle pour identifier les sources de vibrations et de bruit (Fernandez-Grande, 2022; Veronesi & Maynard, 1987). Mentionnons également les travaux de Salin et Kosteev (2020), qui propose le *Far Plane Series* comme méthode adaptant la NAH au champ radiant éloigné.

2.2.1.2.1.3 Fusion de données multimodales :

Cette approche combine des données provenant de divers capteurs, tels que des microphones et des caméras vidéos, pour améliorer la précision et l'interprétation des résultats. Essid et al. (2018) et Padois et al. (2018) ont montré que cette fusion peut améliorer la précision, bien qu'elle partage les mêmes défis de résolution spatiale et de traitement des sources corrélées.

---

[73] Traduit depuis l'anglais par l'auteur.

[74] *Near-field Acoustic Holography*

2.2.1.2.1.4 Méthodes CLEAN et CLEAN-SC :

CLEAN et CLEAN-SC sont des techniques avancées pour améliorer la résolution spatiale des cartes de sources sonores en acoustique. CLEAN a été initialement développée pour l'astronomie radio et adaptée à l'acoustique pour supprimer les lobes secondaires des cartes de faisceaux acoustiques. Elle opère dans le domaine temporel, ce qui permet de suivre les variations temporelles des sources en mouvement et de les quantifier précisément. Cousson et al. (2019), notamment, ont démontré l'efficacité de CLEAN pour localiser et quantifier les sources sonores mobiles.

CLEAN-SC (CLEAN based on Spatial Coherence), proposé par Sijtsma (2007), améliore CLEAN en intégrant la cohérence spatiale des signaux. Cette méthode gère mieux les sources corrélées et surmonte les limitations de CLEAN dans des environnements avec un S/B faible. CLEAN-SC est particulièrement utile en aéroacoustique pour améliorer la résolution et la précision des cartes de sources sonores.

Une avancée ultérieure, HR CLEAN-SC (High-Resolution CLEAN-SC), a été proposée par Sijtsma et al. (2017). Cette méthode améliore encore la résolution spatiale en utilisant des techniques de cohérence spatiale haute résolution, validées expérimentalement pour des applications aéroacoustiques.

Ces méthodes offrent des avantages significatifs, comme la capacité de suivre les sources mobiles ou encore améliorer la précision des cartes acoustiques. Toutefois, elles nécessitent des ressources computationnelles importantes et une instrumentation précise.

2.2.1.2.2 Séparation aveugle des sources audio (BASS) [75]

Comme la focalisation par imagerie acoustique et ses variantes, la BASS est une technique avancée en identification des sources sonores. Elle permet d'isoler des signaux sources

[75] *Blind Audio Source Separation*.

individuels à partir d'un ensemble de signaux mélangés, sans connaissance préalable des caractéristiques des sources ni de la manière dont elles sont mélangées, permettant d'identifier les sources sonores individuelles dans des environnements complexes.

Dans « *A Survey of Convolutive Blind Source Separation Methods*[76] », Pedersen et al. (2007) passent en revue plusieurs techniques de BASS, dont les plus significatives sont l'*analyse en composantes indépendantes* (ICA)[77], les *décompositions éparses* (SD)[78] et la CASA. Ces méthodes, bien que variées dans leur approche — qu'il s'agisse de l'exploitation de l'indépendance statistique, de la parcimonie des signaux ou de la modélisation de la perception auditive humaine —, sont toutes confrontées à des défis similaires, tels que la complexité computationnelle et la sensibilité aux conditions acoustiques, notamment dans des environnements réverbérants. Les auteurs soulignent l'efficacité de ces techniques dans des contextes spécifiques tout en notant leurs limitations dans des scénarios plus complexes.

2.2.1.2.3 Techniques de suivi d'objet acoustique

Les techniques de suivi d'objet acoustique reposent sur des algorithmes spécialisés qui traitent les signaux captés par des réseaux de microphones pour suivre des objets en mouvement dans des environnements complexes. Un ouvrage fondateur dans ce domaine est « *Target Tracking and Data Fusion* de Bar-Shalom et al. (2011). Ce livre présente les bases théoriques et les algorithmes pour le suivi des cibles en mouvement, y compris le filtrage de Kalman et les filtres particulaires.

Des travaux récents ont apporté des avancées significatives dans ce domaine. Par exemple, Zhang et al. (2023) ont étudié les défis pratiques du suivi acoustique en conditions réelles,

---

[76] Traduction par l'auteur : Une étude des méthodes de séparation aveugle des sources convolutives.

[77] *Independent Component Analysis*

[78] *Sparse Decompositions*

tels que la haute mobilité des cibles, le faible rapport signal-bruit (S/B) et la réponse en fréquence des appareils de mesure. Ils ont proposé des solutions pour améliorer la précision du suivi, notamment en compensant les décalages Doppler et en optimisant la réponse fréquentielle des capteurs.

Un autre travail notable est celui de Abu et al. (2024), qui présente un algorithme pour la détection et le suivi d'objets mobiles sous-marins en utilisant une approche de suivi avant détection. Leur méthode utilise un modèle de vitesse presque constante (NCV[79]) et un filtre de Kalman pour estimer la position et la vitesse des cibles, démontrant une continuité et une précision de suivi robustes dans des scénarios simulés et réels. Ils ont également surmonté le problème du taux élevé de fausses alarmes en utilisant un détecteur à taux constant de fausses alertes (CFAR[80]).

Ces techniques se sont montrées efficaces dans les domaines de la surveillance de la faune et la gestion du trafic. Cependant, la précision du suivi peut être affectée par la résolution spatiale des réseaux de microphones ainsi que par les interférences acoustiques. De plus, les algorithmes de suivi nécessitent des ressources computationnelles significatives, ce qui peut poser des défis en matière de puissance de calcul et de temps de traitement.

2.2.1.2.4 Modèle de Markov cachés (HMM[81])

Les HMM ont été largement utilisés, avant l'avènement des techniques d'apprentissage automatique modernes, pour l'identification des sons. Ces modèles statistiques, qui modélisent les transitions entre des états cachés d'un système à partir d'observations

---

[79] *Nearly Constant Velocity*

[80] *Constant False Alarm Rate*

[81] *Hidden Markov Model*

visibles, ont été particulièrement efficaces pour des applications telles que la reconnaissance vocale et la classification ou reconnaissance d'événements sonores.

Un ouvrage de référence est *Speech and Language Processing* de Jurafsky et Martin (2024), d'abord publié en 2000, qui détaille l'application des HMM dans le traitement de la parole, notamment pour modéliser les transitions phonétiques. Les HMM ont en effet permis des avancées significatives dans la reconnaissance vocale en améliorant la capacité à reconnaître des séquences de paroles dans divers environnements.

Dans le domaine de la classification des sons environnementaux, Couvreur et al. (1998) ont appliqué les HMM pour la classification automatique des événements sonores environnementaux. Leur étude a démontré que les HMM pouvaient atteindre un taux de classification élevé sur un ensemble de données comprenant divers sons environnementaux, montrant l'efficacité de ces modèles pour cette tâche spécifique. Cependant, ces résultats sont souvent limités par la capacité des HMM à modéliser des relations temporelles complexes sur de longues séquences, ce qui peut entraîner une performance dégradée lorsque les conditions acoustiques sont très variées ou lorsque les données sont bruitées.

Par ailleurs, l'étude de Geiger et al. (2014) a exploré une application novatrice des HMM pour l'identification des personnes à partir du son de leur démarche. Bien que leur approche se soit montrée efficace, l'identification des individus basée sur ces données a révélé plusieurs limitations. Par exemple, le modèle HMM utilisé dans cette étude était sensible aux variations dans les conditions d'enregistrement, telles que le bruit de fond et les changements de surface, ce qui a affecté la robustesse du système. De plus, les HMM ont montré des difficultés à généraliser efficacement à des personnes ou à des conditions qui n'étaient pas présentes dans l'ensemble d'entraînement, ce qui a limité leur précision dans des scénarios réels.

2.2.2 Limites et défis des approches traditionnelles

Les approches traditionnelles d'identification des sons, qui précèdent l'intégration des approches d'apprentissage automatique, bien qu'efficaces dans certains contextes, présentent plusieurs limitations et défis importants.

2.2.2.1 Robustesse et généralisation

Les méthodes traditionnelles, comme l'analyse spectrale à l'aide de la transformée de Fourier ou les techniques de fonction de transfert, souffrent souvent de limitations sur le plan de la robustesse et de la généralisation. En effet, ces approches sont généralement très sensibles aux variations environnementales telles que les changements de conditions météorologiques, la présence d'obstacles et les variations des sources sonores elles-mêmes. Ainsi, une méthode conçue pour fonctionner dans un environnement spécifique peut échouer à identifier correctement les sources sonores dans un environnement différent, rendant leur déploiement en conditions réelles problématique. Ces limitations peuvent être attribuées à la dépendance excessive aux caractéristiques statiques des signaux sonores et à une incapacité à s'adapter dynamiquement aux changements contextuels (Kopp & Wachsmuth, 2004; Rabiner & Juang, 1993).

2.2.2.2 Complexité des modèles et des données, et temps de traitement

Les approches traditionnelles nécessitent souvent des modèles complexes et une quantité considérable de données de haute qualité pour obtenir des résultats fiables, ce qui peut limiter leur utilisation pratique (Randall, 2021). De plus, ces méthodes exigent souvent une puissance de traitement considérable, ce qui peut entraîner des retards significatifs dans le traitement des données, rendant difficile l'application de ces méthodes pour des tâches nécessitant une analyse rapide ou en temps réel comme la surveillance environnementale ou l'identification des sources sonores dans des environnements dynamiques (Jekateryńczuk & Piotrowski, 2023; Zhuo & Cao, 2021).

De plus, la qualité des données utilisées est cruciale pour la performance de ces modèles puisque les données bruitées ou mal calibrées peuvent dégrader significativement les

résultats, rendant l'identification des sources sonores moins précise et moins fiable. Par exemple, Chen et al. (2002) soulignent que les techniques de beamforming sont particulièrement sensibles au bruit et aux interférences, ce qui peut compromettre la précision de la localisation des sources. De même, Brandstein et Ward (2013) discutent des défis associés à l'utilisation de réseaux de microphones dans des environnements réels où la qualité des données peut être affectée par divers facteurs.

2.2.2.3 Distances de détection

Les approches traditionnelles d'identification des sources sonores présentent des limitations significatives sur le plan des distances de détection. La précision de l'identification diminue souvent avec l'augmentation de la distance entre la source sonore et les capteurs. Par exemple, les techniques de beamforming et de localisation de sources sonores sont particulièrement sensibles à la puissance du signal qui diminue avec la distance, ce qui rend plus difficile la séparation du signal pertinent du bruit de fond (Chen et al., 2002). De plus, des facteurs environnementaux tels que la température, l'humidité, et la présence d'obstacles peuvent altérer la propagation des ondes sonores, ce qui complique davantage la détection précise à longue distance. Ces facteurs peuvent induire des réflexions, des diffractions et des interférences qui dégradent la qualité du signal reçu, réduisant ainsi l'efficacité des techniques traditionnelles dans des conditions réelles (Brandstein & Ward, 2013).

Un autre défi majeur réside dans la corrélation de phase entre les signaux captés par différents microphones. Dans le cas des relations entre 2 points de mesures distincts, par exemple pour la méthode expérimentée par Wang et Crocker (1983), l'augmentation de distance rend la corrélation impossible à établir, faisant échouer le système. Et dans le cas des antennes microphoniques, à mesure que la distance entre la source et les capteurs augmente, les différences de phase entre les signaux captés par différents microphones tendent à diminuer, ce qui peut compliquer l'estimation précise de la direction d'arrivée du signal et, par conséquent, la localisation de la source sonore (Brandstein & Ward, 2013). Ce problème est particulièrement pertinent dans les environnements où les

réflexions et les échos sont présents, car ces phénomènes peuvent introduire des déphasages supplémentaires qui compliquent encore plus la corrélation de phase.

Ces limitations mettent en lumière la nécessité d'explorer des approches alternatives plus robustes et adaptatives, capables de surmonter les défis associés à la variabilité environnementale, à la complexité des données, aux exigences de temps de traitement, et aux distances de détection.

2.2.3 Approches basées sur l'apprentissage automatique

2.2.3.1 Contexte et évolution

Les limitations des approches traditionnelles pour l'identification des sources sonores, comme discuté dans la section précédente, mettent en évidence la nécessité de solutions plus robustes et adaptatives. L'émergence des techniques d'apprentissage automatique s'inscrit dans cette recherche de méthodes capables de surmonter les défis liés à la variabilité environnementale, à la complexité des données, aux exigences de temps de traitement et aux distances de détection.

Dans cette perspective, des travaux comme ceux de Ellis (1996) ont marqué une étape clé. Bien que n'utilisant pas directement l'apprentissage automatique, Ellis, par exemple, ici dans le domaine de l'analyse computationnelle de scène auditive, a proposé une approche « prediction-driven » qui illustre la progression naturelle vers des méthodes plus avancées de modélisation et d'analyse des scènes auditives. Ce type de travail a préparé le terrain pour les développements futurs de l'analyse automatique des signaux sonores, en posant les bases conceptuelles des méthodes computationnelles qui seront ensuite enrichies par l'apprentissage automatique. De manière similaire, Choe et al. (1996), ici en classification automatique des sons, ont démontré la nécessité de techniques capables de gérer la variabilité des signaux acoustiques dans des conditions réelles, renforçant ainsi l'idée que l'apprentissage automatique représenterait une avancée nécessaire et logique dans le domaine.

2.2.3.2 Avantages de l'apprentissage automatique

Les techniques d'apprentissage automatique offrent des solutions prometteuses pour améliorer la robustesse, la généralisation et la performance des modèles d'identification des sources sonores dans des environnements variés. L'intégration de l'apprentissage automatique permet non seulement d'automatiser le processus d'identification, réduisant ainsi la dépendance aux méthodes manuelles et subjectives, mais aussi d'améliorer significativement la précision des systèmes de détection. Les modèles peuvent être entraînés sur de grandes quantités de données, capturant des schémas complexes qui échappent aux approches conventionnelles. De plus, l'apprentissage automatique permet une meilleure généralisation des modèles, leur permettant de s'adapter à des conditions environnementales variées, augmentant ainsi la fiabilité des systèmes dans des contextes réels et dynamiques (Hershey et al., 2017; LeCun et al., 2015; Schlüter & Grill, 2015).

2.2.3.3 Techniques d'apprentissage automatique sans réseaux de neurones artificiels

Avant l'avènement des ANN, plusieurs approches d'apprentissage automatique ont été développées pour améliorer l'identification des sources sonores. Ces méthodes, bien que ne reposant pas sur des architectures neuronales, ont permis des avancées significatives en comparaison aux méthodes traditionnelles. Parmi elles, des techniques comme les SVM (Wei et al., 2020) ou encore les méthodes bayésiennes (Godsill et al., 2007) ont été explorées pour extraire et classifier les caractéristiques sonores de manière plus flexible et adaptative.

Un exemple notable est le travail de Salamon et Bello (2015) dans leur article « Unsupervised Feature Learning for Urban Sound Classification[82] ». Leur approche utilise des techniques d'apprentissage non supervisé basées sur des méthodes de

[82] *Apprentissage non supervisé des caractéristiques pour la classification des sons urbains* (traduction par l'auteur)

clustering, en particulier le k-means, pour extraire automatiquement des caractéristiques pertinentes des sons urbains, sans recourir à un réseau de neurones artificiels. Ils démontrent ainsi que l'apprentissage automatique peut être appliqué avec succès pour améliorer la classification sonore dans des environnements complexes, tout en améliorant les performances depuis les méthodes traditionnelles.

Cependant, malgré les avancées apportées par ces méthodes, les techniques d'apprentissage automatique sans ANN peuvent présenter les mêmes limites que les modèles précédents, comme être restreintes dans leur capacité à capturer des relations complexes et non linéaires au sein des données sonores. Leur performance peut également être limitée par une capacité moindre à généraliser à de nouveaux contextes ou à s'adapter aux variations des conditions acoustiques. De plus, ces méthodes peuvent parfois nécessiter des étapes de prétraitement ou de sélection de caractéristiques qui dépendent fortement de l'expertise humaine, limitant ainsi leur automatisation complète.

Ces limitations ont conduit à l'exploration de modèles plus sophistiqués, tels que les ANN, qui offrent une capacité accrue à modéliser les complexités inhérentes aux signaux sonores, tout en s'adaptant de manière dynamique à une variété de conditions. C'est pourquoi la prochaine partie se concentre sur les approches ANN, qui représentent une évolution clé dans le domaine de l'identification des sources sonores.

2.2.3.4 Travaux exemplaires utilisant des ANN

Les études modernes explorent très majoritairement, pour la classification et la détection sonore, l'application des techniques de ANN. Cependant, à notre connaissance, aucune recherche à ce jour ne se concentre directement sur l'identification des sources sonores principales dans des environnements bruyants. Cette lacune dans la littérature présente une opportunité de recherche significative.

Ainsi, dans le but de recenser les travaux qui ont servi de base à notre étude, nous examinerons ici diverses applications des ANN dans le domaine de l'analyse sonore. Nous commencerons par explorer des études générales sur la *classification automatique des*

sons, en mettant l'accent sur les domaines spécifiques de la reconnaissance vocale de même que la récupération d'informations musicales. Nous terminerons par une revue des approches centrées sur la détection et la classification des événements sonores, en distinguant d'une part les méthodes de ESC, et d'autre part les techniques de séparation des sources sonores à l'aide des DNN.

2.2.3.4.1 Études générales sur la classification automatique des sons

2.2.3.4.1.1 Reconnaissance vocale et identification des locuteurs

La reconnaissance vocale est l'une des applications les plus développées de l'apprentissage automatique dans l'analyse sonore. Les premières approches basées sur les HMM ont été progressivement remplacées par des RNN et des DNN, qui ont démontré une amélioration significative dans la précision de la reconnaissance de la parole, même dans des environnements bruyants.

Parmi les travaux notables, l'étude de Graves et al. (2013) a montré que les RNN, combinés avec la technique de *Connectionist Temporal Classification* (CTC), permettent de reconnaître des séquences de paroles sans besoin de segmentation préalable. Sur le corpus TIMIT (Garofolo et al., 1993), leur modèle a atteint un taux d'erreur phonétique (PER)[83] d'environ 17,7 %, ce qui représente une avancée significative en comparaison aux méthodes traditionnelles. En effet, Hinton et al. (2012) rapporte des valeurs PER, sur le corpus TIMIT, aux environ de 27 % pour les modèles qui n'utilisent pas les ANN.

---

[83] PER (*Phone Error Rate*) : Mesure la précision d'un système de reconnaissance vocale à identifier correctement les phonèmes, qui sont les unités sonores de base dans une langue. Il est généralement utilisé pour évaluer les systèmes travaillant sur des tâches de reconnaissance de phonèmes plutôt que de mots entiers.

Une autre référence importante est l'article « *Convolutional, long short-term memory, fully connected deep neural networks*[84] », où Sainath et al. (2015) proposent une architecture combinant des CNN, des réseaux LSTM et des DNN, appelée CLDNN[85]. Dans cette approche, les CNN sont utilisés pour réduire la variation spectrale des caractéristiques d'entrée, les couches LSTM assurent la modélisation temporelle, et les DNN produisent une représentation des caractéristiques plus facilement séparables. Selon les auteurs, les résultats obtenus sur diverses tâches de reconnaissance vocale à grande échelle (LVCSR)[86], en contexte de recherche vocale, montrent que l'architecture CLDNN présente une réduction relative du WER[87] de 4 à 6 % comparativement à un modèle LSTM seul.

Un autre exemple est *Deep Speech 2* (Amodei et al., 2016), où des RNN bidirectionnels sont utilisés pour améliorer la précision de la reconnaissance. *Deep Speech 2* repose principalement sur des RNN et utilise des CNN en amont pour la préextraction des caractéristiques, ce qui permet au modèle de mieux capturer les informations locales avant l'étape de modélisation temporelle par les RNN. Les auteurs rapportent une performance

---

[84] Traduction par l'auteur : Réseaux de neurones profonds convolutionnels à mémoire long court terme entièrement connectés.

[85] *Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks* (réseaux de neurones profonds convolutifs, à mémoire long court terme et à couches entièrement connectées [traduction par l'auteur])

[86] LVCSR (*Large Vocabulary Continuous Speech Recognition*) : Désigne des systèmes de reconnaissance vocale capables de traiter des phrases complètes avec un large vocabulaire (potentiellement des centaines de milliers de mots). Ces systèmes sont utilisés dans des applications où l'utilisateur peut parler de manière naturelle et fluide, sans avoir à se limiter à un ensemble restreint de mots ou de phrases.

[87] WER (*Word Error Rate*) : Mesure la précision d'un système de reconnaissance vocale à transcrire correctement des mots. Le WER est une métrique standard dans la reconnaissance vocale pour évaluer la performance globale d'un système sur des tâches où l'objectif est de comprendre ou transcrire des phrases complètes.

WER de 5,15 % sur le corpus Switchboard et de 3,6 % sur le corpus Fisher[88], démontrant l'efficacité de cette architecture dans des environnements de reconnaissance vocale à grande échelle.

Enfin, *ContextNet* (Han et al., 2020) est une architecture CNN conçue pour la reconnaissance vocale de bout en bout, qui intègre des mécanismes d'attention pour améliorer la modélisation du contexte global dans les données audio. Cette approche s'est montrée efficace pour traiter des séquences longues et complexes. *ContextNet* a atteint une meilleure précision que les modèles précédents, avec un WER de 8,2 % sur un ensemble de données YouTube, surpassant ainsi les architectures combinant convolutions et LSTM bidirectionnels tout en utilisant moins de paramètres et de ressources de calcul.

En parallèle, l'identification des locuteurs a également bénéficié de l'apprentissage automatique. L'étude de Zhang et Koishida (2017), par exemple, a démontré l'efficacité de cette approche pour la vérification des locuteurs indépendamment du texte, même avec des segments de parole très courts. Leur modèle, basé sur une architecture SCNN couplée à une fonction de perte par triplet, a atteint un taux d'égalité des erreurs (EER[89]) de 2,97 % après un seul énoncé d'enrôlement, et jusqu'à 1,84 % avec 10 énoncés sur un corpus propriétaire composé de 2800 locuteurs, chacun comprenant 300 énoncés courts.

Ces avancées montrent que l'apprentissage automatique peut surmonter certaines des limitations des approches traditionnelles, non seulement pour la reconnaissance vocale, mais aussi pour l'identification des locuteurs. Cependant, ces travaux restent concentrés

---

[88] *Switchboard* et *Fisher* sont des corpus de conversations en anglais largement utilisés en recherche pour entraîner et tester des systèmes ASR. Ils servent de références standards pour mesurer la performance des systèmes en utilisant des métriques comme le WER.

[89] *Equal Error Rate*

sur des tâches spécifiques de reconnaissance et d'identification, sans aborder l'identification des sources sonores principales dans des environnements complexes.

2.2.3.4.1.2 Récupération d'informations musicales

Dans le domaine de l'analyse sonore, la récupération d'informations musicales (RIM[90]) est une autre application importante des techniques d'apprentissage automatique à l'aide de réseaux neuronaux artificiels. Cette technique consiste à développer des méthodes et des outils pour extraire, organiser et analyser automatiquement des informations pertinentes à partir de contenus musicaux numériques, tels que la reconnaissance automatique des instruments (RAIM), la classification des genres ou encore la recherche de morceaux similaires[91].

Initialement, cette tâche reposait sur des méthodes traditionnelles basées sur l'extraction de caractéristiques acoustiques comme les MFCC, combinées à des classificateurs tels que les SVM ou les k-NN. Par exemple, Eronen et Klapuri (2000) ont démontré que l'utilisation de méthodes basées sur l'extraction de caractéristiques acoustiques, telles que les MFCC, combinées à des classificateurs traditionnels comme les SVM, permettait d'améliorer la précision de la classification des instruments comparativement aux approches antérieures, ouvrant la voie à des techniques plus avancées.

Avec l'arrivée des DNN, la performance des systèmes de classification a connu une amélioration notable. Van den Oord et al. (2013) ont innové en appliquant des CNN à la recommandation musicale, en se concentrant sur la recommandation basée sur le contenu

---

[90] En anglais, on utilise *MIR* pour *Music Information Retrieval*. Cependant, cet acronyme présente une confusion avec une des techniques faisant partie des RIM (*MIR*), soit la reconnaissance automatique des instruments de musique (RAIM) qui devient, en anglais, *Musical Instrument Recognition* (*MIR* également). Nous choisissons donc, pour plus de clarté, d'utiliser les acronymes français, même s'ils sont moins usuels.

[91] D'autres exemples de tâches RIM : identification des artistes ou des albums, détection de plagiat ou de similitude musicale, reconnaissance des chansons, analyse de sentiment ou de l'humeur musicale.

plutôt que sur des approches collaboratives. En effet, leur modèle a permis d'analyser directement les caractéristiques audio pour capturer des préférences musicales subtiles, ce qui a révolutionné les systèmes de recommandation. Cette approche a jeté les bases des algorithmes modernes qui intègrent des analyses de contenu profondes pour offrir des suggestions plus personnalisées et pertinentes aux utilisateurs.

Dieleman et Schrauwen (2014) ont été parmi les premiers à proposer une approche de bout en bout utilisant des réseaux neuronaux convolutionnels profonds (DCNN[92]) pour l'analyse des signaux musicaux. Leur travail a démontré que les DCNN pouvaient apprendre directement à partir des données brutes sans nécessiter d'extraction manuelle des caractéristiques, marquant un tournant dans l'application des ANN à la musique.

Dans la continuité de cette évolution, Han et al. (2016) ont utilisé un réseau DCNN pour la reconnaissance des instruments prédominants dans des morceaux polyphoniques, où plusieurs instruments jouent simultanément. Leur approche a montré que les DCNN étaient capables d'extraire des caractéristiques hiérarchiques directement à partir des spectrogrammes, permettant ainsi de distinguer efficacement différents instruments dans des environnements complexes.

Pons et al. (2017) ont ensuite approfondi l'utilisation des DCNN pour l'analyse des timbres musicaux, démontrant que ces réseaux pouvaient améliorer la classification des genres et des instruments grâce à une analyse détaillée des caractéristiques sonores. Ce travail a renforcé l'importance des DCNN dans le domaine de l'analyse musicale.

De leur côté, Choi et al. (2017) ont montré que la combinaison de CNN et de RNN pouvait améliorer significativement les performances de l'étiquetage musical, une tâche clé de la RAIM qui inclut la classification des genres. Leur approche hybride a permis de capturer

---

[92] *Deep Convolutional Neural Network*

et d'interpréter les caractéristiques complexes des enregistrements audio, consolidant l'importance des réseaux profonds dans le traitement des signaux musicaux.

Plus récemment, l'application des Transformers à la classification des instruments et à d'autres tâches de RIM a montré des résultats prometteurs. Par exemple, l'étude de *Efficient Supervised Training of Audio Transformers for Music Representation Learning*[93] (Alonso-Jiménez et al., 2023) a démontré que les Transformers, lorsqu'ils sont correctement entraînés, peuvent surpasser d'autres architectures dans des tâches de classification musicale en capturant des informations complexes à partir des données musicales. De même, l'article *Equipping Pretrained Unconditional Music Transformers with Instrument and Genre Controls*[94] (Xu et al., 2023) a mis en avant l'utilisation des Transformers pour la génération musicale avec contrôle des instruments et des genres, prouvant leur potentiel pour des tâches complexes telles que la classification d'instruments.

2.2.3.4.2 Détection et classification des événements sonores

La détection et la classification des événements sonores sont un domaine fondamental dans l'analyse automatique des sons. Cette partie de notre revue se focalise sur les approches de ESC et de séparation des sources sonores à l'aide des DNN.

L'ESC se focalise sur l'identification et la classification de sons spécifiques dans divers environnements. Cette tâche consiste à distinguer des types de sons individuels parmi un ensemble de données sonores, et à leur attribuer une catégorie prédéfinie, comme les klaxons ou les sirènes, par exemple. La capacité à catégoriser ces sons est essentielle pour

---

[93] Traductions par l'auteur : *Entraînement supervisé efficace des Transformers audio pour l'apprentissage de représentations musicales.*

[94] *Doter les Transformers musicaux préentraînés et non conditionnés de contrôles pour les instruments et les genres*.

développer des systèmes capables de reconnaître et de différencier des sons spécifiques, ce qui constitue une base fondamentale pour des tâches plus complexes telles que la séparation des sources dans des scènes sonores plus denses et variées.

Cependant, dans des environnements plus complexes où plusieurs sources sonores se chevauchent, la séparation des sources sonores devient cruciale. Cette tâche, connue sous le nom de séparation aveugle des sources, a été traditionnellement abordée par des méthodes comme l'ICA, mais a récemment bénéficié des avancées en DNN.

Nous présentons donc successivement les travaux sur la détection d'événements sonores isolés, suivis de ceux sur la détection et la séparation de sources multiples, en mettant en avant les techniques DNN qui ont montré des performances significatives dans ces contextes.

2.2.3.4.2.1 Classification des sons environnementaux (ESC)

Les détection et identification d'événements sonores isolés consistent à identifier et à classer des sons spécifiques dans un flux audio continu.

Cette discipline bénéficie de jeux de données[95] de référence qui permettent l'entraînement des algorithmes ainsi que leur comparaison de performance. Les plus utilisés dans la littérature sont le UrbanSound8K et les ESC-50 et ESC-10 :

- UrbanSound8K, proposé par Salamon et al. (2014) est un jeu de données composé de 8732 clips audio de moins de 4 secondes, chacun appartenant à l'une des 10 classes sonores représentant des bruits urbains courants, tels que les klaxons de voiture, les alarmes ou encore les moteurs. Ce jeu de données est particulièrement pertinent pour l'évaluation des modèles dans des environnements urbains réels. Les enregistrements sont étiquetés et organisés pour permettre une validation croisée, ce qui en fait une référence essentielle pour le développement de systèmes de détection sonore.

[95] *Dataset*, en anglais, est courant.

- ESC-50 et ESC-10 sont des jeux de données créés par Karol J Piczak (2015) pour la classification des sons environnementaux. ESC-50 comprend 2000 clips audio couvrant 50 classes distinctes, et ESC-10 est une version simplifiée de ESC-50, contenant 400 clips audio répartis en 10 classes. Chaque clip a une durée de 5 secondes. Ces jeux de données sont organisés en catégories telles que les bruits d'animaux, les sons humains, et les bruits de transport, ce qui les rend pertinents pour tester la robustesse des modèles de classification sonore dans des contextes variés. Selon son auteur, la performance des humains pour la classification des sons du jeu de données ESC-50 est de 90 % de réussite.

Le Tableau 2.6 présente un sommaire des performances, sur ces 2 jeux de données, des modèles que nous répertorions dans cette partie de notre revue.

Tableau 2.6 : Comparatif des performances des modèles de classification présentés dans cette revue.

Étude	**Modèle utilisé**	**Précision UrbanSound8K**	**Précision ESC-50**
Karol J Piczak (2015)	Caractéristiques manuelles + SVM	-	44 %
Karol J. Piczak (2015)	CNN	74 %	65 %
Salamon et Bello (2017)	CNN + Data Augmentation	74 %	-
Tsalera et al. (2021)	CNN (VGGish)	80 %	70 %
Vidaña-Vila, Navarro, Borda-Fortuny, et al. (2020)	CNN sur Raspberry Pi (SqueezeNet)	80 %	-
Tsalera et al. (2021)	CNN (YAMNet - MobileNet)	82 %	72 %
Aytar et al. (2016)	CNN (SoundNet)	-	74 %
Esmaeilpour et al. (2020)	WCCGAN	94 %	77 %
Al-Hattab et al. (2021)	CNN optimisé (SEnv-Net)	96 %	-
İnik (2023)	CNN avec optimisation automatique des hyperparamètres	98 %	97 %

Comme nous le mentionnions en 2.2.1, les premières méthodes de classification automatique des sons se basaient sur l'extraction de caractéristiques acoustiques telles que les descripteurs de forme d'onde[96] et les MFCC, suivie par des algorithmes de

[96] Descripteurs du domaine temporel : amplitude, puissance, énergie, durée, enveloppe du signal.

classification traditionnels comme les SVM ou les HMM. Selon Karol J Piczak (2015), les performances de ces systèmes « à caractéristiques manuelles[97] » atteignent une valeur de base établie à 44 % de succès sur ESC-50.

Avec l'avènement des DNN, la précision et la robustesse des systèmes de détection d'événements sonores isolés se sont considérablement améliorées. Par exemple, Karol J. Piczak (2015) a utilisé des CNN pour la détection d'événements sonores dans des environnements variés. Le modèle utilisé par Piczak est composé de deux couches convolutionnelles suivies de couches de pooling, conçu pour extraire des caractéristiques à partir de spectrogrammes d'entrée et les classifier en utilisant des couches entièrement connectées avec une fonction softmax en sortie, atteignant une précision de 73,7 % sur le jeu de données UrbanSound8K, et 64,5 % sur ESC-50. Cette méthode a donc montré des performances supérieures aux approches traditionnelles, en particulier dans des environnements bruyants ou non contrôlés.

Avec l'évolution des DNN dans le domaine de la classification sonore, le *Transfer Learning* est devenu une approche de plus en plus courante. Cette technique consiste à utiliser des modèles préentraînés sur de grandes bases de données pour ensuite les affiner sur des tâches spécifiques avec des ensembles de données plus petits. Cette approche permet non seulement de gagner du temps et des ressources, mais aussi d'améliorer les performances en s'appuyant sur des représentations de caractéristiques déjà bien optimisées. Des exemples notables de cette approche dans le domaine de l'audio sont *VGGish*, *SoundNet*, et *YAMNet* :

---

[97] Piczak écrit « manually-engineered features » pour décrire les approches qui reposent sur des caractéristiques acoustiques extraites manuellement, telles que les MFCC, combinées à des classificateurs traditionnels comme les SVM ou les HMM.

- *VGGish*[98] est une variante de l'architecture VGG[99] (Simonyan & Zisserman, 2014), adaptée par Google pour la classification sonore. Ce modèle transforme les signaux audio en spectrogrammes log-mel qu'il utilise comme entrées pour extraire des caractéristiques acoustiques profondes. Pour son préentraînement, *VGGish* utilise *YouTube-8M* (Abu-El-Haija et al., 2016), un corpus de plus de 8 millions de vidéos YouTube annotées automatiquement. Cependant, contrairement à SoundNet, ci-dessous, *VGGish* se concentre exclusivement sur l'extraction de caractéristiques audio des vidéos, sans exploiter la composante visuelle. Cela permet au modèle de capturer des représentations acoustiques généralisables, indépendamment du contexte visuel. Selon Tsalera et al. (2021), *VGGish* a atteint une précision de 80 % sur UrbanSound8K et de 70 % sur ESC-50.
- *SoundNet* (Aytar et al., 2016) est un autre exemple marquant d'approche par transfer learning dans le domaine de l'audio. Ce modèle, basé sur un DNN comprenant 8 couches convolutionnelles suivies de 2 couches entièrement connectées, a été préentraîné sur un large corpus de vidéos YouTube. Contrairement à *VGGish*, *SoundNet* tire parti de la synchronisation naturelle entre l'audio et le visuel dans ces vidéos pour apprendre des représentations audio plus riches et contextuellement informées. Cette approche multimodale permet à SoundNet de généraliser efficacement à diverses tâches de classification sonore, y compris dans des environnements urbains complexes. Selon ses auteurs, SoundNet a atteint une précision de 74 % sur ESC-50.
- *YAMNet* (Google Research, 2021) est un modèle plus récent, basé sur l'architecture MobileNet (Howard et al., 2017) et préentraîné sur *AudioSet* (Gemmeke et al., 2017), un corpus de plus de deux millions de clips sonores. Soulignons que MobileNet a pour objectif d'être économe en ressource afin de permettre le déploiement sur des appareils mobiles. *YAMNet*, qui est conçu pour la reconnaissance des sons dans des environnements complexes, atteint une précision de 82 % sur UrbanSound8K et 72 % sur ESC-50, toujours selon Tsalera et al. (2021).

À la même période, plusieurs études dans le domaine de la classification des sons environnementaux ont intégré des techniques d'augmentation de données pour améliorer

---

[98] Le terme VGGish a été utilisé par des ingénieurs et chercheurs de Google pour décrire une variante de l'architecture VGG adaptée à la classification sonore. La référence habituellement citée pour ce modèle est Hershey, et al. (2017) qui présentent une comparaison de différents modèles CNN, dont cette variante de VGG.

[99] Voir aussi la Figure 2.4.

les performances des modèles DNN. Ces techniques, telles que le time-shifting (décalage temporel), le pitch-shifting (modification de la hauteur tonale), par exemple, permettent de générer des variantes synthétiques des données d'entraînement, augmentant ainsi la diversité des exemples disponibles pour le modèle. L'ajout de ces données artificiellement enrichies aide à prévenir le surapprentissage en forçant le modèle à mieux généraliser face à des variations inattendues dans les données réelles. Cette approche s'est avérée importante dans l'évolution de la reconnaissance des sons à l'aide de CNN, comme en témoignent des travaux influents tels que ceux de Salamon et Bello (2017) et d'autres chercheurs de l'époque, notamment Zhang et al. (2018). L'intégration systématique de l'augmentation de données a ainsi établi un nouveau standard dans la formation des modèles de classification sonore, permettant des avancées significatives en matière de robustesse et de précision.

Pour leur part, Esmaeilpour et al. (2020) ont proposé une approche novatrice en utilisant un *réseau antagoniste génératif pondéré et cohérent par cycle* (WCCGAN)[100] pour l'augmentation de données en mode non supervisé. Un réseau antagoniste génératif (GAN[101]) est un modèle composé de deux réseaux neuronaux : un générateur, qui crée des données synthétiques, et un discriminateur, qui tente de distinguer ces données générées des données réelles. Ces deux réseaux s'affrontent dans un processus d'apprentissage compétitif : le générateur s'efforce de produire des exemples synthétiques qui ressemblent le plus possible aux données réelles, tandis que le discriminateur essaie de différencier ces exemples générés des véritables données. Au fur et à mesure que l'entraînement progresse, le générateur devient de plus en plus habile à créer des données synthétiques qui trompent le discriminateur, et ce dernier s'améliore à identifier les faux. Ce duel améliore continuellement la qualité des données générées, jusqu'à ce qu'elles deviennent presque

---

[100] *Weighted Cycle-Consistent Generative Adversarial Network*.

[101] *Generative Adversarial Network*

indiscernables des données réelles. Grâce à l'intégration des données synthétiques générées par le WCCGAN, le modèle CNN proposé par Esmaeilpour et al. a atteint un score de performance de 93 % sur le jeu de données UrbanSound8K.

Plus récemment, Al-Hattab et al. (2021) ont apporté une contribution significative, non pas en se concentrant sur l'augmentation de données, mais en optimisant l'utilisation des CNN pour la classification des sons environnementaux. Leur étude s'est focalisée sur l'optimisation des paramètres de l'extraction de caractéristiques, en montrant qu'un réglage fin des hyperparamètres des CNN pouvait améliorer considérablement les performances de la détection des événements sonores tout en conservant un modèle DNN moins gourmand en ressources computationnelles. Leur modèle, *SEnv-Net*, a atteint des performances de 96 % sur UrbanSound8K, confirmant la capacité des CNN à s'adapter à des conditions complexes grâce à une optimisation soignée.

Notons que, dans le cadre de ce travail, nous avons modifié le modèle *SEnv-Net* proposé par Al-Hattab et al. pour l'utiliser en contexte de réseaux siamois avec une fonction de perte par triplet. Cette adaptation permet de répondre aux besoins spécifiques de notre projet, qui consiste à identifier et à différencier les sources sonores principales dans des environnements complexes à l'aide d'un système rapide.

Comme exemples de modèles de ESC très récents qui excellent en performance, mentionnons le travail de Özkan İnik (2023), qui a mené une étude où il optimise automatiquement les hyperparamètres d'un CNN comprenant six couches convolutionnelles. Grâce à cette optimisation systématique, le modèle atteint une précision de plus de 98 % sur UrbanSound8K, confirmant l'importance de l'ajustement des hyperparamètres pour maximiser les performances des CNN dans la classification sonore.

D'autre part, une approche complémentaire à la détection d'événements sonores isolés repose sur l'utilisation de réseaux de capteurs déployés dans des environnements urbains pour la détection et la classification automatique des sons. Dans l'étude *Low-Cost*

Distributed Acoustic Sensor Network for Real-Time Urban Sound Monitoring, l'équipe Vidaña-Vila, Navarro, Borda-Fortuny, et al. (2020) utilise un CNN pour l'identification des événements sonores urbains en temps réel à l'aide de dispositifs à faible coût, tels que le Raspberry Pi[102]. Le modèle DCNN utilisé, SqueezeNet 18 (Iandola et al., 2016), est optimisé pour une classification rapide des événements sonores malgré les capacités limitées en calcul et en stockage des dispositifs. Il a permis d'atteindre une précision de 80 % sur UrbanSound8K, et de 64 % en réseau de 4 capteurs, en laboratoire, où des enregistrements de bruit de fond urbain étaient ajoutés au jeu de données original. Notons que le modèle n'a pas été réentraîné pour cette dernière expérience, de façon à reproduire au mieux les conditions réelles. Mentionnons également les travaux de Claudi Socoró et al. (2017), Ginovart-Panisello et al. (2021) ou encore Bonet-Solà et al. (2023a, 2023b), qui travaillent aussi sur des recherches en ESC par réseaux de capteurs.

Ces réseaux de capteurs à faible coût sont une solution potentielle pour répondre aux défis liés à la comparaison des sources mobiles individuelles à une captation complexe, fixe, et éloignée, comme requis par notre projet. Cette approche permettrait de créer un système de surveillance acoustique distribué capable de capturer des sons provenant de diverses sources mobiles, tout en maintenant une analyse centralisée efficace et capable d'identification en temps quasi réel.

2.2.3.4.2.2 Séparation des sources sonores à l'aide des DNN

La détection et la séparation de sources multiples ajoutent un niveau de complexité à l'identification d'événements sonores isolés. Ici, l'objectif est non seulement de détecter des événements sonores dans un environnement complexe, mais aussi de les séparer en sources distinctes si elles sont concurrentes.

[102] https://www.raspberrypi.com/

Dans le cadre de ce travail, une revue des techniques de séparation des sources sonores permet d'évaluer les approches capables d'identifier et de distinguer les sources sonores principales dans des environnements complexes. Les techniques de séparation des sources permettent de décomposer un signal audio en ses composantes d'origine, une capacité cruciale pour isoler les sons d'intérêt dans des enregistrements où plusieurs sources sont présentes simultanément.

Les premières approches pour cette tâche utilisaient des méthodes de séparation aveugle des sources (BSS[103]) telles que l'ICA. Cependant, comme nous l'expliquions en 2.2.1.2.2, ces techniques avaient souvent des limites en matière de robustesse, en particulier lorsqu'il s'agit de séparer des sources qui se chevauchent dans le temps ou dont le contenu fréquentiel est similaire.

Notons que, bien que la littérature sur les avancées en BSS spécifiquement appliquées à l'ESC soit relativement limitée, les techniques explorées dans d'autres contextes, comme la séparation des voix dans la musique ou la parole, sont applicables et offrent des enseignements pertinents pour l'ESC.

Une étude significative dans ce domaine est celle de Huang et al. (2014), qui a utilisé des RNN pour la séparation de la voix dans des enregistrements musicaux chantés. Leur méthode repose sur l'apprentissage des caractéristiques temporelles des signaux audio, permettant une séparation plus efficace des sources sonores. Le modèle a montré de très bonnes performances dans la séparation des voix, mais il est limité lorsqu'il est confronté à des signaux non musicaux ou plus complexes. Par exemple, la précision de la séparation diminue lorsque les sources sont fortement superposées ou lorsque les données d'entraînement sont insuffisantes. Les auteurs soulignent en outre que l'efficacité des

[103] *Blind Source Separation*

modèles DNN ou DRNN[104] dépend fortement de la qualité et de la diversité des données d'entraînement, ces éléments étant cruciaux pour garantir des performances robustes, en particulier dans les cas d'environnements complexes.

Un autre travail phare pour la séparation des sources sonores est le modèle *Deep Clustering*, proposé par Luo et al. (2017). Il s'agit d'une approche marquante qui a trouvé des applications dans divers domaines où la séparation des sons concurrents est requise. Cette méthode repose sur l'apprentissage non supervisé, où chaque segment audio est transformé en un vecteur numérique représentant ses caractéristiques acoustiques. Ces vecteurs sont ensuite organisés dans un espace complexe, regroupant les segments provenant de la même source sonore tout en séparant ceux provenant de sources différentes. Cette organisation permet de faciliter la séparation des sources, même dans des environnements sonores complexes.

Cette méthode innovante a permis d'améliorer la capacité à séparer des sources sonores multiples, même dans des environnements complexes où les signaux sont fortement superposés. Le *Deep Clustering* a ouvert la voie à des avancées majeures dans la séparation des sources, en particulier pour des applications comme l'isolement de voix dans des enregistrements multiorateurs ou la séparation de sons dans des scènes sonores riches.

Cependant, bien que le *Deep Clustering* représente un bond en avant significatif, il reste limité par le fait qu'il ne prend pas directement en compte les relations temporelles entre les segments audio. C'est ici qu'intervient le travail de Wang et Wang (2018), qui a introduit la combinaison des CNN et RNN pour améliorer la détection et la séparation de sources multiples. Leur modèle exploite à la fois les caractéristiques spatiales capturées par les CNN et les dépendances temporelles modélisées par les RNN. Le RNN est

---

[104] *Deep Recurrent Neural Network* (réseau neuronal récurrent profond)

particulièrement adapté pour gérer les transitions temporelles entre les sources sonores, permettant ainsi au modèle de suivre de manière plus fluide et précise l'évolution des sources dans le temps. Cette approche a montré une capacité accrue à séparer des sources concurrentes, même lorsqu'elles se chevauchent temporellement, ce qui est un défi majeur dans des environnements réalistes tels que des scènes urbaines ou des enregistrements de conversations multiples.

Malgré ces avancées, ces modèles ne répondent pas entièrement à notre objectif d'identification des sources sonores principales dans un environnement complexe. Le *Deep Clustering* est puissant pour la séparation des sources, mais peut échouer dans des situations où les sources ont beaucoup de similitudes ou si peu de données d'entraînement sont disponibles. De plus, l'approche RNN+CNN de Wang et Wang (2018), bien qu'efficace pour capturer les relations temporelles, peut souffrir de problèmes de complexité computationnelle et de difficulté d'entraînement lorsque les séquences sont longues ou lorsque les sources sont très mélangées. Ces limitations montrent que, pour atteindre une identification robuste et précise des sources principales, des méthodes plus avancées ou des combinaisons de techniques pourraient être nécessaires, incluant potentiellement des modèles de réseaux neuronaux plus sophistiqués ou des approches hybrides intégrant d'autres formes d'apprentissage automatique.

2.2.4 Absence de travaux directs sur l'identification des sources sonores principales

Comme nous avons tenté de le démontrer, la recherche dans le domaine de l'analyse sonore à l'aide des réseaux neuronaux artificiels a, jusqu'à présent, principalement ciblé la classification des sons ou encore la séparation des sources sonores, typiquement en vue, ici aussi, de leur classification. Ces approches, bien qu'avancées, ne tentent pas directement d'identifier la source sonore dominante dans des environnements complexes et dynamiques. Or, notre travail se distingue par son approche intégrative qui combine la séparation des sources sonores à la comparaison des signaux en captation rapprochée et éloignée. De plus, cette identification doit être accomplie en temps réel.

En effet, dans le contexte de notre étude, l'objectif est de relier des sources sonores simples captées de manière rapprochée à un environnement sonore complexe capté à distance, ceci afin d'identifier la source prédominante. Ce processus nécessite non seulement une séparation des sources, mais aussi une comparaison des signaux alors que la source dominante n'a pas été utilisée lors de l'entraînement du modèle. En outre, notre système doit être suffisamment agile pour s'adapter en temps réel à des environnements sonores imprévus.

Ce défi, impliquant à la fois une adaptation en temps réel et une robustesse face à des données non supervisées, ne semble pas avoir encore été directement abordé dans la littérature. Notre recherche vise à combler cette lacune.

2.2.4.1 Classification

Les techniques existantes en classification sonore permettent de comparer des sons aux caractéristiques similaires, offrant ainsi des solutions adaptées à la différenciation de sources. Cependant, elles fonctionnent généralement en attribuant chaque son à une catégorie prédéfinie, dans le but de l'étiqueter. Cette approche, bien que pertinente pour des tâches de classification, ne répond pas à nos besoins. Dans notre cas, l'étiquette est inutile, car nous cherchons à établir une correspondance entre un son et une référence qui n'existe pas avant l'événement à analyser. Notre système doit être capable de constamment adapter cette référence, puisque « l'entraînement » se fait au fur et à mesure que les nouveaux sons sont rencontrés.

2.2.4.2 Séparation des sources sonores

En ce qui concerne la séparation des sources sonores, les techniques actuelles permettent d'isoler des signaux imbriqués, réduisant ainsi le bruit de fond pour mettre en avant le son cible. Cela répond en partie à nos besoins, car la mise en évidence d'un son à analyser est essentielle. Cependant, dans notre cas, toutes les sources peuvent potentiellement être la source principale. Les sons secondaires ne sont pas simplement des bruits parasites à éliminer, mais des concurrents qu'il faut analyser comme des candidats possibles pour la

position de source dominante. Cela nécessite une approche plus flexible, où l'identification de la source principale est déterminée en fonction du contexte sonore en temps réel, sans catégorisation préalable.

En résumé, bien que les techniques existantes offrent des solutions pour la classification et la séparation sonore, elles ne prennent pas en compte les spécificités de notre approche. Il manque une méthode capable d'adapter dynamiquement les références sonores et de traiter les sources sonores concurrentes comme des candidats potentiels à l'identification de la source principale, tout en assurant une réponse en temps réel.

2.2.5 Synthèse des techniques et justification du choix expérimental

À travers notre revue de la littérature, plusieurs techniques se sont toutefois révélées cruciales pour l'analyse sonore et la classification des événements sonores, contribuant directement à la décision de notre approche expérimentale. Les principales contributions sont les suivantes :

- Extraction de caractéristiques avec les MFCC :

Les MFCC ont démontré leur efficacité comme caractéristiques robustes pour la classification sonore, en particulier dans des environnements variés. Leur capacité à capturer les informations spectrales pertinentes tout en réduisant la dimensionnalité des données a été maintes fois validée. Leur application dans les CNN pour l'extraction de caractéristiques à partir de spectrogrammes a été une étape décisive vers des systèmes de reconnaissance plus performants (2.1.2.7).

- Utilisation des CNN pour la classification sonore :

Les CNN ont considérablement amélioré la précision des systèmes de détection et de classification des sons. Ces modèles exploitent les caractéristiques spectrales des signaux audio pour différencier des classes sonores même dans des environnements bruités, ce qui est essentiel pour notre projet d'identification des sources sonores principales (2.1.3.5).

- Optimisation des CNN pour les ESC :

L'étude d'Al-Hattab et al. (2021) a apporté des améliorations significatives dans l'utilisation des CNN pour la classification des sons environnementaux. Leur modèle SEnv-Net, optimisé pour réduire les besoins en ressources tout en améliorant la précision, a servi de base à notre expérimentation. La capacité de ce modèle à s'adapter à des conditions complexes par un réglage fin des hyperparamètres a été déterminante pour le choix de notre architecture (2.2.3.4.2.1).

- Approche par réseaux siamois avec fonction de perte par triplet :

L'architecture CNN siamoise couplée à une fonction de perte par triplet offre une solution prometteuse pour la comparaison des signaux audio dans des environnements complexes. Cette approche permet de différencier efficacement les sources sonores, même sans préentraînement sur ces dernières, ce qui est crucial pour notre objectif d'identification des sources dominantes dans un environnement dynamique (2.1.3.6).

Ces éléments combinés justifient l'utilisation des MFCC vers un modèle SCNN avec fonction de perte par triplet, en utilisant le modèle SEnv-Net comme point de départ. Cette approche répond à la complexité de la tâche d'identification des sources sonores principales, en particulier dans des environnements bruyants et en temps réel.

* * *

CHAPITRE 3 : MÉTHODOLOGIE

Dans ce chapitre, nous présentons la méthodologie développée pour l'identification de la source sonore principale dans des environnements complexes et dynamiques. Notre approche combine l'extraction de caractéristiques acoustiques par MFCC avec un SCNN utilisant une fonction de perte par triplet. En nous appuyant sur le modèle SEnv-Net comme point de départ, nous avons conçu un système capable de s'adapter à des sources sonores non préalablement rencontrées.

En raison de l'absence d'accès à des captations éloignées réelles, nous avons simulé des enregistrements d'environnements complexes à partir des captations rapprochées. Cette simulation nous a permis de contrôler précisément les sources sonores présentes et de générer des scénarios variés pour l'entraînement et l'évaluation du modèle.

3.1 Note terminologique sur « identification » et « classification »

Tout au long de ce mémoire, nous utilisons le terme « identification » pour désigner la tâche spécifique qui consiste à retrouver, parmi plusieurs sources candidates, celle qui est la plus présente (ou dominante) dans un signal complexe mesuré à distance.

Cette approche se distingue de la classification automatique, au sens strict utilisé en apprentissage supervisé, où chaque entrée est associée à une étiquette de classe prédéfinie. Ici, aucune classe catégorielle n'est prédéfinie dans le modèle : le système ne prédit pas une étiquette, mais compare la similarité entre un signal ancre et différents candidats afin d'identifier celui qui correspond le mieux.

Toutefois, plusieurs des techniques utilisées — notamment l'extraction de caractéristiques ou les principes d'entraînement des réseaux — proviennent du champ de la classification automatique des sons. Leur emploi dans le cadre d'un modèle siamois à fonction de perte par triplet constitue un détournement méthodologique adapté à notre objectif d'identification comparative.

3.2 Description générale de l'approche

L'objectif principal de notre méthodologie est de relier des sources sonores simples, captées de manière rapprochée, à des environnements sonores complexes simulés, afin d'identifier la source prédominante. Pour ce faire, nous procédons selon les étapes suivantes :

1. Collecte des données : Acquisition d'enregistrements audio en captation rapprochée (sources individuelles) et génération de simulations de captations éloignées.
2. Prétraitement des données : Normalisation et segmentation des signaux audio, extraction des caractéristiques à l'aide des MFCC.
3. Construction du modèle : Adaptation du modèle SEnv-Net en une architecture siamoise avec fonction de perte par triplet.
4. Entraînement du modèle : Utilisation de triplets de signaux pour entraîner le réseau à distinguer les sources sonores.
5. Évaluation et ajustement : Test du modèle sur des simulations variées et ajustement des hyperparamètres pour optimiser les performances.

Nous décrivons ces étapes dans la suite de ce texte.

3.3 Collecte et préparation des données

3.3.1 Enregistrements en captation rapprochée

Nous bénéficions d'une banque de sons enregistrés automatiquement par un système de monitorage de la nuisance sonore situé en bordure des terrains de notre partenaire. Ce système, en place depuis 3 années au moment de ce travail, a enregistré plus de 93 000 clips de 20 secondes au format WAV (22 kHz, 8 bits). Ce système est réglé de façon à démarrer un enregistrement lorsque des sources sonores dépassent un niveau de puissance correspondant à des sources rapprochées.

Les sources sont :

- Alarmes de recul (à tonalité ou bruit blanc) ;
- Moteurs à combustion interne (au ralenti, en fonctionnement, en passage) ;

- Impacts (bois, métal) ;
- Manipulation de grandes plaques métalliques ;
- Manipulation de poutres et barres d'acier (frottement, résonance) ;
- Passage de trains de marchandises (grondement, grincement).

Pour évaluer la robustesse de notre modèle et assurer une généralisation efficace, nous avons créé successivement 12 groupes distincts de 10 000 sons choisis aléatoirement parmi les 93 000 clips disponibles. Chaque groupe a été traité indépendamment pour générer les données d'entraînement, de validation et de test.

Pour chaque groupe :

- Préparation des ancres : Les 10 000 enregistrements ont généré environ 8 400[105] sons ancres après prétraitement.
- Génération des jumeaux : Pour chaque ancre, nous avons produit 20 jumeaux positifs et 20 jumeaux négatifs[106].
- Division des données : Les données ont été réparties en 30 % pour le test, 42 % pour l'entraînement (soit 60 % du 70 % restant), et 28 % pour la validation (soit 40 % du 70 % restant).

Ainsi, en totalisant les 12 groupes, nous avons réalisé 60 entraînements (12 groupes × 5 S/B), permettant de renforcer la fiabilité statistique de nos observations.

3.3.2 Prétraitement des enregistrements

3.3.2.1 Sélection et segmentation des extraits sonores

Afin de préparer les enregistrements en captation rapprochée pour l'entraînement, nous avons développé un script en Python utilisant les bibliothèques librosa (McFee et al.,

[105] Les 10 000 fichiers de départ étant choisis aléatoirement dans la banque, le nombre de fichiers produit varie.

[106] Soit 168 000 jumeaux positifs et autant de négatifs.

2015a) et soundfile (Bittner, 2020). Ce script permet de sélectionner et de segmenter automatiquement les extraits sonores pertinents à partir des enregistrements bruts.

Le code Python complet utilisé et expliqué dans ce chapitre est présenté à l'annexe A. Nous présentons dans le texte la logique générale sous forme de pseudocode :

```
Encadré 1. Prétraitement des enregistrements
(Code Python correspondant : annexe A, section A.1)

FONCTION Nettoyer_audio(input_filename, paramètres...)
    Charger le signal audio depuis le fichier
    Calculer l'enveloppe RMS du signal (fenêtrage hop_length)
    Convertir l'enveloppe en décibels (dB)

    SI pondération A activée:
        Calculer les fréquences pour la FFT
        Appliquer la pondération A aux niveaux dB
        Ajuster l'échelle (ex. : +80 dB)
    SINON :
        Utiliser les niveaux dB bruts

    Calculer le niveau moyen pondéré
    Définir un seuil ajusté (niveau moyen + seuil)

    Déterminer les portions du signal dépassant ce seuil
    Extraire ces segments et les sauvegarder
    Retourner les segments et la liste des fichiers créés
FIN FONCTION
```

Dans cette fonction :

- Chargement et normalisation : Le signal audio est chargé à partir du fichier d'entrée, avec sa fréquence d'échantillonnage native. Aucune normalisation explicite n'est faite dans le code, mais l'analyse ultérieure se fait en termes d'énergie RMS[107] et de niveaux en dB, ce qui assure une cohérence dans les comparaisons de puissance.
- Calcul de l'enveloppe RMS : Cela permet d'estimer l'énergie moyenne du signal sur des fenêtres temporelles définies par `hop_length`. Cette mesure sert de base pour évaluer les variations d'énergie du signal.

[107] *Root Mean Square* : Mesure correspondant à la racine carrée de la moyenne des carrés des valeurs d'un signal. Utilisé pour évaluer l'amplitude moyenne efficace du signal.

- Conversion en dB et application de la pondération A :
    - La conversion en dB est réalisée à partir de l'enveloppe RMS, ce qui permet de manipuler les niveaux d'énergie sur une échelle logarithmique.
    - La pondération A est appliquée après la conversion en dB, et elle ajuste les niveaux d'énergie pour refléter la sensibilité humaine aux différentes fréquences. Les fréquences sont calculées à l'aide de la FFT afin d'appliquer correctement la pondération A.
    - L'ajout de 80 dB compense les ajustements de la pondération A pour ramener les valeurs dans une échelle comparable aux niveaux sonores courants.
- Calcul du niveau moyen et ajustement du seuil :
    - Le niveau moyen pondéré A est calculé pour estimer le bruit de fond ou l'énergie globale du signal. Le seuil est ajusté à partir de ce niveau moyen afin de déterminer quelles parties du signal sont suffisamment significatives pour être extraites.
- Détection des segments significatifs : Les segments du signal dont l'énergie pondérée A dépasse le seuil ajusté sont considérés comme pertinents, et sont extraits pour être sauvegardés dans de nouveaux fichiers.

Ce processus permet de segmenter et de sauvegarder les parties significatives du fichier audio de manière cohérente, avec une pondération A pour mieux correspondre à la perception humaine du son, comme c'est le cas dans les applications de gestion du bruit ou de réglementation acoustique.

3.3.2.2 Paramètres utilisés

Les paramètres spécifiques pour la sélection et la segmentation sont :

- Seuil d'énergie : Fixé à 6 dB au-dessus de la puissance moyenne de chaque clip de 20 secondes pour détecter les événements sonores significatifs.
- Durée minimale des segments : Les segments doivent avoir une durée minimale de 0,5 seconde pour être conservés.
- Durée cible des fichiers : Les extraits sont découpés en segments de 2 secondes pour l'entraînement. S'ils sont plus courts, ils sont dupliqués et enchaînés jusqu'à ce que le fichier dure 2 secondes.

3.3.3 Simulation des enregistrements en captation éloignée

3.3.3.1 Méthodologie de simulation

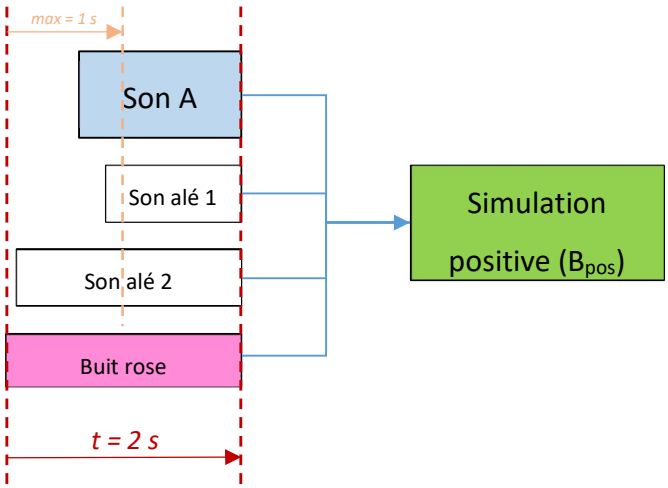
En l'absence de captations éloignées réelles, nous avons généré des enregistrements simulés d'environnements complexes à partir des captations rapprochées. En préparation de l'analyse avec fonction de perte par triplet, la démarche est la suivante :

- Création de triplets (ancre, positif, négatif) : Pour chaque son de la banque (noté *A*), plusieurs paires de jumeaux positifs et négatifs sont générées, avec l'ajout de bruit rose et deux sons aléatoires, selon le S/B requis.
- Jumeaux positifs :
    - Son principal (dominant) : Le son ancre (*A*) est utilisé comme source dominante dans les jumeaux positifs.
    - Retard aléatoire : Le son ancre (*A*) est retardé aléatoirement avec un retard maximum fixé à la moitié de la durée du fichier, soit 1 seconde. Ce retard simule une différence temporelle correspondant à environ 340 m entre les captations rapprochées et éloignées. Il permet toutefois de s'assurer que chaque son soit présent, dans le fichier jumeau, pendant une durée significative.
    - Mélange avec des sons aléatoires : Deux autres sons aléatoires (appelés *alé 1* et *alé 2*) sont sélectionnés parmi les autres fichiers, retardés aléatoirement et mélangés avec le son ancre. Ces sons sont différents du son *A* et l'un de l'autre.
    - Ajout de bruit rose : Du bruit rose est ajouté au mélange pour simuler le bruit ambiant. Ce bruit est réglé à -10 dBFS avant d'être ajouté à *alé 1* et *alé 2* pour créer la partie *bruit total* du jumeau simulé.
    - Le S/B est défini comme le rapport entre *A* (le signal dominant) et le *bruit total* constitué de *alé 1*, *alé 2*, et du bruit rose. Ce rapport est réglé à différentes valeurs en cours d'expérimentation.
- Jumeaux négatifs :

  La démarche de création des jumeaux positifs est reprise, avec les différences suivantes :
    - Son principal (dominant) : Un son aléatoire (*alé 1*) est sélectionné comme source dominante dans les jumeaux négatifs, plutôt que le son ancre *A*.
    - Mélange : Le mélange final inclut le son dominant (*alé 1*), ainsi que le *bruit total* constitué de l'ancre retardée (*A*), un autre son aléatoire (*alé 2*) et du bruit rose.

Ainsi, dans les jumeaux négatifs, le son ancre est présent dans le mélange, mais il n'est pas la source dominante. Le modèle doit donc apprendre à distinguer non pas la présence du son ancre, mais bien s'il est la source principale présente dans le mélange (Figure 3.1).

a)



b)

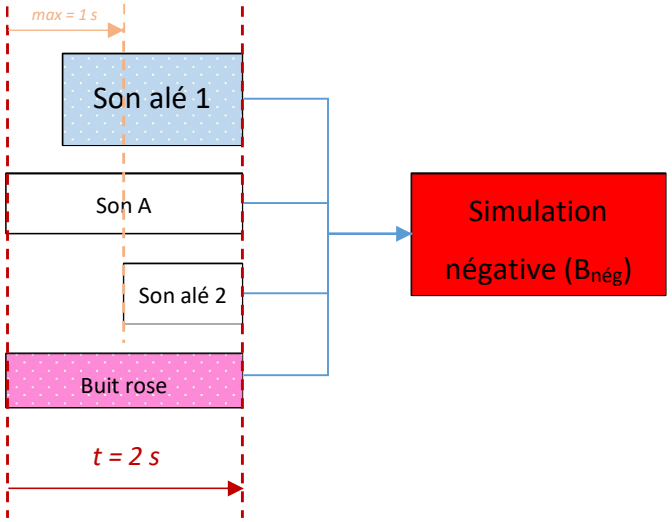


Figure 3.1 : Représentation schématique des simulations de captations éloignées. a) A est la composante principale en B (jumeau positif), et b) A n'est pas la composante principale en B (jumeau négatif).

Le pseudocode suivant résume la logique utilisée :

```
Encadré 2. Pseudocode de la génération des jumeaux positifs et négatifs
(Code Python correspondant : annexe A, section A.2)

FONCTION Ajouter_retard_aleatoire(signal, retard_max)
    Générer un retard aléatoire entre 0 et retard_max
    Couper la fin du signal selon le retard
    Insérer un silence équivalent au début
    Retourner le signal retardé
FIN FONCTION

FONCTION Créer_jumeau_positif(ancre, sons_aléatoires, bruit_rose, S/B)
    Retarder aléatoirement l'ancre
    Retarder deux sons aléatoires
    Mélanger les sons non dominants avec le bruit rose
    Normaliser le mélange
    Ajuster le niveau du fond pour obtenir le S/B souhaité
    Ajouter le fond sonore à l'ancre
    Normaliser le résultat
    Retourner le jumeau positif
FIN FONCTION

FONCTION Créer_jumeau_négatif(ancre, sons_aléatoires, bruit_rose, S/B)
    Retarder un son aléatoire (source dominante)
    Retarder l'ancre (présente, mais non dominante)
    Retarder un autre son aléatoire
    Mélanger l'ancre, ce son aléatoire et le bruit rose
    Normaliser le mélange
    Ajuster le niveau du fond pour obtenir le S/B souhaité
    Ajouter le fond sonore à la source dominante
    Normaliser le résultat
    Retourner le jumeau négatif
FIN FONCTION
```

Ce procédé permet de générer les triplets nécessaires à l'entraînement du réseau selon la logique de la fonction de perte par triplet.

3.3.3.2 Paramètres de simulation

- Nombre de jumeaux : Pour chaque ancre, nous avons généré 20 paires positif-négatif.
- Rapport signal/bruit (S/B) : Nous avons entraîné notre modèle sous 5 niveaux de S/B, soit 0, 3, 6, 10 et 13 dB. Pour chaque groupe de données (12 groupes au total), le modèle a été entraîné séparément sous chaque S/B, totalisant 60 entraînements (12 groupes × 5 S/B).

3.3.3.3 Justification de la méthode

- Complexité accrue : En incluant l'ancre dans le jumeau négatif comme son non dominant, nous créons une situation où le modèle doit apprendre non seulement à reconnaître le son ancre, mais aussi à déterminer s'il est prédominant.
- Simuler des environnements réels : Les environnements sonores réels sont souvent complexes, avec de multiples sources sonores présentes simultanément. Notre méthode vise à reproduire cette complexité.
- Robustesse du modèle : En entraînant le modèle sur ces scénarios complexes, nous améliorons sa capacité à généraliser et à fonctionner efficacement dans des conditions réelles variées.

3.4 Extraction des caractéristiques

3.4.1 Calcul des MFCC

Dans notre classe personnalisée de type `TripletDataset`, chaque triplet (ancre, positif, négatif) est chargé à partir de fichiers audio, puis transformé en représentation cepstrale à l'aide de la transformation MFCC fournie par la bibliothèque `torchaudio` (Meta AI, 2023). Ces caractéristiques sont ensuite utilisées comme entrées du modèle siamois.

Le pseudocode suivant illustre la logique appliquée.

```
Encadré 3. Pseudocode de l'extraction des MFCC pour chaque triplet
(Code Python correspondant : annexe A, section A.3)

FONCTION Extraire_MFCC_Triplet(fichiers_audio, paramètres_MFCC)
    Initialiser la transformation MFCC avec les paramètres spécifiés

    POUR chaque fichier audio du triplet (ancre, positif, négatif) :
        Charger le signal audio
        Appliquer la transformation MFCC
        Réorganiser la forme du tenseur pour correspondre au format
attendu
    FIN POUR

    Retourner les trois représentations MFCC
FIN FONCTION
```

Cette étape permet de transformer les signaux temporels bruts en représentations compactes de leur contenu fréquentiel, selon l'échelle Mel. Les MFCC extraits intègrent les propriétés spectrales pertinentes pour l'identification des sons, tout en réduisant la

dimensionnalité des données. Une même transformation est appliquée à chacun des fichiers audio du triplet, assurant une cohérence dans les comparaisons ultérieures effectuées par le modèle.

3.4.2 Paramètres des MFCC

Les paramètres spécifiques utilisés pour l'extraction des MFCC sont :

- Nombre de coefficients cepstraux : 20
- Coefficient de départ : Nous avons testé notre modèle sous des réglages de 0, 1, et 2 premiers coefficients ignorés. La valeur 1 a été conservée (1er coefficient ignoré ; voir 2.1.2.7.2).
- Paramètres de la transformation mel (*melkwargs*) :
  - Nombre de points FFT : 512
  - Longueur de saut : 256
  - Fenêtre : Hamming
  - Nombre de filtres mel : 32

Ces paramètres sont choisis pour capturer les caractéristiques spectrales pertinentes des signaux audio tout en restant compatibles avec les contraintes computationnelles.

3.4.3 Dimensions de la matrice MFCC

Pour des fichiers sonores d'une durée de 2 secondes sous fréquence d'échantillonnage de 22 050 Hz (= 44 100 échantillons), le nombre d'images spectrales (frames) correspond à :

Équation 3-1 : Nombre de frames

$$Nombre\ de\ frames = \left[\frac{44100\ (n.échantillons) - 512\ (FFT)}{256\ (saut)}\right] + 1 = 171$$

Le nombre de coefficients est de :

$$20\ (n.coefficients\ MFCC) - 1\ (1er\ coefficient\ ignoré) = 19$$

Les dimensions de la matrice sont donc :

$$Dimensions = 19 \times 171$$

3.5 Architecture du modèle

3.5.1 Présentation du modèle SEnv-Net

SEnv-Net est un réseau neuronal convolutionnel conçu pour la classification des sons environnementaux. Ses caractéristiques principales sont :

- Couches convolutionnelles : Pour extraire des caractéristiques locales des MFCC ;
- Couches de pooling : Pour réduire la dimensionnalité tout en conservant les informations essentielles ;
- Couches entièrement connectées : Pour agréger les caractéristiques extraites et effectuer la classification.

3.5.2 Adaptation en réseau siamois

Afin d'adapter le modèle SEnv-Net à notre problématique d'identification de sources sonores principales dans un mélange complexe, nous avons transformé l'architecture originale en un SCNN. Celui-ci est composé de deux sous-réseaux identiques partageant les mêmes poids, ce qui permet de garantir que les caractéristiques extraites des différentes entrées sont comparables sur un même plan de représentation.

Chaque sous-réseau applique une série de convolutions et de couches entièrement connectées à ses entrées (MFCC), et la distance entre les sorties des jumeaux est utilisée pour entraîner le modèle avec une fonction de perte par triplet.

Le pseudocode suivant résume l'architecture d'un sous-réseau. :

```
Encadré 4. Pseudocode de l'architecture du sous-réseau convolutionnel
(Code Python correspondant : annexe A, section A.4)

CLASSE Réseau_Siamois

    INITIALISATION(input_shape)
        Définir un module convolutionnel séquentiel :
```

```
            - Convolution 2D (1 → 32) + ReLU + MaxPool
            - Convolution 2D (32 → 64) + ReLU + MaxPool
            - Convolution 2D (64 → 128) + ReLU

        Calculer la taille de sortie après les convolutions (avec un
tenseur fictif)

        Définir les couches entièrement connectées :
            - Flatten
            - Linéaire vers 128 unités + ReLU
            - Linéaire vers 1 sortie

    FONCTION forward(x)
        Ajouter une dimension pour le canal
        Appliquer le bloc convolutionnel
        Appliquer le bloc de couches entièrement connectées
        Retourner la sortie
```

Ce réseau applique successivement des filtres convolutifs 2D suivis de fonctions d'activation ReLU et de couches de max pooling, ce qui permet de réduire la dimension tout en conservant les structures discriminantes. L'architecture est conçue pour traiter des entrées monocanal (i.e., un seul spectre MFCC à la fois), avec une sortie finale sous forme de score scalaire (de dimension 1), qui représente une mesure de similarité projetée dans un espace latent.

Le même sous-réseau est utilisé pour encoder l'ancre, le jumeau positif et le jumeau négatif. Cette configuration permet de comparer les sorties par calcul de distances, et ainsi d'entraîner le modèle à rapprocher les paires similaires (ancre et jumeau positif) et à éloigner les paires dissemblables (ancre et jumeau négatif), conformément à la logique de la fonction de perte par triplet.

3.5.3 Fonction de perte par triplet

Nous utilisons la fonction de perte par triplet pour entraîner le réseau siamois. Cette approche permet d'apprendre une structure de similarité adaptée au problème, sans passer par une classification explicite. Le pseudocode ci-dessous illustre le calcul de cette fonction :

```
Encadré 5. Pseudocode de la fonction de perte par triplet
(Code Python correspondant : annexe A, section A.5)
```

```
FONCTION Perte_par_triplet(ancre, positif, négatif, marge)
    Calculer la distance positive : distance entre ancre et positif
    Calculer la distance négative : distance entre ancre et négatif
    Calculer la perte : distance positive - distance négative + marge
    Appliquer ReLU pour conserver uniquement les pertes positives
    Retourner la moyenne des pertes
FIN FONCTION
```

Cette fonction correspond à :

Équation 3-2 : Fonction de perte par triplet telle qu'implémentée

$$\mathcal{L} = max(0, \| f(a) - f(p) \|_2^2 - \| f(a) - f(n) \|_2^2 + \alpha)$$

où α est la marge choisie.

Nous utilisons les distances euclidiennes au carré entre les représentations vectorielles des données, ce qui présente le double avantage d'une plus grande efficacité computationnelle en plus d'amplifier les différences, ce qui peut aider le modèle à apprendre plus rapidement. La marge α doit cependant être réglée avec attention afin d'éviter que la fonction de perte soit trop sensible aux données aberrantes. Ce paramètre est réglé à la valeur 0,5 dans le code d'entraînement (section 3.6.3)[108].

3.6 Procédure d'entraînement

3.6.1 Génération des triplets

Les triplets utilisés pour l'entraînement sont générés à partir des données prétraitées en utilisant le script décrit en 3.3.3.1. Les étapes clés sont :

- Ancre (*A*) : Segment du signal de référence (captation rapprochée) ;
- Positif (*P*) : Simulation avec le son ancre comme source dominante ;

[108] Voir aussi 2.1.3.7 pour la description du paramètre *marge (margin)* dans la fonction de perte par triplet.

- Négatif (*N*) : Simulation avec une autre source sonore dominante, mais comprenant le son ancre en rôle secondaire (intensité dBA plus faible que le son dominant).

Les chemins vers les fichiers sonores constituant les triplets sont stockés dans un fichier CSV qui est ensuite utilisé par le `DataLoader` de PyTorch pour charger les données.

3.6.2 Chargement des données

Les données sont préparées sous forme de triplets à l'aide de la classe personnalisée `TripletDataset`, présentée précédemment. Cette classe applique à chaque élément du triplet l'extraction des MFCC, puis renvoie les trois représentations prêtes à être traitées par le modèle siamois.

Le chargement est effectué à l'aide d'un `DataLoader` de PyTorch, qui permet une lecture par lot, un mélange aléatoire des échantillons, et une parallélisation des accès disques.

Le pseudocode suivant résume la logique utilisée :

```
Encadré 6. Pseudocode du chargement des données
(Code Python correspondant : annexe A, section A.6)

Initialiser le dataset d'entraînement avec les paramètres requis :
    - Données tabulaires (chemins des fichiers)
    - Chemin de base des fichiers audio
    - Paramètres MFCC (nombre de coefficients, paramètres Mel, etc.)

Créer un DataLoader avec :
    - Le dataset préparé
    - La taille de lot désirée
    - L'activation du mélange aléatoire des échantillons
    - Un nombre de processus pour le chargement parallèle
```

Cette étape assure une alimentation efficace du réseau pendant l'entraînement. La parallélisation via l'argument `num_workers` permet de limiter le goulot d'étranglement lié à l'accès aux fichiers audio. De plus, le mélange des échantillons (shuffle=True) garantit que le modèle ne voit pas les triplets dans un ordre prévisible, ce qui améliore la généralisation.

3.6.3 Entraînement du modèle

L'entraînement du réseau siamois a été réalisé à l'aide d'une fonction dédiée, intégrant un optimiseur Adam et un planificateur de taux d'apprentissage afin de favoriser une convergence stable et efficace. Les données d'entraînement et de validation sont parcourues par lots, et le modèle est mis à jour à chaque itération à partir de la perte calculée par la fonction de perte par triplet.

Le pseudocode suivant illustre le déroulement de l'entraînement :

Encadré 7. Pseudocode de la fonction d'entraînement du réseau siamois (Code Python correspondant : annexe A, section A.7)

```
FONCTION Entraîner_réseau_siamois(modèle, données, paramètres)
    Initialiser le taux d'apprentissage à 0,01
    Définir l'optimiseur Adam avec les paramètres du modèle
    Définir un planificateur de taux d'apprentissage (décroissance à
toutes les 5 époques)

    POUR chaque époque (jusqu'au maximum spécifié) :
        Mettre le modèle en mode entraînement
        POUR chaque lot dans les données d'entraînement :
            Transférer les données sur l'appareil (CPU ou GPU)
            Réinitialiser les gradients
            Calculer les sorties du modèle pour l'ancre, le positif et
le négatif
            Calculer la perte par triplet
            Propager l'erreur (backpropagation)
            Mettre à jour les poids avec l'optimiseur
        FIN POUR

        Mettre à jour le taux d'apprentissage via le planificateur
    FIN POUR
FIN FONCTION
```

Les hyperparamètres utilisés sont :

- Optimiseur : Adam avec un taux d'apprentissage $lr = 0{,}01$;
- Marge (α) : fixée à 0,5 ;
- Taille du lot : 512 ;
- Nombre d'époques (maximum) : 50.

3.6.3.1 Planification du taux d'apprentissage

Afin d'améliorer la stabilité de l'apprentissage et d'accroître la capacité de généralisation du modèle, un planificateur de type `StepLR` a été utilisé. Ce planificateur réduit le taux d'apprentissage par un facteur de 10 à toutes les 5 époques. Une telle stratégie permet :

- de commencer l'entraînement avec un taux relativement élevé, favorisant une convergence rapide dans les premières phases ;
- puis de réduire progressivement ce taux, afin de stabiliser la descente de gradient et de faciliter l'ajustement fin des paramètres dans les dernières époques.

Cette approche contribue à limiter les risques d'oscillation du modèle autour d'un minimum et à augmenter les chances d'atteindre une solution localement optimale.

3.6.4 Arrêt d'entraînement automatique

Pour éviter le surapprentissage et limiter le temps d'entraînement inutile, une procédure d'arrêt anticipé (*early stopping*) a été utilisée. Cette méthode interrompt l'entraînement si la performance du modèle sur l'ensemble de validation cesse de s'améliorer pendant un certain nombre d'itérations consécutives.

Le pseudocode suivant résume le fonctionnement du mécanisme :

Encadré 8. Pseudocode de la fonction d'arrêt d'entraînement automatique (Code Python correspondant : annexe A, section A.8)

```
CLASSE ArretAnticipe(patience, delta, chemin_sauvegarde)
    Initialiser les compteurs et le meilleur score

    METHODE __call__(val_loss, modèle)
        Si aucun meilleur score enregistré :
            Enregistrer score et sauvegarder le modèle
        Sinon si amélioration < delta :
            Incrémenter le compteur
            Si compteur atteint la patience :
                Déclencher l'arrêt
        Sinon :
            Enregistrer le nouveau meilleur score
            Sauvegarder le modèle
            Réinitialiser le compteur

    METHODE sauvegarder(val_loss, modèle)
```

```
        Sauvegarder les poids du modèle
        Mettre à jour la perte minimale observée
```

Procédure d'arrêt automatique :

1. Surveillance de la perte de validation : À chaque époque, la perte de validation est calculée parallèlement à la perte d'entraînement. Elle reflète la capacité du modèle à généraliser à des données non vues.

2. Critère d'arrêt : Si aucune amélioration significative de la perte de validation n'est observée pendant un certain nombre d'époques consécutives (défini par la patience), l'entraînement est interrompu.

Paramètres utilisés :

- Patience : 5 époques. Ce choix est cohérent avec le planificateur de taux d'apprentissage (section 3.6.3), dont le pas est également de 5 époques.
- Delta : 0,001, soit une amélioration minimale égale à marge / 500. Cette valeur permet de considérer comme significative toute réduction de perte supérieure à ce seuil, même minime, en tenant compte de la finesse requise par la tâche de comparaison de triplets.
- Sauvegarde du modèle : Le modèle est sauvegardé dès qu'une amélioration est détectée, et le meilleur état est restauré à la fin de l'entraînement, qu'il soit interrompu manuellement ou automatiquement.

3.7 Évaluation du modèle

3.7.1 Métriques utilisées

Pour évaluer les performances du modèle, nous avons utilisé différents ensembles de données et métriques :

- Courbes de perte : Analyse de la convergence du modèle en observant la perte pour les ensembles d'entraînement et de validation.
- Taux de précision (accuracy) : Proportion des bonnes identifications de la source dominante dans l'ensemble de test. La précision est définie comme suit :

Équation 3-3 : Calcul de la mesure de précision

$$Précision = \frac{vrais\ positifs}{vrais\ positifs + faux\ positifs}$$

- Mesure F1 (F1-score) : Harmonie entre la précision et le rappel pour évaluer l'équilibre entre ces deux mesures dans les prédictions. La mesure F1 est définie comme :

Équation 3-4 : Calcul de la mesure F1

$$F1 = 2 \times \frac{précision \times rappel}{précision + rappel}$$

Équation 3-5 : Calcul de la mesure de rappel

$$Rappel = \frac{vrais\ positifs}{vrais\ positifs + faux\ négatifs}$$

- Spécificité : Vérification de la détection des négatifs. Elle est définie comme suit :

Équation 3-6 : Calcul de la mesure de spécificité

$$Spécificité = \frac{vrais\ négatifs}{vrais\ négatifs + faux\ positifs}$$

L'ensemble de validation a été utilisé pendant l'entraînement pour ajuster les hyperparamètres et déclencher l'arrêt anticipé lorsque nécessaire. L'ensemble de test, resté entièrement non vu pendant l'entraînement, a ensuite permis une évaluation finale des performances du modèle.

Le pseudocode ci-dessous résume la logique générale de l'évaluation :

```
Encadré 9. Pseudocode de la procédure d'évaluation
(Code Python correspondant : annexe A, section A.9)

FONCTION Évaluer_modèle(modèle, ensemble_test, marge)
    Passer le modèle en mode évaluation
    Désactiver le calcul des gradients

    POUR chaque lot dans les données de test :
        Charger les triplets (ancre, positif, négatif)
        Obtenir les sorties du modèle pour chaque élément
        Calculer la perte par triplet
```

```
        Mesurer les distances entre les paires
        Enregistrer les décisions et résultats pour les métriques
    FIN POUR

    Calculer la précision, la mesure F1 et la spécificité
    Générer les courbes de perte et les visualisations
    Retourner les résultats d'évaluation
FIN FONCTION
```

Les performances finales du modèle, mesurées sur l'ensemble de test, sont présentées dans la prochaine section de ce mémoire (4.2).

3.8 Implémentation et environnement expérimental

3.8.1 Logiciels utilisés

Notre implémentation repose sur les logiciels et bibliothèques suivants :

- Python 3.9.16
- Bibliothèques principales :
    - Librosa : pour le traitement du signal audio, y compris le calcul des MFCC et des transformations spectrales.
    - Torchaudio : pour la transformation et le traitement des fichiers audio.
    - NumPy : pour les opérations matricielles et la gestion des données.
    - PyTorch : pour l'implémentation des réseaux de neurones et des méthodes d'optimisation.
    - Scikit-learn : pour les métriques et l'évaluation des modèles.
    - Matplotlib : pour la visualisation des résultats et des graphiques.

Toutes ces bibliothèques ont été installées et gérées via conda-forge, un environnement de gestion de paquets adapté aux besoins de développement scientifique et de machine learning.

3.8.2 Structure du code

Le code complet de l'implémentation est organisé de la manière suivante :

1. Préparation des données :

- *Prep.py* : Script pour la sélection et la segmentation des enregistrements en captation rapprochée.
- *Jumeaux.py* : Script pour la génération des jumeaux (ancre, positif, négatif).

2. Chargement des données :
   - `TripletDataset` : Classe pour charger les triplets et extraire les MFCC.
3. Définition du modèle :
   - `Siamese` : Classe définissant l'architecture du réseau siamois.
4. Entraînement et évaluation :
   - `train_siamese_network` : Fonction pour entraîner le modèle.
   - `evaluate_model` : Fonction pour évaluer les performances du modèle.

3.8.3 Matériel et ressources

Les expérimentations ont été réalisées sur un poste de travail équipé de :

- Processeur : Apple M1 Max (10 cœurs)
- Mémoire : 64 Go
- GPU : Apple M1 Max avec 32 cœurs
- Système d'exploitation : macOS

3.8.3.1 Exploitation du GPU avec Metal et MPS

Afin de maximiser la portabilité et les performances du code, nous avons conçu notre système pour qu'il puisse s'exécuter automatiquement sur le meilleur périphérique disponible, qu'il s'agisse :

- d'un GPU NVIDIA avec support CUDA,
- d'un GPU Apple Silicon via l'API Metal et son interface torch.backends.mps,
- ou d'un CPU, si aucun GPU n'est accessible.

Ce comportement est géré dynamiquement à l'aide d'une fonction qui interroge les capacités du système au moment de l'exécution. Le code Python de cette fonction est présenté à l'annexe A, section A.10.

Pour accélérer l'entraînement de notre modèle sur l'Apple M1 Max, nous avons utilisé le backend MPS (Metal Performance Shaders) intégré à PyTorch. L'utilisation du backend MPS améliore significativement les performances de calcul sur les GPU d'Apple Silicon, en particulier pour les opérations intensives telles que les convolutions et la gestion de grandes quantités de données.

Afin de profiter pleinement des 10 cœurs haute performance du processeur Apple M1 Max, le paramètre *num_workers* du *DataLoader* a été ajusté. Ce paramètre contrôle le nombre de sous-processus utilisés pour le chargement parallèle des données, permettant ainsi une gestion plus efficace du pipeline d'entrée/sortie. En augmentant ce nombre, nous avons pu paralléliser le chargement des données, réduire le temps d'attente du GPU pour les données, et améliorer significativement la vitesse d'entraînement du modèle.

3.9 Conclusion de la méthodologie

Dans ce chapitre, nous avons présenté les étapes méthodologiques clés permettant la mise en œuvre d'un système d'identification des sources sonores principales, avec recours à des simulations pour remplacer les captations éloignées réelles. Nous avons intégré des extraits de code pertinents pour illustrer l'application des concepts théoriques à notre solution.

La préparation des données a joué un rôle essentiel en générant un jeu de données simulé, ajusté à notre problématique. L'utilisation du réseau neuronal siamois, associé à la fonction de perte par triplet, s'est montrée efficace pour différencier les sources sonores dans des environnements complexes.

Le chapitre suivant présentera et analysera les résultats obtenus, tout en évaluant les performances de notre modèle.

* * *

CHAPITRE 4 : RÉSULTATS ET DISCUSSION

Les résultats présentés dans ce chapitre ont pour but d'évaluer la faisabilité de la méthode proposée dans le cadre d'une preuve de concept exploratoire. L'objectif n'est ni de comparer plusieurs approches, ni de valider un déploiement opérationnel, mais bien de vérifier si le modèle siamois développé peut, en contexte simulé, apprendre à distinguer la source sonore dominante à partir d'un signal complexe.

Nous présentons ici les performances du modèle SCNN dans des environnements synthétiques, à travers une analyse des résultats obtenus sous différents niveaux de S/B. Ce S/B est défini comme le rapport, dans le mélange simulé, entre le signal principal (le son ancre A) et le bruit total, composé de deux sons aléatoires et d'un bruit rose.

En faisant varier les valeurs de S/B, nous avons pu simuler des environnements plus ou moins favorables à l'identification du son dominant, et ainsi tester la capacité du modèle à généraliser dans des conditions acoustiques changeantes.

4.1 Performances globales du modèle selon le S/B

Les graphiques de la Figure 4.1 présentent les performances du modèle en fonction du S/B. Les nuages de points montrent les valeurs individuelles obtenues lors des 12 mesures effectuées pour chaque S/B, tandis que la courbe lissée relie les moyennes des mesures pour chaque condition. Les performances du modèle ont été évaluées en fonction de trois métriques principales : la précision, la mesure F1 et la spécificité. Les résultats sont présentés selon les S/B de 0 dB, 3 dB, 6 dB, 10 dB, et 13 dB.

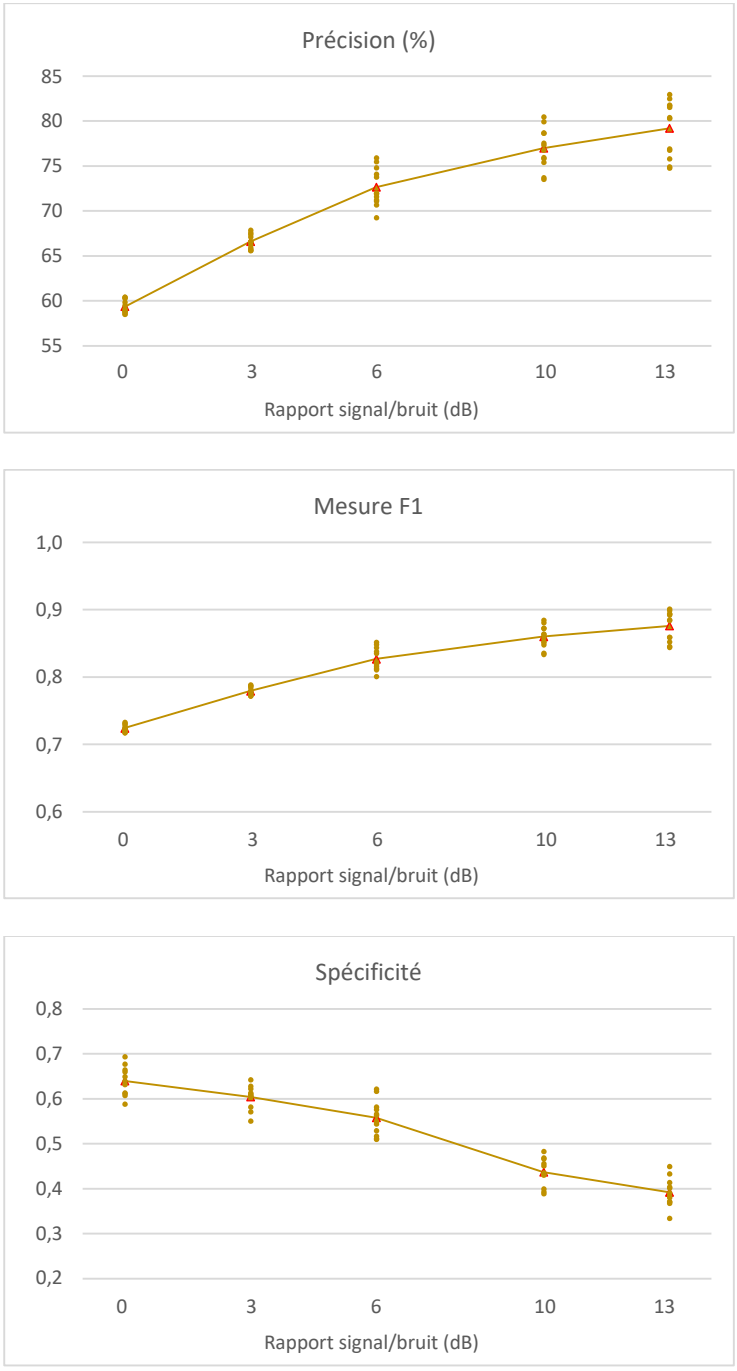


Figure 4.1 : Performances du modèle sous différents S/B.

4.1.1 Précision

La précision est la proportion de prédictions positives qui sont correctes, c'est-à-dire la capacité du modèle à limiter les faux positifs. Elle mesure l'exactitude des prédictions positives du modèle.

Comme le montre la Figure 4.1 (Précision), une amélioration continue de la précision est observée avec l'augmentation du S/B. À 0 dB, la précision est d'environ 60 %, ce qui montre que le modèle peut identifier correctement le son ancre dans des conditions de bruit intense. Cette performance, bien que modeste, reste supérieure à un choix aléatoire (50 %). À partir de 6 dB, la précision augmente sensiblement, atteignant près de 80 % à 13 dB. Cette tendance confirme que plus le son ancre se distingue du bruit, plus le modèle est capable de le détecter correctement.

4.1.2 Mesure F1

La mesure F1 combine la précision et le rappel (ou sensibilité) pour offrir une mesure harmonique qui équilibre les faux positifs et les faux négatifs.

Bien que la mesure F1 soit traditionnellement utilisée dans des tâches de classification, elle reste utile dans le cadre d'un réseau siamois pour évaluer la capacité du modèle à différencier correctement les paires similaires (positives) des paires différentes (négatives), même si l'objectif principal n'est pas une classification stricte.

L'usage de la mesure F1 dans cette étude permet donc d'évaluer l'équilibre entre deux objectifs :

- Précision : Le modèle doit éviter de considérer à tort les jumeaux négatifs contenant un son ancre dilué comme similaires aux jumeaux positifs, ce qui pourrait augmenter les faux positifs.
- Rappel : Le modèle doit être capable de reconnaître correctement les jumeaux positifs où le son ancre est dominant, même dans des conditions de bruit.

Dans le cadre de notre expérimentation, la mesure F1 (Figure 4.1 : *Mesure F1*) suit une trajectoire similaire à celle de la précision, avec une augmentation progressive de 0,7 à

0 dB à environ 0,9 à 13 dB. Cela reflète une amélioration simultanée de la précision et du rappel. À mesure que le S/B augmente, le modèle devient plus équilibré dans sa capacité à identifier correctement les sons ancres tout en limitant les faux négatifs. Cependant, la dispersion des valeurs individuelles indique que, malgré une amélioration des performances moyennes, la variabilité des résultats augmente également avec le S/B, suggérant une diminution de la stabilité du modèle dans certaines conditions.

4.1.3 Spécificité

La spécificité mesure la capacité du modèle à éviter les faux positifs, en se concentrant sur la proportion de vrais négatifs correctement identifiés. Elle évalue la capacité du modèle à ignorer les éléments qui ne devraient pas être classés comme positifs. Dans le cas de notre expérimentation, il s'agira des jumeaux négatifs, où le son ancre est présent en bruit.

La spécificité (Figure 4.1 : Spécificité) montre une tendance à la baisse avec l'augmentation du S/B. Elle passe de 0,6 pour 0 dB à environ 0,3 pour 13 dB. Cela peut sembler contre-intuitif, car un S/B plus élevé devrait théoriquement permettre de mieux différencier les sons. En effet, dans cette étude, où le son ancre est également présent dans les jumeaux négatifs en tant qu'élément du bruit, l'augmentation du S/B réduit l'intensité du son ancre dans les jumeaux négatifs. Néanmoins, le modèle semble parvenir à détecter des traces résiduelles de ce son. Cette tendance à surclasser les jumeaux négatifs contenant un son ancre faible comme positifs explique probablement la diminution de la spécificité. Ce comportement pourrait refléter un excès de confiance du modèle dans l'identification des sons dominants dans des conditions où le bruit, dont le signal ancre fait partie, est faiblement présent.

4.2 Convergence du modèle pendant l'entraînement

La Figure 4.2 présente l'évolution de la perte triplet (*Triplet Loss*) pendant l'entraînement du modèle pour trois niveaux de S/B : 0 dB, 6 dB et 13 dB. Pour chaque niveau de S/B, trois courbes sont tracées :

- Bleu : perte moyenne sur l'ensemble d'entraînement ;
- Orange : perte moyenne sur l'ensemble de validation ;
- Vert : perte moyenne sur l'ensemble de test.

Les traits utilisés permettent de distinguer les niveaux de S/B :

- Ligne pointillée pour 0 dB,
- Ligne en tirets pour 6 dB,
- Ligne pleine pour 13 dB.

Chaque courbe représente la moyenne des 12 entraînements réalisés pour le niveau de S/B correspondant.

Pour un S/B de 0 dB (lignes pointillées), les courbes montrent une convergence lente, avec des pertes stables atteintes autour de l'époque 10. Les pertes d'entraînement (ligne bleue) sont les plus basses, mais les pertes de validation et de test (orange et verte) restent significativement plus élevées, indiquant la difficulté du modèle à généraliser dans cet environnement très bruité. Les performances finales demeurent faibles, reflétant l'impact important du bruit qui masque le son ancre.

Avec un S/B de 6 dB (lignes en tirets), les courbes suivent une trajectoire similaire, mais atteignent des valeurs de perte finales nettement inférieures comparativement au S/B de 0 dB. Cette amélioration montre que la distinction accrue entre le son ancre et le bruit à ce niveau de S/B permet au modèle d'apprendre de manière plus efficace. Cependant, les écarts relatifs entre les courbes d'entraînement (bleue), de validation (orange) et de test (verte) restent comparables à ceux observés pour 0 dB, suggérant que la généralisation du modèle ne s'améliore que marginalement.

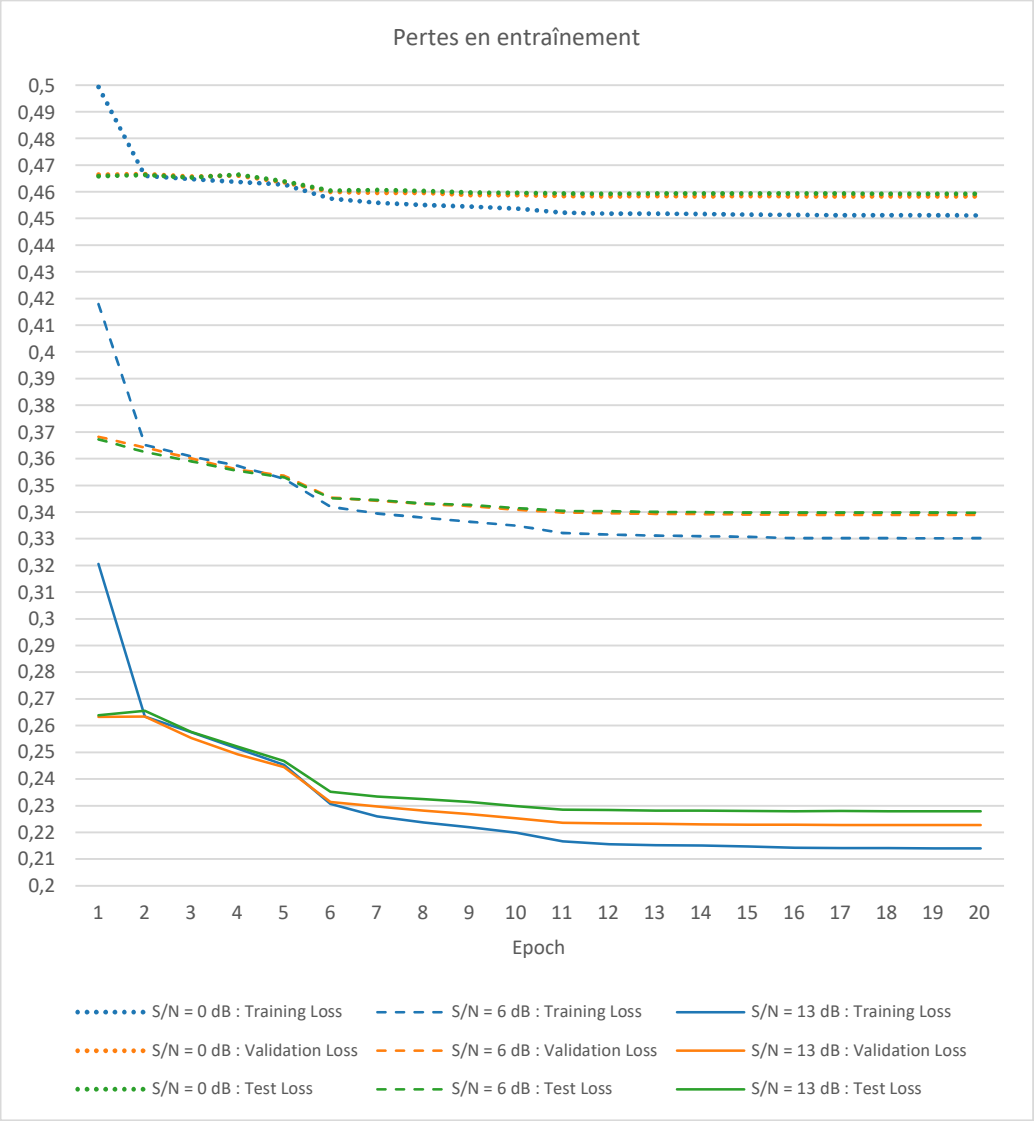


Figure 4.2 : Convergence de la perte triplet (Triplet Loss) pour les niveaux de S/B de 0 dB (pointillés), 6 dB (tirets) et 13 dB (pleins), selon les ensembles d'entraînement (bleu), de validation (orange) et de test (vert).

Pour un S/B de 13 dB (lignes pleines), les performances atteignent un niveau nettement supérieur. Les pertes d'entraînement (bleue), de validation (orange) et de test (verte) convergent rapidement vers des valeurs faibles dès l'époque 5 à 7. Ces résultats reflètent une meilleure séparation entre le son ancre et le bruit dans cet environnement, facilitant

l'apprentissage et la généralisation. Cependant, des écarts entre les courbes subsistent, ce qui suggère que le modèle continue d'être influencé par la présence résiduelle du son ancre dans les jumeaux négatifs, même à S/B élevé.

4.3 Conclusion de la discussion sur les résultats

Ce chapitre a présenté les performances de notre modèle SCNN dans des environnements simulés sous différents S/B, en mettant l'accent sur trois métriques principales : la précision, la mesure F1, et la spécificité.

Les résultats montrent une amélioration nette des performances du modèle à mesure que le S/B augmente, en particulier pour la précision et la mesure F1. Le modèle parvient à mieux identifier la source sonore dominante lorsque celle-ci est plus distincte du bruit. À un S/B élevé (10 à 13 dB), la précision atteint environ 80 %, et la mesure F1 suit une tendance similaire, atteignant presque 0,9, ce qui reflète un bon équilibre entre la capacité du modèle à identifier correctement les sons ancres et à limiter les faux négatifs.

Nous observons toutefois une baisse progressive de la spécificité avec l'augmentation du S/B, ce qui suggère un affaiblissement de la performance d'identification des faux positifs. Dans cette étude, le son ancre est toujours présent dans les jumeaux négatifs, bien que sous une forme plus faible à des S/B élevés. Ce phénomène peut expliquer pourquoi le modèle parvient encore à détecter des traces résiduelles du son ancre dans le bruit. Le modèle montre ainsi une difficulté croissante à ignorer complètement le son ancre lorsque celui-ci est plus subtilement présent dans les jumeaux négatifs, d'autant plus lorsque le S/B est élevé.

L'analyse des courbes de perte a montré que le modèle converge rapidement à des S/B élevés, avec des pertes faibles atteintes dès les premières époques. Cependant, la différence entre les courbes de validation et de test observée à 13 dB révèle une généralisation imparfaite, probablement attribuable à la présence résiduelle du son ancre dans les jumeaux négatifs. Ainsi, bien que le modèle apprenne efficacement dans des

conditions de S/B élevé, il reste des difficultés à généraliser correctement lorsque le son ancre est plus faiblement présent dans le bruit.

Un autre facteur important à prendre en compte est la diversité des types de sons utilisée lors de l'entraînement. Contrairement à des approches reposant sur un nombre limité de classes, l'exposition du modèle à une variété de sons a probablement ajouté un degré de complexité supplémentaire, ce qui pourrait expliquer l'augmentation de la variabilité des performances et la baisse de spécificité sous des conditions de S/B élevés.

Ces résultats mettent en lumière les limites du modèle dans sa gestion des jumeaux négatifs, où le son ancre, bien que plus faible, reste toujours présent. Cependant, comme nous en discuterons dans la conclusion générale du mémoire, cette limitation théorique peut probablement être exploitée dans le cadre plus large de l'objectif de cette étude.

* * *

CHAPITRE 5 : CONCLUSION ET PERSPECTIVES

Ce mémoire a exploré l'identification des sources sonores principales dans des environnements complexes et bruyants à l'aide d'un modèle SCNN basé sur une fonction de perte par triplet. L'objectif était de proposer une solution robuste pour la gestion du bruit environnemental, en permettant de comparer des captations rapprochées à une captation éloignée complexe afin d'identifier la source sonore dominante. Cette méthode se distingue par son approche d'analyse comparative, différente de la classification classique, afin de cibler précisément la source responsable des dépassements sonores en zone éloignée.

Bien que les résultats obtenus sur des données simulées montrent une preuve de concept valide, il est important de souligner que toute conclusion définitive reste prématurée tant que le système n'aura pas été testé en situation réelle. La simulation a permis de démontrer que le cadre proposé fonctionne, mais l'évaluation finale du modèle dépendra de tests avec des sons distants réels plutôt que de simulations.

5.1 Synthèse des principaux résultats

Les résultats obtenus montrent que le modèle proposé est capable d'identifier la source sonore principale dans des environnements à différents niveaux de S/B. Les observations suivantes peuvent être faites :

5.1.1 Capacité minimale à identifier la source principale à des S/B très faibles

Le modèle a montré sa capacité à identifier la source principale même dans des environnements à S/B très faibles (0 à 3 dB), avec des taux de précision oscillant entre 60 % et 70 %. Bien que ces résultats reflètent une capacité de détection minimale, ils restent encourageants, car ils dépassent nettement les performances attendues par hasard. En outre, aucun résultat significatif ne démontre de différence dans la capacité du modèle à généraliser en fonction du S/B, mais cette performance minimale constitue un bon indicateur de la robustesse globale du modèle.

5.1.2 Amélioration de la précision avec l'augmentation du S/B

À mesure que le S/B augmente, le modèle montre une amélioration progressive sur le plan de la précision et de la mesure F1, atteignant près de 80 % de précision et une mesure F1 de 0,9 pour un S/B de 13 dB. Ces résultats suggèrent que le modèle est capable d'identifier la source dominante lorsque celle-ci est clairement distincte du bruit.

5.1.3 Baisse progressive de la spécificité

En revanche, la spécificité diminue avec l'augmentation du S/B, ce qui indique une tendance du modèle à générer davantage de faux positifs. Cela est probablement dû à la présence continue du son ancre dans les jumeaux négatifs, qui, bien que plus faible à S/B élevé, reste détectable par le modèle comme une source dominante potentielle.

5.2 Contributions de la recherche

La principale contribution de cette recherche est la démonstration de la capacité du modèle à identifier les vrais positifs dans des environnements complexes. Toutefois, la gestion des faux positifs demeure un défi. Cela provient probablement du fait que le son ancre, présent dans les jumeaux négatifs en tant que bruit résiduel, est parfois confondu avec la source dominante. Cette complexité met en lumière une difficulté fondamentale dans la gestion des faux positifs, qui devra être abordée dans les travaux futurs.

Cependant, cette limitation pourrait devenir une opportunité en contexte pratique. En effet, la mesure de similitude utilisée par le modèle pourrait être interprétée comme un indicateur de la présence de sources secondaires, même si celles-ci ne sont pas dominantes. Un degré de similitude élevé entre un jumeau négatif et la captation éloignée pourrait signaler qu'une source sonore reste contributive au bruit total sans pour autant en être la source principale. Cette approche permettrait de réévaluer les faux positifs, non comme des erreurs, mais comme des indicateurs de responsabilité partagée dans la nuisance sonore. Ce mécanisme devra cependant être validé en situation réelle.

5.3 Recommandations pour les recherches futures

Bien que ce travail ait démontré la faisabilité du modèle dans des environnements simulés, plusieurs pistes d'amélioration peuvent être envisagées :

- Validation en situation réelle : Tester le modèle avec des sons distants réels dans un environnement portuaire ou industriel est une priorité afin de valider sa capacité à généraliser et à gérer la complexité des sons. Cela permettra également d'évaluer son efficacité face à la diversité des bruits rencontrés en pratique.

- Mesure de similitude pour hiérarchiser les sources contributives : Tester et valider la capacité de la mesure de similitude à identifier et hiérarchiser les sources contributives permettrait d'affiner la gestion des alertes et d'éviter de traiter les faux positifs comme des erreurs absolues.

- Exploration de l'impact de la diversité sonore : Analyser l'impact de différents sous-groupes de sons réels sur les performances du modèle permettrait de mieux comprendre la façon dont la diversité acoustique influence l'identification des sources. Cela aiderait également à ajuster l'entraînement en fonction des spécificités de l'environnement étudié.

- Utilisation de Transformers : La mise en œuvre de modèles Transformers, réputés pour leur capacité à modéliser des relations temporelles complexes, pourrait apporter des gains significatifs dans la généralisation, surtout dans des environnements sonores dynamiques.

- Application des techniques de BSS : Intégrer des techniques de séparation des sources pourrait améliorer la clarté du signal en permettant au modèle de se concentrer sur des sources plus nettes, réduisant ainsi l'ambiguïté liée à la superposition des sons.

- Réglage du paramètre de marge de la fonction de perte par triplet : Ajuster ce paramètre pourrait aider à réduire le taux de faux positifs, en augmentant la capacité du modèle à différencier plus efficacement les jumeaux positifs et négatifs.

- Exploration de modèles légers pour les réseaux de capteurs à faible coût : Afin de permettre une utilisation du modèle dans le cadre de systèmes déployés en réseau de capteurs à faible coût, il pourrait être nécessaire d'alléger l'architecture actuelle. L'utilisation de modèles comme SqueezeNet 18, optimisés pour une classification rapide des événements sonores avec des capacités limitées en calcul et en stockage, pourra être vérifiée.

5.4 Perspectives pour l'application réelle

Les résultats présentés dans ce mémoire ne doivent pas être interprétés comme une évaluation définitive des performances du modèle en contexte réel, mais bien comme une validation initiale du principe. En effet, la présente recherche vise à établir la plausibilité d'une approche comparative par réseau siamois dans le contexte acoustique simulé — une étape préliminaire avant toute expérimentation en conditions de terrain.

Dans le contexte de la collaboration avec l'Administration portuaire de Trois-Rivières et Logistec, Services maritimes, la prochaine étape clé sera donc de valider ce modèle dans un cadre pratique. En adaptant le modèle pour fournir en temps réel des informations sur la contribution des sources sonores aux dépassements sonores, il sera possible d'aider les opérateurs à ajuster leurs pratiques afin de réduire la nuisance tout en favorisant le maintien du niveau d'activité. Ce système pourra non seulement être utilisé pour ajuster les horaires ou la position des activités bruyantes, mais également pour mieux planifier ces activités en fonction des conditions météorologiques ou de l'intensité du trafic portuaire.

Ce modèle présente également un potentiel d'utilisation dans d'autres domaines, comme la gestion proactive des nuisances dans des environnements urbains ou industriels complexes. Il pourrait être utilisé pour identifier les corridors sonores, ou encore dans des litiges liés à la pollution sonore, permettant d'établir clairement la responsabilité de chaque source contributive.

5.5 Conclusion générale

Cette étude a démontré qu'il est possible d'utiliser un réseau siamois à fonction de perte par triplet pour identifier des sources sonores dominantes dans des environnements complexes. Bien que les résultats obtenus soient prometteurs, ils doivent être validés dans des conditions réelles avant d'être appliqués à grande échelle. Les pistes d'amélioration suggérées, notamment l'ajustement de la fonction de perte et l'introduction de modèles plus sophistiqués, ouvrent des perspectives intéressantes pour renforcer la robustesse et la précision de l'identification sonore.

L'application pratique de ce modèle pourrait avoir un impact significatif dans la gestion des nuisances sonores, permettant d'agir de manière proactive pour limiter les dépassements sonores tout en préservant le niveau d'activité des environnements industriels.

* * * * *

RÉFÉRENCES

Abayomi-Alli, O. O., Damaševičius, R., Qazi, A., Adedoyin-Olowe, M., & Misra, S. (2022). Data augmentation and deep learning methods in sound classification: A systematic review. *Electronics*, *11*(22), 3795.

Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM transactions on audio, speech, and language processing*, *22*(10), 1533-1545.

Abeßer, J. (2020). A review of deep learning based methods for acoustic scene classification. *Applied Sciences*, *10*(6), 2020.

Abu, A., Miskovic, N., Chebotar, O., Cukrov, N., & Diamant, R. (2024). Multiple Mobile Target Detection and Tracking in Active Sonar Array Using a Track-Before-Detect Approach. *arXiv preprint arXiv:2404.10316*.

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., & Vijayanarasimhan, S. (2016). Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*.

Al-Hattab, Y. A., Zaki, H. F., & Shafie, A. A. (2021). Rethinking environmental sound classification using convolutional neural networks: optimized parameter tuning of single feature extraction. *Neural Computing and Applications*, *33*(21), 14495-14506. https://doi.org/10.1007/s00521-021-06091-7

Alías, F., Carrié, J. C., & Sevillano, X. (2016). A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds. *Applied Sciences*, *6*, 143. https://doi.org/10.3390/app6050143

Aljubayri, I. (2023). Comparative Analysis of Different Sampling Rates on Environmental Sound Classification Using the Urbansound8k Dataset. *Journal of Computer and Communications*, *11*(6), 19-27.

Allen, J. B., & Rabiner, L. R. (1977). A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE*, *65*(11), 1558-1564.

Alluri, V., & Toiviainen, P. (2010). Exploring perceptual and acoustical correlates of polyphonic timbre. *Music Perception*, *27*(3), 223-242.

Alonso-Jiménez, P., Serra, X., & Bogdanov, D. (2023). Efficient supervised training of audio transformers for music representation learning. *arXiv preprint arXiv:2309.16418*.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., & Chen, G. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. International conference on machine learning,

Aptr. (2021). *À propos - Port de Trois-Rivières*. Consulté le 17 décembre 2021 à l'adresse https://porttr.com/administration-portuaire/a-propos/

Arora, S. (2017). Audio signal noise reduction using low pass filter. *Int. J. Comput. Sci. Eng.*, *4*(11), 1-7.

Ashraf, M., Abid, F., Din, I. U., Rasheed, J., Yesiltepe, M., Yeo, S. F., & Ersoy, M. T. (2023). A hybrid cnn and rnn variant model for music classification. *Applied Sciences*, *13*(3), 1476.

Aytar, Y., Vondrick, C., & Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems*, *29*.

Bar-Shalom, Y., Willett, P. K., & Tian, X. (2011). *Tracking and data fusion* (Vol. 11). YBS publishing Storrs, CT, USA:.

Barchiesi, D., Giannoulis, D., Stowell, D., & Plumbley, M. D. (2015). Acoustic scene classification: Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, *32*(3), 16-34.

Benetos, E., Stowell, D., & Plumbley, M. D. (2018). Approaches to Complex Sound Scene Analysis. In T. Virtanen, M. D. Plumbley, & D. P. W. Ellis (Eds.), *Computational analysis of sound scenes and events* (pp. 181-204). Springer.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.

Bianco, M. J., Gerstoft, P., Traer, J., Ozanich, E., Roch, M. A., Gannot, S., & Deledalle, C.-A. (2019). Machine learning in acoustics: Theory and applications. *The Journal of the Acoustical Society of America*, *146*(5), 3590-3628.

Bittner, M. (2020). *SoundFile: An audio library based on libsndfile, CFFI and NumPy*. GitHub repository. https://github.com/bastibe/python-soundfile

Boddapati, V., Petef, A., Rasmusson, J., & Lundberg, L. (2017). Classifying environmental sounds using image recognition networks. *Procedia computer science*, *112*, 2048-2056.

Bonet-Solà, D., & Alsina-Pagès, R. M. (2021). A comparative survey of feature extraction and machine learning methods in diverse acoustic environments. *Sensors*, *21*(1274). https://doi.org/10.3390/s21041274

Bonet-Solà, D., Vidaña-Vila, E., & Alsina-Pagès, R. M. (2023a). Analysis and Acoustic Event Classification of Environmental Data Collected in a Citizen Science Project. *International Journal of Environmental Research and Public Health*, *20*(4), 3683.

Bonet-Solà, D., Vidaña-Vila, E., & Alsina-Pagès, R. M. (2023b). Prediction of the acoustic comfort of a dwelling based on automatic sound event detection. *Noise Mapping*, *10*(1), 20220177.

Boualoulou, N., Belhoussine Drissi, T., & Nsiri, B. (2022). An intelligent approach based on the combination of the discrete wavelet transform, delta delta MFCC for Parkinson's disease diagnosis. *International Journal of Advanced Computer Science and Applications*, *13*(4).

Bracewell, R. N. (2000). *The Fourier Transform & Its Applications* (3 ed.). McGraw-Hill.

Brandstein, M., & Ward, D. (2013). *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, *6*.

Burred, J. J. (2009). From sparse models to timbre learning: new methods for musical source separation.

Chen, J. C., Yao, K., & Hudson, R. E. (2002). Source localization and beamforming. *IEEE Signal Processing Magazine*, *19*(2), 30-39.

Chen, K., & Salman, A. (2011). Extracting speaker-specific information with a regularized siamese deep network. *Advances in neural information processing systems*, *24*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choe, H. C., Karlsen, R. E., Gerhart, G. R., & Meitzler, T. J. (1996). Wavelet-based ground vehicle recognition using acoustic signals. Wavelet Applications III,

Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Convolutional recurrent neural networks for music classification. 2017 IEEE International conference on acoustics, speech and signal processing (ICASSP),

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05),

Claudi Socoró, J., Alías, F., & Alsina-Pagès, R. M. (2017). An Anomalous Noise Events Detector for Dynamic Road Traffic Noise Mapping in Real-Life Urban and Suburban Environments. *Sensors (14248220)*, *17*(10), 2323. https://doi.org/10.3390/s17102323

Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, *19*(90), 297-301.

Cousson, R., Leclere, Q., Pallas, M.-A., & Berengier, M. (2019). A time domain CLEAN approach for the identification of acoustic moving sources. *Journal of sound and vibration*, *443*, 47-62.

Couvreur, C., Fontaine, V., Gaunard, P., & Mubikangiey, C. G. (1998). Automatic classification of environmental noise events by hidden Markov models. *Applied Acoustics*, *54*(3), 187-206.

Crocco, M., Cristani, M., Trucco, A., & Murino, V. (2016). Audio surveillance: A systematic review. *ACM Computing Surveys (CSUR)*, *48*(4), 1-46.

Curcuruto, S., Leo, A. d., & Fabozzi, C. (2000). The noise produced by harbor infrastructures. *The Journal of the Acoustical Society of America*, *108*(5), 2455-2455. https://doi.org/10.1121/1.4743047

Čurović, L., Jeram, S., Murovec, J., Novaković, T., Rupnik, K., & Prezelj, J. (2021). Impact of COVID-19 on environmental noise emitted from the port. *Science of The Total Environment*, *756*, 144147. https://doi.org/10.1016/j.scitotenv.2020.144147

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(4), 357-366. https://doi.org/10.1109/TASSP.1980.1163420

Defréville, B. (2005). *Caractérisation de la qualité sonore de l'environnement urbain: une approche physique et perceptive basée sur l'identification des sources sonores* Université de Cergy Pontoise].

Deng, L., & Li, X. (2013). Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(5), 1060-1089.

Deng, L., & Yu, D. (2014). Foundations and Trends in Signal Processing: DEEP LEARNING–Methods and Applications.

Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dieleman, S., & Schrauwen, B. (2014). End-to-end learning for music audio. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Droghini, D., Vesperini, F., Principi, E., Squartini, S., & Piazza, F. (2018). Few-shot siamese neural networks employing audio features for human-fall detection.

Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence,

El Ayadi, M., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern recognition*, *44*(3), 572-587.

Ellis, D. P. W. (1996). *Prediction-driven computational auditory scene analysis* Massachusetts Institute of Technology].

Eronen, A., & Klapuri, A. (2000). Musical instrument recognition using cepstral coefficients and temporal features. 2000 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 00CH37100),

Esmaeilpour, M., Cardinal, P., & Koerich, A. L. (2020). Unsupervised feature learning for environmental sound classification using weighted cycle-consistent generative adversarial network. *Applied Soft Computing*, *86*, 105912.

Essid, S., Parekh, S., Duong, N. Q., Serizel, R., Ozerov, A., Antonacci, F., & Sarti, A. (2018). Multiview approaches to event detection and scene analysis. In *Computational analysis of sound scenes and events* (pp. 243-276). Springer.

Fang, J., Yin, B., Ji, X., & Du, Z. (2021). *Environmental Sound Classification Method Based on Two-Stream Lightweight Convolutional Neural Network* [preprint].

Fedele, A., Guidotti, R., & Pedreschi, D. (2022). Explaining siamese networks in few-shot learning for audio data. International Conference on Discovery Science,

Fernandez-Grande, E. (2022). Four decades of near-field acoustic holography. *The Journal of the Acoustical Society of America*, *152*(1), R1-R2.

Foote, J. T. (1997). Content-based retrieval of music and audio. Multimedia storage and archiving systems II,

Gade, S., Hald, J., & Ginn, B. (2013). Noise source identification with increased spatial resolution. *Sound & Vibration*, *47*(4), 9-13.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., & Dahlgren, N. L. (1993). *Acoustic-Phonetic Continuous Speech Corpus* National Institute of Standards and Technology (NIST).

Geiger, J. T., Kneißl, M., Schuller, B. W., & Rigoll, G. (2014). Acoustic gait-based person identification using hidden Markov models. Proceedings of the 2014 workshop on mapping personality traits challenge and workshop,

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., & Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP),

Ginn, K. B., & Haddad, K. (2012). Noise source identification techniques: simple to advanced applications. *Acoustics 2012*.

Ginovart-Panisello, G. J., Vidaña-Vila, E., Caro-Via, S., Martínez-Suquía, C., Freixes, M., & Alsina-Pagès, R. M. (2021). Low-Cost WASN for Real-Time Soundmap Generation. *Engineering Proceedings*, *6*(1), 57. https://doi.org/10.3390/I3S2021Dresden-10162

Godsill, S. J., Cemgil, A. T., Févotte, C., & Wolfe, P. J. (2007). Bayesian computational methods for sparse audio and music processing. 2007 15th European Signal Processing Conference,

Gong, Y., Chung, Y.-A., & Glass, J. (2021). Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Google Research. (2021). YAMNet: A pretrained deep net for audio classification. In: https://github.com/tensorflow/models/tree/master/research/audioset/yamnet.

Goudarzi, A., Spehr, C., & Herbold, S. (2021). Automatic source localization and spectra generation from sparse beamforming maps. *The Journal of the Acoustical Society of America*, *150*(3), 1866-1882. https://doi.org/10.1121/10.0005885

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. 2013 IEEE international conference on acoustics, speech and signal processing,

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5-6), 602-610.

Hafiz, N. F., Mashohor, S., Shazril, M., Rasid, M. F. A., & Ali, A. (2023). Comparison of Mel Frequency Cepstral Coefficient (MFCC) and Mel Spectrogram Techniques to Classify Industrial Machine Sound. 2023 15th International Conference on Software, Knowledge, Information Management and Applications (SKIMA),

Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., Gulati, A., Pang, R., & Wu, Y. (2020). Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. *arXiv preprint arXiv:2005.03191*.

Han, Y., Kim, J., & Lee, K. (2016). Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM transactions on audio, speech, and language processing*, *25*(1), 208-221.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition,

Heilmann, G., Doebler, D., & Boeck, M. (2014). Exploring the limitations and expectations of sound source localization and visualization techniques. INTER-

NOISE and NOISE-CON Congress and Conference Proceedings, Institute of Noise Control Engineering,

Heittola, T., Çakır, E., & Virtanen, T. (2018). The Machine Learning Approach for Analysis of Sound Scenes and Events. In T. Virtanen, M. D. Plumbley, & D. P. W. Ellis (Eds.), *Computational analysis of sound scenes and events* (pp. 13-40). Springer.

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., & Seybold, B. (2017). CNN architectures for large-scale audio classification. 2017 ieee international conference on acoustics, speech and signal processing (icassp),

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., & Sainath, T. N. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82-97.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hou, J., Zeng, L., Zhao, D., & Zhong, Y. (2022). A review for the noise source identification methods based microphone array. *Journal of Vibroengineering*, *24*(5), 983-1001.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition,

Huang, P.-S., Kim, M., Hasegawa-Johnson, M., & Smaragdis, P. (2014). Singing-Voice Separation from Monaural Recordings using Deep Recurrent Neural Networks. ISMIR,

Huang, X. (2024). A comprehensive survey of digital watermarking techniques. *AIIT Bulletin*, 17. https://aiit.ac.jp/documents/jp/research_collab/research/bulletin/17th/017_huang.pdf

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Ibarra-Zarate, D., Tamayo-Pazos, O., & Vallejo-Guevara, A. (2019). Bearing fault diagnosis in rotating machinery based on cepstrum pre-whitening of vibration and acoustic emission. *The International Journal of Advanced Manufacturing Technology*, *104*, 4155-4168.

İnik, Ö. (2023). CNN hyper-parameter optimization for environmental sound classification. *Applied Acoustics*, *202*, 109168.

Jamal, N., Shanta, S., Mahmud, F., & Sha'abani, M. (2017). Automatic speech recognition (ASR) based approach for speech therapy of aphasic patients: A review. AIP Conference Proceedings,

Jekateryńczuk, G., & Piotrowski, Z. (2023). A Survey of Sound Source Localization and Detection Methods and Their Applications. *Sensors*, *24*(1), 68.

Jurafsky, D., & Martin, J. H. (2024). *Speech and Language Processing*. https://web.stanford.edu/~jurafsky/slp3/

Katti, A., & Sumana, M. (2022). Pipeline for pre-processing of audio data. In *IOT with Smart Systems: Proceedings of ICTIS 2022, Volume 2* (pp. 191-198). Springer.

Kinnunen, T., & Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, *52*(1), 12-40.

Kopp, S., & Wachsmuth, I. (2004). Synthesizing multimodal utterances for conversational agents. *Computer animation and virtual worlds*, *15*(1), 39-52.

Koutini, K., Schlüter, J., Eghbal-Zadeh, H., & Widmer, G. (2021). Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*.

Kragh, J. (2000). NORD 2000. State-of-the-art overview of the new NORDIC prediction methods for environmental noise.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Lange, A., Xu, R., Kaeding, M., Marx, S., & Ostermann, J. (2024). Acoustic Emission Detection in Noisy Environments using Linear Prediction.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Lee, J.-A., & Kwak, K.-C. (2023). Heart Sound Classification Using Wavelet Analysis Approaches and Ensemble of Deep Learning Models. *Applied Sciences*, *13*(21), 11942.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical systems and signal processing*, *138*, 106587.

Li, J., Deng, L., Haeb-Umbach, R., & Gong, Y. (2015). Robust automatic speech recognition: a bridge to practical applications.

Li, S.-H., Lin, B.-S., Tsai, C.-H., Yang, C.-T., & Lin, B.-S. (2017). Design of wearable breathing sound monitoring system for real-time wheeze detection. *Sensors*, *17*(1), 171.

Logan, B. (2000). Mel Frequency Cepstral Coefficients for Music Modeling. International Society for Music Information Retrieval Conference,

Luo, Y., Chen, Z., Hershey, J. R., Le Roux, J., & Mesgarani, N. (2017). Deep clustering and conventional networks for music separation: Stronger together. 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP),

Lyons, R. G. (1997). *Understanding digital signal processing, 3/E*. Pearson Education India.

Mackenzie, R., Gérard, A., & Pearson, M. (2015). Acoustic imaging and sound mapping of mining and transportation noise sources. *Canadian Acoustics - Acoustique Canadienne*, *43*(2), 38-39.

Mascarenhas, S., & Agarwal, M. (2021). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. 2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON),

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*, 115-133.

McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., & Nieto, O. (2015a). *librosa: Audio and Music Signal Analysis in Python*. GitHub repository. https://librosa.org/

McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., & Nieto, O. (2015b). librosa: Audio and music signal analysis in python. Proceedings of the 14th Python in Science Conference,

Merino-Martínez, R., Sijtsma, P., Snellen, M., Ahlefeldt, T., Antoni, J., Bahr, C. J., Blacodon, D., Ernst, D., Finez, A., & Funke, S. (2019). A review of acoustic imaging methods using phased microphone arrays. *CEAS Aeronautical Journal*, *10*(1), 197-230.

Mesaros, A., & Virtanen, T. (2010). Automatic recognition of lyrics in singing. *EURASIP Journal on Audio, Speech, and Music Processing*, *2010*, 1-11.

Meta AI. (2023). *torchaudio: an audio processing library built on PyTorch*. PyTorch.l'adresse https://pytorch.org/audio/

Minsky, M., & Papert, S. (1969). An introduction to computational geometry. *Cambridge tiass., HIT*, *479*(480), 104.

Mitrović, D., Zeppelzauer, M., & Breiteneder, C. (2010). Features for content-based audio retrieval. In *Advances in computers* (Vol. 78, pp. 71-150). Elsevier.

Moore, B. C., & Glasberg, B. R. (1996). A revision of Zwicker's loudness model. *Acta Acustica united with Acustica*, *82*(2), 335-345.

Moukas, P., Simson, J., & Norton-Wayne, L. (1982). Automatic identification of noise pollution sources. *IEEE Transactions on Systems, Man, and Cybernetics*, *12*(5), 622-634.

Mu, W., Yin, B., Huang, X., Xu, J., & Du, Z. (2021). Environmental sound classification using temporal-frequency attention based convolutional neural network. *Scientific Reports*, *11*(1), 21552.

Murovec, J., Prezelj, J., Čurović, L., & Novaković, T. (2018). Microphone array based automated environmental noise measurement system. *Applied Acoustics*, *141*, 106-114.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th international conference on machine learning (ICML-10),

Nelson, P. A., & Yoon, S.-H. (2000). Estimation of acoustic source strength by inverse methods: Part I, conditioning of the inverse problem. *Journal of sound and vibration*, *233*(4), 639-664.

Nigar, N. (2024). Speech Emotion Recognition Using CNN and Its Use Case in Digital Healthcare. *arXiv preprint arXiv:2406.10741*.

Noël, C. (2004). *Méthode temporelle d'identification de sources sonores bruyantes en milieu industriel*.

Nuha, H. H., & Absa, A. A. (2022). Noise Reduction and Speech Enhancement Using Wiener Filter. 2022 International Conference on Data Science and Its Applications (ICoDSA),

O'shaughnessy, D. (1987). *Speech communications: Human and machine (IEEE)*. Universities press.

O'shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Oppenheim, A. V., & Schafer, R. W. (1975). *Digital Signal Processing*. Prentice-Hall.

Oppenheim, A. V., & Schafer, R. W. (2009). *Discrete-Time Signal Processing*. Pearson.

Padois, T., Doutres, O., Sgard, F., & Bibliothèque numérique, c. (2018). *Développement d'une antenne microphonique intégrant un système optique pour identifier la position des sources sonores les plus bruyantes en milieu industriel*. Institut de recherche Robert-Sauvé en santé et en sécurité du travail. https://www.irsst.qc.ca/media/documents/PubIRSST/R-1038.pdf?v=2019-03-09

Pedersen, M. S., Larsen, J., Kjems, U., & Parra, L. C. (2007). A SURVEY OF CONVOLUTIVE BLIND SOURCE SEPARATION METHODS.

Pereira, A., Leclere, Q., & Antoni, J. (2012). A theoretical and experimental comparison of the equivalent source method and a bayesian approach to noise source identification. *BeBeC-2012-21. February*.

Picone, J. W. (1993). Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, *81*(9), 1215-1247.

Piczak, K. J. (2015, 2015-09). Environmental sound classification with convolutional neural networks. 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP),

Piczak, K. J. (2015). ESC: Dataset for environmental sound classification. Proceedings of the 23rd ACM international conference on Multimedia,

Pons, J., Slizovskaia, O., Gong, R., Gómez, E., & Serra, X. (2017). Timbre analysis of music audio signals with convolutional neural networks. 2017 25th European Signal Processing Conference (EUSIPCO),

Prabakaran, D., & Shyamala, R. (2019). A review on performance of voice feature extraction techniques. 2019 3rd International Conference on Computing and Communications Technologies (ICCCT),

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.-Y., & Sainath, T. (2019). Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 206-219.

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. Prentice-Hall, Inc.

Randall, R. B. (2021). *Vibration-based condition monitoring: industrial, automotive and aerospace applications*. John Wiley & Sons.

Richard, G., Sundaram, S., & Narayanan, S. (2013). An overview on perceptually motivated audio indexing and classification. *Proceedings of the IEEE*, *101*(9), 1939-1954.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP),

Salamon, J., & Bello, J. P. (2015). Unsupervised feature learning for urban sound classification. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),

Salamon, J., & Bello, J. P. (2017). Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Processing Letters*, *24*(3), 279-283. https://doi.org/10.1109/LSP.2017.2657381

Salamon, J., Bello, J. P., Farnsworth, A., & Kelling, S. (2017). Fusing shallow and deep learning for bioacoustic bird species classification. 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP),

Salamon, J., Jacoby, C., & Bello, J. P. (2014, November 3, 2014). A Dataset and Taxonomy for Urban Sound Research.*MM '14*

Salin, M., & Kosteev, D. (2020). Nearfield acoustic holography-based methods for far field prediction. *Applied Acoustics*, *159*, 107099.

Sarkar, S. (2024). Time-domain music source separation for choirs and ensembles.

Sawalhi, N., & Randall, R. B. (2004). The application of spectral kurtosis to bearing diagnostics. Proceedings of ACOUSTICS,

Schenone, C., Pittaluga, I., Borelli, D., Kamali, W., & Moghrabi, Y. E. (2016). The impact of environmental noise generated from ports: outcome of MESP project. *Noise Mapping*, *3*(1). https://doi.org/doi:10.1515/noise-2016-0002

Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. International conference on artificial neural networks,

Schlüter, J., & Grill, T. (2015). Exploring data augmentation for improved singing voice detection with neural networks. ISMIR,

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on computer vision and pattern recognition,

Seo, H., Park, J., & Park, Y. (2019). Acoustic scene classification using various pre-processed features and convolutional neural networks. Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE), New York, NY, USA,

Serizel, R., Bisot, V., Essid, S., & Richard, G. (2018). Acoustic features for environmental sound analysis. *Computational analysis of sound scenes and events*, 71-101.

Shahin, I., Nassif, A. B., & Hindawi, N. (2021). Speaker identification in stressful talking environments based on convolutional neural network. *International Journal of Speech Technology*, *24*(4), 1055-1066.

Sharan, R. V., & Moir, T. J. (2016). An overview of applications and advancements in automatic sound recognition. *Neurocomputing*, *200*, 22-34. https://doi.org/10.1016/j.neucom.2016.03.020

Sharan, R. V., & Moir, T. J. (2016). An Overview of Applications and Advancements in Automatic Sound Recognition. *Neurocomputing*. http://dx.doi.org/10.1016/j.neucom.2016.03.020

Sharan, R. V., Xiong, H., & Berkovsky, S. (2021). Benchmarking audio signal representation techniques for classification with convolutional neural networks. *Sensors*, *21*(10), 3434.

Sharma, G., Umapathy, K., & Krishnan, S. (2020). Trends in audio signal feature extraction methods. *Applied Acoustics*, *158*, 107020.

Sharma, J., Granmo, O.-C., & Goodwin, M. (2020). Environment Sound Classification Using Multiple Feature Channels and Attention Based Deep Convolutional Neural Network. Interspeech,

Sidhu, M. S., Latib, N. A. A., & Sidhu, K. K. (2024). MFCC in audio signal processing for voice disorder: a review. *Multimedia Tools and Applications*, 1-21.

Sijtsma, P. (2007). CLEAN based on spatial source coherence. *International journal of aeroacoustics*, *6*(4), 357-374.

Sijtsma, P., Merino-Martinez, R., Malgoezar, A. M., & Snellen, M. (2017). High-resolution CLEAN-SC: Theory and experimental validation. *International journal of aeroacoustics*, *16*(4-5), 274-298.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

Stevens, E., Antiga, L., & Viehmann, T. (2020). *Deep learning with PyTorch*. Manning Publications.

Stowell, D., & Plumbley, M. D. (2010). Birdsong and C4DM: A survey of UK birdsong and machine recognition for music researchers. *Centre for Digital Music, Queen Mary University of London, Tech. Rep. C4DM-TR-09-12*.

Su, Y., Zhang, K., Wang, J., & Madani, K. (2019). Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion. *Sensors*, *19*(7), 1733. https://doi.org/10.3390/s19071733

Topal, M. O., Bas, A., & van Heerden, I. (2021). Exploring transformers in natural language generation: Gpt, bert, and xlnet. *arXiv preprint arXiv:2102.08036*.

Tsalera, E., Papadakis, A., & Samarakou, M. (2021). Comparison of pre-trained CNNs for audio classification using transfer learning. *Journal of Sensor and Actuator Networks*, *10*(4), 72.

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, *10*(5), 293-302.

Valenzise, G., Gerosa, L., Tagliasacchi, M., Antonacci, F., & Sarti, A. (2007). Scream and gunshot detection and localization for audio-surveillance systems. 2007 IEEE Conference on Advanced Video and Signal Based Surveillance,

Van Breemen, T., Popp, C., Witte, R., Wolfkenfelt, F., & Wooldridge, C. (2008). Good practice guide on port area noise mapping and management. Proceedings of the 8th European Conference on Noise Control,

Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. *Advances in neural information processing systems*, *26*.

Van Trees, H. L. (2002). *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Veronesi, W., & Maynard, J. D. (1987). Nearfield acoustic holography (NAH) II. Holographic reconstruction algorithms and computer implementation. *The Journal of the Acoustical Society of America*, *81*(5), 1307-1322.

Vidaña-Vila, E., Navarro, J., Alsina-Pagès, R. M., & Ramírez, Á. (2020). A two-stage approach to automatically detect and classify woodpecker (Fam. Picidae) sounds. *Applied Acoustics*, *166*, 107312. https://doi.org/10.1016/j.apacoust.2020.107312

Vidaña-Vila, E., Navarro, J., Borda-Fortuny, C., Stowell, D., & Alsina-Pagès, R. M. (2020). Low-Cost Distributed Acoustic Sensor Network for Real-Time Urban Sound Monitoring. *Electronics*, *9*(12), 2119. https://doi.org/10.3390/electronics9122119

Virtanen, T., Plumbley, M. D., & Ellis, D. (2018). *Computational analysis of sound scenes and events*. Springer.

von Platen, P., Tao, F., & Tur, G. (2020). Multi-task Siamese neural network for improving replay attack detection. *arXiv preprint arXiv:2002.07629*.

Wang, D., & Brown, G. J. (2006). *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press.

Wang, M. E. (1978). *The application of coherence function techniques for noise source identification* Purdue University].

Wang, M. E., & Crocker, M. J. (1983). On the application of coherence techniques for source identification in a multiple noise source environment. *The Journal of the Acoustical Society of America*, *74*(3), 861-872. https://doi.org/10.1121/1.389873

Wang, R. (2022). Recognition and Detection of Vehicle Noise and Vibration Signals Relying on Variable Step Size LMS Algorithm. *Mobile Information Systems*, *2022*(1), 9396102.

Wang, Y., Wei-Kocsis, J., Springer, J. A., & Matson, E. T. (2022, 2022//). Deep Learning in Audio Classification. Information and Software Technologies, Cham.

Wang, Z.-Q., & Wang, D. (2018). Combining spectral and spatial features for deep learning based blind speaker separation. *IEEE/ACM transactions on audio, speech, and language processing*, *27*(2), 457-468.

Wei, P., He, F., Li, L., & Li, J. (2020). Research on sound classification based on SVM. *Neural Computing and Applications*, *32*, 1593-1607.

Wieczorkowska, A., Kubera, E., Słowik, T., & Skrzypiec, K. (2018). Spectral features for audio based vehicle and engine classification. *Journal of Intelligent Information Systems*, *50*, 265-290.

Wu, H., Siegel, M., & Khosla, P. (1998, 1998-05). Vehicle sound signature recognition by frequency vector principal component analysis. IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No.98CH36222),

Xiong, J. (2019). Research on Theories and Methods of Vehicle Sound Source Recognition. IOP Conference Series: Materials Science and Engineering,

Xu, W., McAuley, J., Dubnov, S., & Dong, H.-W. (2023). Equipping Pretrained Unconditional Music Transformers with Instrument and Genre Controls. 2023 IEEE International Conference on Big Data (BigData),

Yang, Z. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*.

Zaidi, B. F., Selouani, S. A., Boudraa, M., & Sidi Yakoub, M. (2021). Deep neural network architectures for dysarthric speech analysis and recognition. *Neural Computing and Applications*, *33*(15), 9089-9108.

Zalkow, F., & Müller, M. (2021). CTC-based learning of chroma features for score–audio music retrieval. *IEEE/ACM transactions on audio, speech, and language processing*, *29*, 2957-2971.

Zhang, C., & Koishida, K. (2017). End-to-end text-independent speaker verification with triplet loss on short utterances. Interspeech,

Zhang, Y., Pan, H., Chen, Y.-C., Qiu, L., Lu, Y., Xue, G., Yu, J., Lyu, F., & Wang, H. (2023). Addressing practical challenges in acoustic sensing to enable fast motion tracking. Proceedings of the 22nd International Conference on Information Processing in Sensor Networks,

Zhang, Z., Xu, S., Cao, S., & Zhang, S. (2018). Deep convolutional neural network with mixup for environmental sound classification. Chinese conference on pattern recognition and computer vision (prcv),

Zheng, F., Zhang, G., & Song, Z. (2001). Comparison of different implementations of MFCC. *Journal of Computer science and Technology*, *16*, 582-589.

Zhong, M., Torterotot, M., Branch, T. A., Stafford, K. M., Royer, J.-Y., Dodhia, R., & Lavista Ferres, J. (2021). Detecting, classifying, and counting blue whale calls with Siamese neural networks. *The Journal of the Acoustical Society of America*, *149*(5), 3086-3094.

Zhuo, D.-B., & Cao, H. (2021). Fast sound source localization based on SRP-PHAT using density peaks clustering. *Applied Sciences*, *11*(1), 445.

Zwicker, E., & Terhardt, E. (1980). Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *The Journal of the Acoustical Society of America*, *68*(5), 1523-1525.

ANNEXE A : CODE SOURCE DES FONCTIONS PRÉSENTÉES DANS LA MÉTHODOLOGIE

A.1 – Prétraitement des enregistrements

```
def clean_audio_file(input_filename, base_output_filename, threshold,
min_duration, min_detection_duration, target_duration, created_files,
weighting='A', previous_short_segments=None):
    # Chargement du signal audio
    signal, sample_rate = librosa.load(input_filename, sr=None)
    # Calcul de l'enveloppe RMS
hop_length = 512
    envelope = librosa.feature.rms(y=signal, frame_length=hop_length,
hop_length=hop_length, center=True)
    # Conversion de l'enveloppe RMS en dB
    envelope_db = librosa.amplitude_to_db(envelope, ref=np.max)
    # Initialisation de la pondération A
    a_weighting = np.zeros(envelope_db.shape)
    # Application de la pondération A
    if weighting == 'A' :
        frequencies = np.fft.fftfreq(hop_length, 1 / sample_rate)
        frequencies += 1e-10  # Évite la division par zéro
        a_weighting = librosa.A_weighting(frequencies)
        envelope_weighted_db = envelope_db + a_weighting[:, np.newaxis] +
80
    else:
        envelope_weighted_db = envelope_db
    # Calcul du niveau moyen pondéré
    mean_level = np.mean(envelope_weighted_db - 80)
    # Ajustement du seuil
    threshold_adjusted = mean_level + threshold
    # Détection des segments significatifs
    non_significant_mask = envelope_weighted_db < threshold_adjusted
    # Extraction et sauvegarde des segments pertinents
    # (Code d'extraction et sauvegarde des segments)
return previous_short_segments, created_files
```

A.2 – Génération des jumeaux positifs et négatifs

```
# Fonction pour ajouter un retard aléatoire à un signal
def delay_signal(signal, max_delay, sr) :
    delay = random.randint(0, max_delay)
    signal_short = signal[: len(signal) - delay]
    signal_delayed = np.pad(signal_short, (delay, 0))
    return signal_delayed
# Génération des jumeaux positifs
def create_positive_twin(anchor, random_sounds, rose_noise, sr) :
```

```
    # Retard aléatoire pour l'ancre
    anchor_delayed = delay_signal(anchor, max_delay=int(duree * sr / 2),
sr=sr)
    # Retards aléatoires pour les sons aléatoires
    sound1_delayed = delay_signal(random_sounds[0], max_delay=int(duree *
sr / 2), sr=sr)
    sound2_delayed = delay_signal(random_sounds[1], max_delay=int(duree *
sr / 2), sr=sr)
    # Mélange des sons non dominants et du bruit rose
    background = sound1_delayed + sound2_delayed + rose_noise
    background = normalize(background)
    # Ajustement du S/B
    att = getAtt(anchor_delayed, background, sn=SN)
    background_adjusted = background * att
    # Création du jumeau positif
    positive_twin = anchor_delayed + background_adjusted
    positive_twin = normalize(positive_twin)
    return positive_twin
# Génération des jumeaux négatifs
def create_negative_twin(anchor, random_sounds, rose_noise, sr) :
    # Retard aléatoire pour le son principal (son aléatoire)
    main_sound_delayed = delay_signal(random_sounds[0], max_delay=int(duree
* sr / 2), sr=sr)
    # Retard pour l'ancre (présente mais non dominante)
    anchor_delayed = delay_signal(anchor, max_delay=int(duree * sr / 2),
sr=sr)
    # Retard pour un autre son aléatoire
    sound2_delayed = delay_signal(random_sounds[1], max_delay=int(duree *
sr / 2), sr=sr)
    # Mélange de l'ancre (non dominante), un autre son et du bruit rose
    background = anchor_delayed + sound2_delayed + rose_noise
    background = normalize(background)
    # Ajustement du S/B
    att = getAtt(main_sound_delayed, background, sn=SN)
    background_adjusted = background * att
    # Création du jumeau négatif
    negative_twin = main_sound_delayed + background_adjusted
    negative_twin = normalize(negative_twin)
    return negative_twin
```

A.3 – Extraction des MFCC

```
class TripletDataset(Dataset):
    def __init__(self, data, base_path, n_mfcc=20, mfcc_start_coef=1,
num_mfcc_used=None, log_mels=True, melkwargs=None):
        self.mfcc_transform = torchaudio.transforms.MFCC(n_mfcc=n_mfcc,
log_mels=log_mels, melkwargs=melkwargs)
        # ...
    def __getitem__(self, idx):
        # Chargement des fichiers audio
        anchor_waveform, _ = torchaudio.load(anchor_path)
        positive_waveform, _ = torchaudio.load(positive_path)
```

```
        negative_waveform, _ = torchaudio.load(negative_path)
        # Extraction des MFCC
anchor_mfcc = self.mfcc_transform(anchor_waveform).squeeze().transpose(0,
1)
        positive_mfcc =
self.mfcc_transform(positive_waveform).squeeze().transpose(0, 1)
        negative_mfcc =
self.mfcc_transform(negative_waveform).squeeze().transpose(0, 1)
        # ...
        return anchor_mfcc, positive_mfcc, negative_mfcc
```

A.4 – Architecture du sous-réseau convolutionnel

```
class Siamese(nn.Module):
    def __init__(self, input_shape):
        super(Siamese, self).__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(1, 32, kernel_size=(3, 3)),
            nn.ReLU(),
            nn.MaxPool2d((2, 2)),
            nn.Conv2d(32, 64, kernel_size=(3, 3)),
            nn.ReLU(),
            nn.MaxPool2d((2, 2)),
            nn.Conv2d(64, 128, kernel_size=(3, 3)),
            nn.ReLU()
        )
        # Calcul de la taille de sortie
        with torch.no_grad() :
            dummy_x = torch.zeros(1, *input_shape)
            dummy_x = self.conv(dummy_x)
            out_shape = dummy_x.view(-1).shape[0]
        # Couches entièrement connectées
        self.fc = nn.Sequential(
            nn.Flatten(),
            nn.Linear(out_shape, 128),
            nn.ReLU(),
            nn.Linear(128, 1)
        )
    def forward(self, x):
        x = x.unsqueeze(1)
        x = self.conv(x)
        x = self.fc(x)
        return x
```

A.5 – Fonction de perte par triplet

```
def triplet_loss(anchor, positive, negative, margin):
    pos_dist = (anchor - positive).pow(2).sum(1)
    neg_dist = (anchor - negative).pow(2).sum(1)
    loss = torch.relu(pos_dist - neg_dist + margin)
```

```
    return loss.mean()
```

A.6 – Chargement des données

```
train_dataset = TripletDataset(train_df, base_path, n_mfcc=n_mfcc,
mfcc_start_coef=mfcc_start_coef, num_mfcc_used=num_mfcc_used,
melkwargs=melkwargs)
train_dataloader = DataLoader(train_dataset, batch_size=batch_size,
shuffle=True, num_workers=num_workers)
```

A.7 – Entraînement du modèle

```
def train_siamese_network(model, train_loader, val_loader, n_epochs,
margin, device, early_stopping, min_epochs, summary_file,
early_stopping_model_filename):
    lr = 0.01  # Taux d'apprentissage initial
    optimizer = optim.Adam(model.parameters(), lr=lr)
    scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=5,
gamma=0.1)
    for epoch in range(epochs):
        model.train()
        for batch in train_dataloader:
            anchor, positive, negative = [x.to(device) for x in batch]
            optimizer.zero_grad()
            # Passage dans le modèle
            anchor_embedding = model(anchor)
            positive_embedding = model(positive)
            negative_embedding = model(negative)
            # Calcul de la perte
            loss = triplet_loss(anchor_embedding, positive_embedding,
negative_embedding, margin)
            loss.backward()
            optimizer.step()
```

A.8 – Arrêt automatique

```
class EarlyStopping:
    def __init__(self, patience=5, verbose=True, delta=0,
path='checkpoint.pt', trace_func=print):
        # Initialisation des paramètres pour l'arrêt automatique
        self.patience = patience
        self.verbose = verbose
        self.counter = 0
        self.best_score = None
        self.early_stop = False
        self.val_loss_min = float('inf')
        self.delta = delta
        self.path = path
        self.trace_func = trace_func

    def __call__(self, val_loss, model):
        score = -val_loss
```

```
        if self.best_score is None:
            self.best_score = score
            self.save_checkpoint(val_loss, model)
        elif score < self.best_score + self.delta:
            self.counter += 1
            if self.verbose:
                self.trace_func(f'EarlyStopping counter: {self.counter} out
of {self.patience}')
            if self.counter >= self.patience:
                self.early_stop = True
        else:
            self.best_score = score
            self.save_checkpoint(val_loss, model)
            self.counter = 0

    def save_checkpoint(self, val_loss, model):
        """Sauvegarde du modèle lorsque la perte de validation diminue."""
        if self.verbose:
            self.trace_func(f'Validation loss decreased
({self.val_loss_min:.6f} --> {val_loss:.6f}).  Saving model ...')
        torch.save(model.state_dict(), self.path)
        self.val_loss_min = val_loss
```

A.9 – Évaluation du modèle

```
def evaluate_model(model, dataloader, device, margin):
    model.eval()
    with torch.no_grad():
        for batch in dataloader:
            anchor, positive, negative = [x.to(device) for x in batch]
            # Calcul des représentations vectorielles des données
            anchor_embedding = model(anchor)
            positive_embedding = model(positive)
            negative_embedding = model(negative)
            # Calcul de la perte
            loss = triplet_loss(anchor_embedding, positive_embedding,
negative_embedding, margin)
            # Calcul des distances pour les métriques
            # ...
```

A.10 – Détection automatique du périphérique

```
def get_device():
    if torch.cuda.is_available():
        return torch.device("cuda")
    elif torch.backends.mps.is_available():
        return torch.device("mps")
    else:
        return torch.device("cpu")
```

ANNEXE B : TABLE DES MATIÈRES DÉTAILLÉE

Ce document utilise une hiérachisation des sections jusqu'au 6e niveau. Nous mettons ici à la disposition du lecteur, en complément à la table des matières du début du document, une table des matières détaillée.