REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ: ABBES LAGHROUR –KHENCHELA-

FACULTÉ: GÉNIE ELECTRIQUE

OPTION : SYSTEMES DES TELECOMMUNICATIONS

PROJET DE FIN DE CYCLE

MEMOIRE

Présenté en vue de l'obtention du diplôme de

Master en Télécommunication

THEME

Application de modèles de réseaux de neurones convolutifs pré entraînés à la classification des différents types de bruits sonores environnementaux

Réalisé par : Fahloul Loutfi

Soutenu le : 14/06/2026 , devant le jury constitué de :

Dr. GHERIBI Khadidja	*Encadrante*
Dr.MAAMRI Fouzia	*Présidente*
Dr FRIHA Souad	*Examinatrice*

Année Universitaire : 2025 – 2026

Remerciements

Avant de commencer, il est important de souligner que notre gratitude envers Allah est primordiale pour la réalisation de ce travail.

Nous souhaitons exprimer notre profonde reconnaissance envers nos parents pour leur aide inestimable, leur patience et leur amour inconditionnel.

Nous tenons à exprimer notre sincère gratitude à notre encadrante, Madame la Professeure Ghribi Khadidja, pour sa patience, sa disponibilité et surtout pour la confiance qu'elle nous a accordée. Ses remarques pertinentes, ses conseils avisés et son accompagnement bienveillant ont été d'une valeur inestimable, et nous lui en sommes profondément reconnaissants.

Nous adressons également nos vifs remerciements à tous les membres du jury pour l'intérêt qu'ils ont porté à ce travail ainsi que pour la pertinence de leurs remarques.

Nos remerciements s'adressent aussi à l'ensemble des enseignants de la faculté des Sciences et Technologies, spécialité Systèmes des Télécommunications, de l'Université Abbes Laghrour – Khenchela, qui ont grandement contribué à notre formation académique et professionnelle.

Nous tenons également à remercier notre frère Isslam Fahloul pour son soutien et ses encouragements tout au long de ce parcours.

Enfin, nous adressons nos remerciements à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet, ainsi qu'à l'ensemble de la promotion Systèmes des Télécommunications 2025/2026.

Merci pour tout, cette réussite est aussi la vôtre.

Dédicace :

Je souhaite exprimer ma sincère reconnaissance envers mes parents pour leur soutien indéfectible, leur patience inébranlable, leurs sacrifices incommensurables et leur amour inconditionnel. Leur contribution mérite les plus grands éloges. C'est grâce à eux que je suis devenu ce que je suis aujourd'hui. J'aspire à être à la hauteur de leurs attentes.

Que Dieu veille sur eux et les comble de bénédictions.

Je tiens également à dédier ce travail à mon cher frère Isslam Fahloul, dont les marques d'affection et les encouragements ont toujours été inestimables pour moi.

Je salue ceux qui ont fait des sacrifices, qui ont fait preuve de patience et qui ont tout donné pour poursuivre leurs études. Leur soif d'apprendre et de se perfectionner pour illuminer le monde par leur savoir est une source d'inspiration.

J'adresse mes remerciements à tous mes amis et à toutes les personnes qui m'ont apporté leur aide. À tous ceux que j'aime, je dédie ce travail avec une profonde gratitude.

ملخص

تم اقتراح عدة مقاربات لتصنيف الأصوات البيئية بهدف معالجة المشكلات الرئيسية المرتبطة بالتعرف الآلي على الإشارات الصوتية. وقد تم إيلاء اهتمام خاص للتعلم العميق والتعلم بالنقل، ولا سيما الشبكات العصبية الالتفافية(CNN) ، التي أثبتت فعاليتها في هذا المجال.

في هذا العمل، نقدم دراسة مقارنة بين معمارين مسبقَي التدريب، ConvNeXtBase وEfficientNetV2S، مطبَّقين على تصنيف الأصوات البيئية المستخرجة من قاعدة البيانات .ESC-50وقد اخترنا 26 فئة صوتية، واستعملنا تمثيلين بصريين للإشارة الصوتية، وهما السبكتروغرامات والسبكتروغرامات الميلية، لتقييم أثرهما على أداء التصنيف. وتم تدريب النموذجين وتحسينهما داخل بيئة Google Colabمع الاستفادة من موارد الـ GPUالمتاحة. وبناءً على ذلك، نلاحظ أن التهيئة التي تجمع بين EfficientNetV2Sو Mel-spectrogramقد حققت أفضل دقة بنسبة .97.88%

الكلمات المفتاحية :الأصوات البيئية، ESC-50، الشبكات العصبية الالتفافية، التعلم بالنقل، ConvNeXtBase، EfficientNetV2S، السبكتروغرام، السبكتروغرام الميلّي.

Abstract

Several approaches to environmental sound classification have been proposed to address the main challenges of automatic audio signal recognition. Particular attention has been given to deep learning and transfer learning, especially convolutional neural networks (CNNs), which have proven to be effective solutions in this field.

In this work, we present a comparative study of two pre-trained architectures, ConvNeXtBase and EfficientNetV2S, applied to the classification of environmental sounds from the ESC-50 dataset. We selected 26 sound classes and used two visual representations of the audio signal, namely spectrograms and Mel-spectrograms, in order to evaluate their impact on classification performance. Both models were trained and optimized in the Google Colab environment, taking advantage of its available GPU resources. Based on the results, we observe that the configuration combining EfficientNetV2S with the Mel-spectrogram achieved the best accuracy, with a score of 97.88%.

Keywords: Environmental sounds, ESC-50, CNN, Transfer Learning, ConvNeXtBase, EfficientNetV2S, Spectrogram, Mel-spectrogram.

Résumé

Plusieurs approches de classification des sons environnementaux ont été introduites afin de résoudre les principaux problèmes liés à la reconnaissance automatique des signaux audio. Une attention particulière a été accordée à l'apprentissage profond et au transfert d'apprentissage, en particulier aux réseaux de neurones convolutifs (CNN), qui se sont imposés comme des solutions performantes dans ce domaine.

Dans ce travail, nous présentons une étude comparative de deux architectures pré-entraînées, ConvNeXtBase et EfficientNetV2S, appliquées à la classification des sons environnementaux issus de la base de données ESC-50. Nous avons sélectionné 26 classes de sons et utilisé deux représentations visuelles du signal audio, à savoir les spectrogrammes et les Mel-spectrogrammes, afin d'évaluer leur impact sur les performances de classification. Les deux modèles ont été entraînés et optimisés dans l'environnement Google Colab, en exploitant ses ressources GPU. Sur ce, nous remarquons que la configuration associant EfficientNetV2S au Mel-spectrogramme a fourni la meilleure précision avec un taux de 97.88%.

Mots clés : Sons environnementaux, ESC-50, CNN, Transfer Learning, ConvNeXtBase, EfficientNetV2S, Spectrogramme, Mel-spectrogramme.

Sommaire :

Liste des figures

Liste des tableaux

LISTE DES ABRÉVIATIONS

- **ANN** : Artificial Neural Networks (Réseaux de neurones artificiels).
- **CNN** : Convolutional Neural Network (Réseau de neurones convolutifs).
- **CONV** : Convolution / Couche de convolution.
- **dB** | Décibel (Unité d'amplitude sonore).
- **DCNN** : Deep Convolutional Neural Network (Réseau de neurones convolutifs profonds).
- **DFT** : Discrete Fourier Transform (Transformée de Fourier discrète).
- **DL** : Deep Learning (Apprentissage profond).
- **DNN** : Deep Neural Network (Réseau de neurones profonds).
- **ESC** : Environmental Sound Classification (Classification des sons environnementaux).
- **FC** : Fully Connected / Couche entièrement connectée.
- **FN** : Faux Négatif / False Negative.
- **FP** : Faux Positif / False Positive.
- **GPU** : Graphics Processing Unit (Unité de traitement graphique).
- **Hz** | Hertz (Unité de fréquence).
- **IA** : Intelligence Artificielle.
- **LAN** : Local Area Network (Réseau local).
- **MAN** : Metropolitan Area Network (Réseau métropolitain).
- **MBConv** : Mobile Inverted Bottleneck Convolution (Bloc d'architecture spécifique d'EfficientNet).
- **MFCC** : Mel-Frequency Cepstral Coefficients.
- **ML** : Machine Learning (Apprentissage automatique).
- **PAN** : Personal Area Network (Réseau personnel).
- **RAC** : Resource Adaptive Convolution (Module convolutif adapté aux ressources).
- **RACNN** : Resource Adaptive Convolutional Neural Network.
- **RAM** : Random Access Memory (Mémoire vive).
- **ReLU** : Rectified Linear Unit (Unité linéaire rectifiée).
- **RGB** : Red, Green, Blue (Rouge, Vert, Bleu - Format d'image).
- **STFT** : Short-Time Fourier Transform (Transformée de Fourier à court terme).
- **Tanh** : Tangente Hyperbolique.
- **TN** : Vrai Négatif / True Negative.
- **TP** : Vrai Positif / True Positive.
- **TPU** : Tensor Processing Unit (Unité de traitement tensoriel).
- **VRAM** : Video Random Access Memory.
- **WAN** : Wide Area Network (Réseau étendu).
- **WAV** : Waveform Audio File Format (Format de fichier audio brut).

INTRODUCTION GENERALE

Introduction Générale :

L'environnement sonore qui nous entoure constitue une source d'informations riche et complexe, dont l'analyse automatique représente un enjeu majeur pour de nombreux domaines applicatifs, tels que la surveillance urbaine, l'assistance aux personnes malentendantes, la robotique mobile ou encore l'écologie acoustique. Contrairement à la reconnaissance de la parole, qui bénéficie de décennies de recherche et de corpus massifs, la classification des sons environnementaux demeure un problème ouvert, en raison de la grande variabilité intra-classe des signaux acoustiques, de la superposition fréquente de sources sonores et de la rareté des jeux de données annotés de grande ampleur [76]. L'émergence de l'apprentissage profond, et plus particulièrement des réseaux de neurones convolutifs (CNN), a toutefois ouvert de nouvelles perspectives en offrant la capacité d'extraire automatiquement des représentations hiérarchiques discriminantes à partir des données brutes ou de leurs transformées temps-fréquence [77].

Dans ce contexte, la transformation des signaux audio en images spectrographiques — spectrogramme et Mel-spectrogramme — constitue une stratégie privilégiée, car elle permet d'exploiter la puissance des architectures de vision par ordinateur pré-entraînées sur des corpus d'images à grande échelle, telles qu'ImageNet [78]. Le transfert d'apprentissage, qui consiste à adapter les connaissances acquises sur une tâche source vers une tâche cible, s'avère particulièrement adapté à la classification des sons environnementaux, où la taille des jeux de données demeure limitée. Parmi les architectures récentes, ConvNeXtBase [79], qui modernise le design des réseaux convolutifs en s'inspirant des avancées des Transformers, et EfficientNetV2S [80], qui optimise le compromis entre précision et efficacité grâce à un entraînement progressif, figurent parmi les modèles les plus performants. Néanmoins, la question de l'interaction entre le type de représentation visuelle du signal et l'architecture du modèle demeure insuffisamment explorée : quel couple représentation-modèle offre les meilleures performances pour la classification des bruits environnementaux ?

Le présent mémoire s'inscrit dans cette problématique en proposant une étude comparative rigoureuse de deux architectures pré-entraînées — ConvNeXtBase et EfficientNetV2S — appliquées à deux représentations visuelles du signal audio — le spectrogramme et le Mel-spectrogramme — sur un sous-ensemble de 26 classes extraites de la base de données ESC-50 (Environmental Sound Classification) [76]. L'objectif est de

déterminer la configuration optimale pour la classification des sons environnementaux et d'analyser l'incidence du choix de la représentation sur les performances de chaque modèle.

Le mémoire est structuré en trois chapitres. Le premier chapitre pose les fondements théoriques nécessaires, en abordant l'intelligence artificielle, l'apprentissage automatique, l'apprentissage profond, ainsi que les réseaux de neurones et les réseaux de neurones convolutifs. Le deuxième chapitre est consacré à la présentation de la base de données ESC-50, du prétraitement des données, des techniques d'augmentation, des représentations spectrogramme et Mel-spectrogramme, ainsi que des modèles retenus. Enfin, le troisième chapitre expose l'implémentation expérimentale, l'évaluation des performances, la comparaison des résultats et l'analyse de la meilleure configuration obtenue.

Chapitre I

Les réseaux de neurones convolutifs

I.1 Introduction

L'intelligence artificielle connaît un essor considérable grâce aux avancées du calcul numérique et à la disponibilité croissante des données. Parmi ses applications majeures, l'analyse et la classification des signaux complexes, notamment les signaux sonores environnementaux, occupent une place importante en raison de leur variabilité et de leur complexité.

Porté par l'essor rapide des puissances de calcul et le perfectionnement continu des algorithmes d'apprentissage, le domaine des réseaux de neurones enregistre des avancées majeures. Ces progrès se manifestent de manière particulièrement probante dans des applications clés telles que le traitement d'images, la traduction automatique et la reconnaissance de la parole. [2]

I.2 Intelligence artificielle

I.2.1 Définition et cadre general

L'intelligence artificielle (IA) est probablement l'un des plus anciens domaines de l'informatique et l'un des plus vastes. Elle couvre tous les aspects liés à la reproduction des fonctions cognitives afin de résoudre des problèmes du monde réel et de concevoir des systèmes capables d'apprendre et de raisonner comme les êtres humains. [49]

Ces missions englobent notamment la compréhension du langage naturel, la reconnaissance visuelle et sonore, ainsi que la résolution de problèmes, la prise de décision et l'apprentissage continu. L'enjeu fondamental réside dans la conception de systèmes aptes à traiter, analyser et interpréter les données selon des modalités analogues à celles de l'esprit humain, en vue d'exécuter des tâches complexes et d'arrêter des choix stratégiques pertinents. Pour réaliser de tels objectifs, l'intelligence artificielle mobilise un ensemble de leviers méthodologiques, au premier rang desquels figurent l'apprentissage automatique, les réseaux de neurones artificiels et le traitement automatique des langues. [30]

I.2.2 Intelligence artificielle et analyse des signaux sonores

La classification audio possède une grande importance aussi bien théorique que pratique dans les domaines de la reconnaissance des formes et de l'intelligence artificielle. [48]

Dans cette perspective, l'intelligence artificielle offre un cadre conceptuel particulièrement adapté à l'analyse et à la classification des sons environnementaux. Plus précisément, les approches automatisées issues de l'apprentissage profond (*deep learning*) ouvrent des voies innovantes pour affiner notre compréhension des dynamiques et de l'évolution du climat. [4]

L'intelligence artificielle regroupe plusieurs sous-domaines, dont l'apprentissage automatique et l'apprentissage profond. Dans ce mémoire, elle constitue le cadre de référence des méthodes d'apprentissage automatique et des réseaux de neurones.

Le domaine de l'intelligence artificielle se divise en plusieurs sous-domaines interconnectés (présentés dans la figure I.1). L'apprentissage automatique constitue l'un de ces sous-domaines et repose sur l'acquisition de règles implicites à partir de l'expérience ou d'une base de données afin de répondre à un problème spécifique. Ce domaine s'appuie principalement sur l'analyse statistique des données d'entraînement. [50]

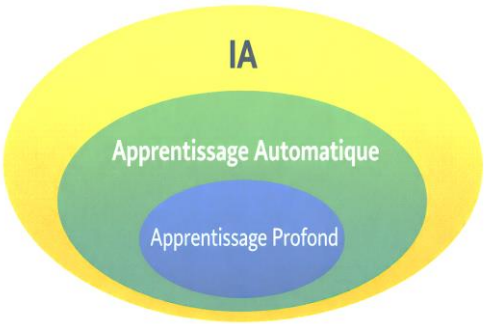

Figure I.1 : L'imbrication des sous-domaines de l'intelligence artificielle.

I.3 Apprentissage automatique et apprentissage profond

L'apprentissage automatique et l'apprentissage profond sont deux domaines liés de l'intelligence artificielle.

I.3.1 Apprentissage automatique

L'apprentissage automatique est un domaine de l'intelligence artificielle centré sur l'analyse statistique des données d'apprentissage. Historiquement, cette discipline est définie comme le développement de machines capables d'apprendre sans être explicitement programmées pour accomplir une tâche donnée. [50]

L'apprentissage automatique se décline en différents paradigmes, au premier rang desquels figurent les approches supervisées et non supervisées. Ces méthodes trouvent une application privilégiée dans des domaines clés tels que la reconnaissance vocale, l'analyse d'images et la prévision de séries chronologiques. En raison de leur polyvalence et de leur efficacité, ces algorithmes font aujourd'hui l'objet d'une adoption massive et transversale au sein du tissu industriel. [30]

I.3.1.a Principe de Fonctionnement

Fondamentalement, cette méthode repose sur l'exploitation d'algorithmes — s'apparentant à des suites de règles logiques — optimisés et ajustés à partir d'ensembles de données historiques. Ce processus d'ajustement permet aux algorithmes de formuler des prédictions ou d'opérer des catégorisations rigoureuses lorsqu'ils sont confrontés à de nouvelles observations. Au terme d'une phase d'entraînement approfondie, ces algorithmes se structurent sous la forme de "modèles d'apprentissage automatique", désormais configurés pour exécuter de manière autonome des tâches hautement spécifiques. [29]

Dans la classification des bruits environnementaux, le modèle est entraîné à partir d'enregistrements audio contenant divers sons (circulation, sirènes, conversations, sons naturels). En analysant leurs caractéristiques, il apprend à les distinguer et peut ensuite classifier automatiquement de nouveaux enregistrements.

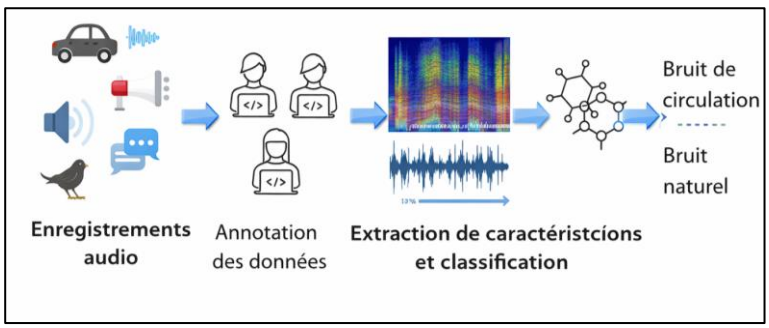

Figure I.2 : Processus global de l'apprentissage automatique appliqué à la classification sonore.

I.3.1.b Apprentissage Supervisé

Les algorithmes d'apprentissage supervisé s'appuient sur des données préalablement labellisées, c'est-à-dire associées à des étiquettes représentant les variables cibles ou les résultats attendus. Ce paradigme est classiquement mobilisé pour résoudre des problématiques de classification et de régression. [30]

- **La classification :** l'objectif réside dans l'affectation de chaque observation à une classe ou une catégorie préalablement définie.
- **La régression :** le modèle vise à estimer une variable numérique continue correspondant à la valeur de sortie attendue.

Ce paradigme d'apprentissage repose sur la modélisation d'une fonction mathématique, où un algorithme (F) est déployé pour prédire une variable de sortie (Y, l'*output*) à partir d'un ensemble de variables explicatives d'entrée (X, l'*input*), par le biais de l'analyse structurelle des données X.[30]

De façon générale, une machine peut apprendre une relation $f: x \rightarrow y$ reliant les variables d'entrée x à la variable de sortie y à partir de l'analyse de millions d'exemples d'associations $x \rightarrow y$. [30]

L'objectif ultime de l'apprentissage supervisé réside ainsi dans la construction d'un modèle prédictif — ou classifieur — capable d'assigner une catégorie à de nouvelles instances, absentes de l'ensemble d'entraînement X, tout en minimisant la marge d'erreur de généralisation. [30]

La figure illustre le principe de fonctionnement de l'apprentissage supervise:

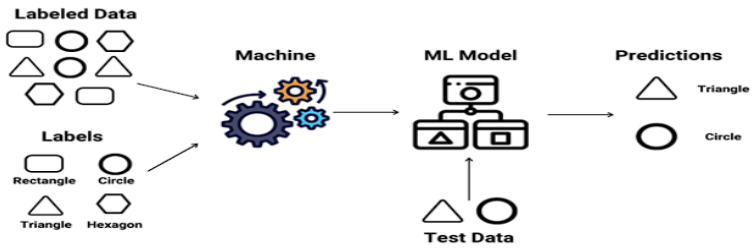

Figure I.3 : Schéma illustrant le concept de classification.

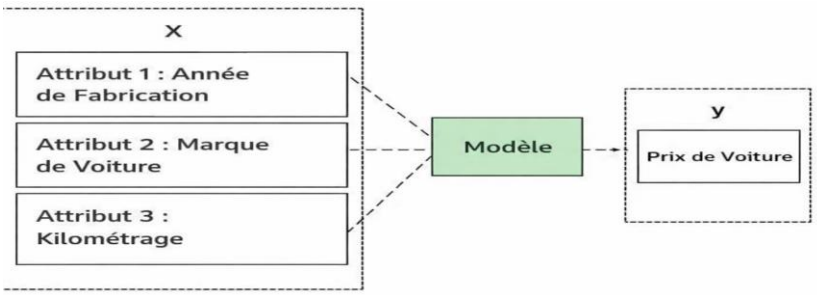

Figure : I.4 Schéma illustrant le concept de régression

Parmi les principaux algorithmes d'apprentissage supervisé figurent notamment les forêts aléatoires (*Random Forests*), les arbres de décision, la méthode des k plus proches voisins (k-NN), la régression linéaire et la régression logistique. À ces approches s'ajoutent le classifieur bayésien naïf (*Naïve Bayes*), les machines à vecteurs de support (SVM) ainsi que les techniques de boosting de gradient (*Gradient Boosting*). [19]

I.3.1.c Apprentissage Non Supervisé

À l'inverse, l'apprentissage non supervisé s'affranchit de toute intervention humaine directe ou d'annotations préalables lors de la phase d'entraînement. Les algorithmes sont ici confrontés à des données non labellisées, ce qui implique qu'ils doivent identifier de manière autonome les structures sous-jacentes, les corrélations et les régularités inhérentes aux données elles-mêmes. [30]

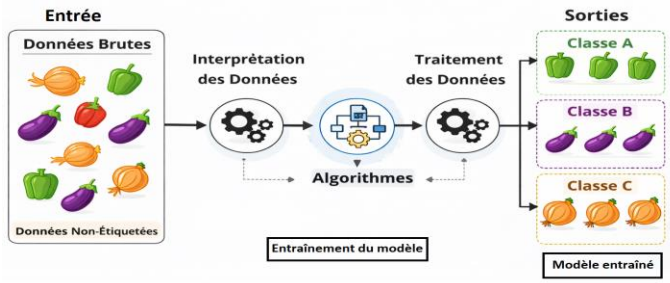

Figure I.5 : Principe de fonctionnement de l'apprentissage non supervisé (Exemple de regroupement/clustering).

L'apprentissage non supervisé se structure principalement autour de trois grandes catégories d'approches :

- **Le partitionnement (*clustering*) :** cette démarche consiste à regrouper des données non étiquetées en sous-ensembles homogènes, en se fondant sur l'évaluation de leurs similitudes et de leurs divergences.
- **Les algorithmes d'association :** ils reposent sur une modélisation par règles d'association, visant à détecter des relations structurelles ou des corrélations significatives entre les différentes variables d'un ensemble de données.
- **Les méthodes de réduction de la dimensionnalité :** ces techniques visent à projeter les données d'un espace de grande dimension vers un espace de dimension inférieure, tout en veillant à préserver l'information essentielle et les caractéristiques intrinsèques de la distribution d'origine.

Les principaux algorithmes utilisés en apprentissage non supervisé sont K-Means, le regroupement hiérarchique (*clustering hiérarchique*) ainsi que les techniques de réduction de dimensionnalité. [20]

I.3.2 Apprentissage profond

Les méthodes d'apprentissage profond (*deep learning*) constituent un sous-ensemble des techniques d'apprentissage automatique (*machine learning*), qui s'inscrit lui-même comme une branche fondamentale de l'intelligence artificielle. [4]

Les méthodes de l'apprentissage profond sont des méthodes d'apprentissage de représentations comportant plusieurs niveaux de représentation, obtenus en composant des modules simples mais non linéaires, chacun transformant la représentation à un niveau donné (à partir de l'entrée brute) en une représentation à un niveau supérieur, légèrement plus abstrait. [1]

l'apprentissage profond réalise des avancées majeures dans la résolution de problèmes qui ont résisté pendant de nombreuses années aux meilleures tentatives de la communauté de l'intelligence artificielle. Il s'est révélé très efficace pour découvrir des structures complexes dans des données de grande dimension et est, par conséquent, applicable à de nombreux domaines de la science, des affaires et du gouvernement. [1]

L'apprentissage profond trouve de nombreuses applications à travers une grande diversité de domaines [11], parmi lesquels :

— la vision par ordinateur et la reconnaissance d'images ;

— la traduction automatique et le traitement des langues ;

— le diagnostic médical assisté par ordinateur ;

— la cybersécurité, notamment pour la détection de fraudes et de codes malveillants ;

— le pilotage des véhicules autonomes.

I.3.2.a Principe de Fonctionnement

Les modèles d'apprentissage profond présentent un avantage par rapport aux modèles classiques d'apprentissage automatique grâce à un nombre plus élevé de couches d'apprentissage et à un niveau d'abstraction plus important. [51] L'apprentissage profond

fonctionne à l'aide d'une structure de nœuds qui communiquent entre eux au sein d'un réseau neuronal artificiel. [52]

L'apprentissage profond (*Deep Learning*) est une méthode de réduction de dimensionnalité fondée sur des réseaux de neurones multicouches. Grâce à la présence de plusieurs couches cachées profondes, le réseau effectue une extraction progressive des caractéristiques à partir des données de grande dimension fournies en entrée. Il permet ainsi de découvrir une structure imbriquée de plus faible dimension et de construire des représentations plus abstraites et plus efficaces des données. [81]

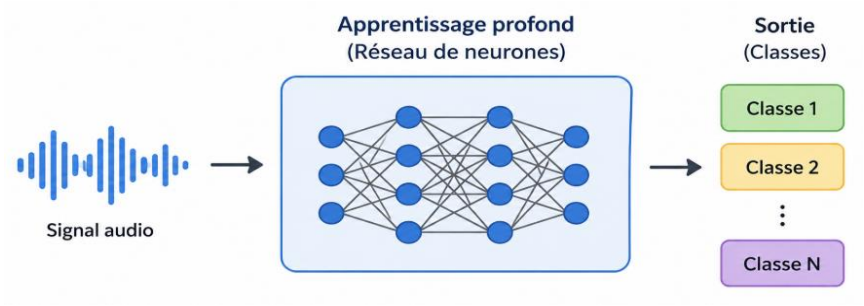

Figure I.6 : Schéma conceptuel du fonctionnement d'un réseau de neurones profond pour le traitement audio.

I.3.3 Comparaison entre l'apprentissage automatique et l'apprentissage profond

À titre de comparaison simple entre l'apprentissage profond et les techniques classiques d'apprentissage automatique, le tableau I.1 montre comment la modélisation en DL peut améliorer les performances à mesure que le volume de données augmente.

Critère de Comparaison	Apprentissage Automatique (Machine Learning)	Apprentissage Profond (Deep Learning)
Intervention Humaine	Requiert une contribution humaine régulière et continue pour l'ajustement des résultats.	Présente une mise en œuvre initiale complexe, mais fonctionne de manière quasi autonome par la suite.
Matériel (Hardware)	Repose sur des architectures logicielles simples, exécutables sur des ordinateurs de bureau standards.	Exige des ressources informatiques hautement performantes, notamment des GPU, pour gérer le parallélisme et réduire les temps de latence.
Temps (Time)	S'installe et se déploie rapidement, bien que les performances finales puissent s'avérer limitées.	Nécessite un temps de configuration initial plus long, mais fournit des résultats évolutifs avec l'accumulation des données.
Approche (Approach)	Traite prioritairement des données structurées via des méthodes statistiques classiques comme la régression.	S'appuie sur des réseaux de neurones artificiels conçus pour analyser de grands volumes de données non structurées.
Applications	Déployé couramment dans la gestion des courriels, les services bancaires et les systèmes médicaux.	Intégré dans des technologies hautement autonomes telles que la robotique chirurgicale ou la conduite automatisée.
Utilisation (Usage)	Adapté aux tâches de classification, de régression et de partitionnement de données (clustering).	Privilégié pour résoudre des problématiques complexes comme la reconnaissance vocale, le traitement d'images et du langage.
Données (Data)	Opère efficacement sur des volumes de données réduits, où la qualité de l'échantillon reste déterminante.	Dépend d'une quantité massive de données pour l'entraînement des réseaux et l'amélioration autonome du modèle.

Tableau I.1 : Synthèse comparative entre l'apprentissage automatique traditionnel et l'apprentissage profond. [51]

Ce figure I.7 illustre la différence fondamentale dans l'architecture du traitement des données entre les deux approches.Il met en évidence l'automatisation de l'extraction de caractéristiques par le Deep Learning, contrairement à l'approche manuelle requise par le Machine Learning traditionnel.

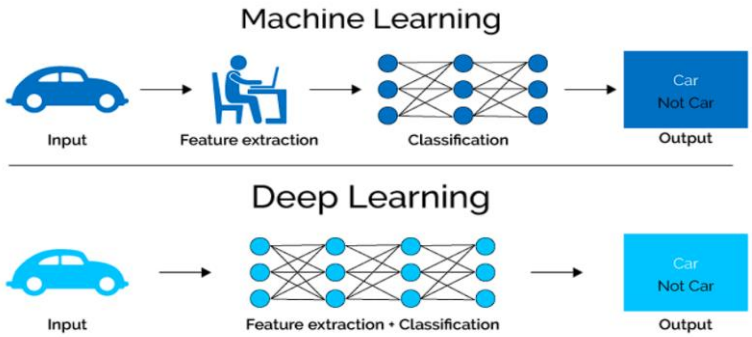

Figure I.7 : Comparaison conceptuelle du flux de traitement : Machine Learning traditionnel vs. Deep Learning

I.4 Réseaux de neurones

I.4.1 Neurones biologiques

Le neurone biologique se compose d'un corps cellulaire, de dendrites et d'un axone. Les dendrites captent les signaux électrochimiques émis par les neurones adjacents pour les acheminer vers le corps cellulaire, ou soma, qui abrite le noyau et les organites nécessaires aux fonctions vitales de la cellule. L'axone assure ensuite la propagation de l'influx nerveux vers d'autres neurones ou cellules cibles. Enfin, la zone de jonction permettant la transmission de ce signal entre deux neurones, ou entre un neurone et une cellule effectrice, est qualifiée de synapse. [5]

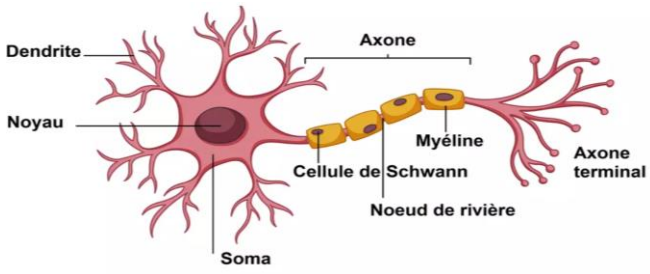

Figure I.8 : Structure biologique d'un neurone naturel et ses principaux composants.

I.4.2 Neurones artificiels

Un réseau de neurones artificiels est composé d'unités de traitement élémentaires appelées neurones, dont le fonctionnement s'inspire directement de l'analogie biologique.

Chaque neurone artificiel reçoit un ensemble de variables d'entrée, applique un traitement mathématique à ces données, puis génère une variable de sortie unique. La dynamique comportementale de cette unité est régie par une fonction d'activation, qui détermine l'amplitude de la réponse du neurone en fonction des informations cumulées en entrée. [6]

Les réseaux de neurones artificiels s'inspirent de l'architecture fonctionnelle des neurones biologiques, qu'ils formalisent en intégrant des méthodes statistiques et probabilistes. [9]

Ce tableau I.2 présente une analogie directe entre la structure biologique du système nerveux et sa modélisation mathématique. Il permet de comprendre comment chaque élément physiologique est transposé en un paramètre algorithmique pour concevoir les réseaux artificiels.

Neurone biologique	Neurone artificiel
Dendrite	Signal d'entrée
Synapses	Poids de connexions
Corps cellulaires	Fonctions d'activation
Axones	Signal de sortie

Tableau I.2 : Analogie conceptuelle entre les composants du neurone biologique et du neurone artificiel.

I.5 Réseaux de neurones artificiels ANN

I.5.1 Définition d'un Réseaux de neurones artificiels ANN

Les réseaux de neurones artificiels (ANN) sont des systèmes computationnels inspirés de la neurobiologie, conçus pour simuler les processus d'apprentissage adaptatif du cerveau humain à travers un réseau d'unités de traitement interconnectées. Cette architecture imite la capacité du système nerveux à traiter l'information en s'appuyant sur la collaboration de neurones artificiels pour résoudre des problèmes complexes. À l'instar des mécanismes d'apprentissage humains, les ANN optimisent leurs performances par l'expérience, l'exploitation d'ensembles de données massifs favorisant généralement une meilleure capacité de généralisation et une précision accrue. Leur entraînement repose sur des algorithmes d'apprentissage spécifiques, calibrés selon la tâche dédiée, telle que la reconnaissance de motifs ou la classification de données. [7]

I.5.2 Modélisation mathématique d'un neurone artificiel

La fonction d'activation détermine la valeur de sortie d'un neurone artificiel à partir d'une entrée ou d'un ensemble d'entrées combinées. En pratique, elle permet de décider si le neurone doit être activé ou non afin de converger vers le résultat attendu, en appliquant un seuillage sur la somme pondérée des entrées à laquelle est ajouté un biais. De surcroît, elle introduit une transformation non linéaire au sein du réseau, une propriété fondamentale qui permet aux architectures neuronales complexes de modéliser des relations hautement non linéaires et de résoudre des problèmes de grande dimension. [11]

La couche de mise à l'échelle vise à normaliser les variables d'entrée afin qu'elles s'inscrivent dans un intervalle et un format prédéfini, optimisant ainsi la vitesse et la précision de la phase d'entraînement. Au sein d'un réseau de neurones artificiels (ANN), les nœuds de cette couche intègrent des descripteurs statistiques calculés sur les données d'entrée, tels que la moyenne, l'écart-type, les valeurs minimales ou maximales [10, 11, 12]. Par la suite, chaque nœud (perceptron) des couches perceptives reçoit ces entrées numériques (X1, X2, X3, etc.), leur applique des coefficients de pondération respectifs (W1, W2, W3, etc.), et les combine à un terme de biais afin de générer une valeur d'entrée nette unique. [53]

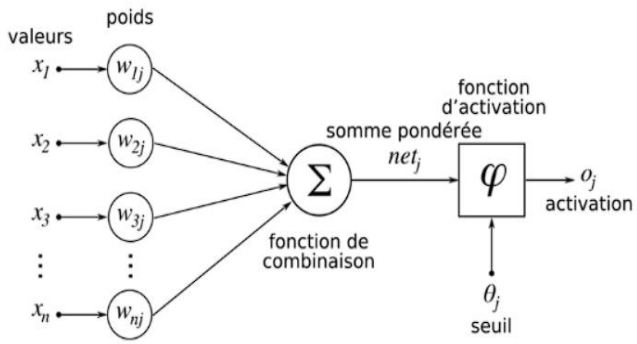

Figure I.9 : Modélisation mathématique et architecture interne d'un neurone artificiel (Perceptron).

Soit un vecteur d'entrée :

$$X = (x_1, x_2, \dots, x_n) \quad \text{.....(I.1)}$$

et un vecteur de poids associé :

$$W = (w_1, w_2, \dots, w_n) \quad \text{.....(I.2)}$$

La première étape consiste à calculer la **somme pondérée** des entrées :

$$net = \sum_{i=1}^{n} w_i\, x_i + b \quad \text{.....(I.3)}$$

où :

- x_i représente la ième entrée,
- w_i représente le poids associé à l'entrée x_i,
- b est le biais (bias),
- net est le potentiel d'activation du neurone.

Cette opération correspond à une **combinaison linéaire** des entrées.

Ensuite, le résultat obtenu est transformé par une **fonction d'activation non linéaire** $\varphi(.)$, donnant la sortie finale du neurone :

$$y = \varphi(net) = \varphi(\sum_{i=1}^{n} WiXi + b) \quad \text{.....(I.4)}$$

I.5.3 Architecture générale des réseaux de neurones artificiels

L'architecture d'un réseau de neurones artificiels (ANN) est généralement décrite comme un ensemble de nœuds interconnectés organisés en trois groupes. Le premier groupe est constitué des nœuds d'entrée qui reçoivent les informations externes destinées au modèle. Le deuxième groupe correspond aux « couches cachées » du réseau. Chaque couche cachée regroupe des nœuds qui reçoivent les données de la couche précédente, effectuent un certain traitement (combinaison et transfert), puis transmettent une sortie aux nœuds de la couche suivante. [54]

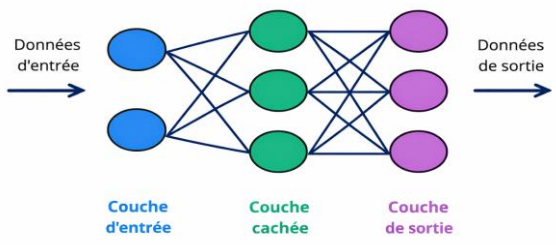

Figure I.10 : Structure générale d'un réseau de neurones artificiels (Multi-Layer Perceptron).

1. **La couche d'entrée (*Input Layer*) :** cette interface a pour fonction de recevoir les données brutes initiales afin de les injecter dans le système. En règle générale, chaque neurone de cette couche est structurellement associé à une variable descriptive du

problème ; par exemple, dans le cadre du traitement d'images, chaque unité d'entrée peut correspondre à la valeur d'un pixel donné [14].

2. **Les couches cachées (*Hidden Layers*) :** intercalées entre les couches d'entrée et de sortie, elles sont qualifiées de « cachées » car elles n'interagissent pas directement avec l'environnement extérieur. Ces couches prennent en charge la majeure partie du traitement computationnel en transformant les signaux d'entrée en représentations caractéristiques de plus en plus abstraites et complexes. Le nombre de couches intermédiaires varie selon la complexité intrinsèque de la tâche et les ressources de calcul disponibles [14].

3. **La couche de sortie (*Output Layer*) :** cette dernière composante formalise et restitue les prédictions finales issues des traitements du réseau. Sa dimension algorithmique, à savoir son nombre de neurones, dépend directement de la nature de la variable cible du problème. À titre d'illustration, une tâche de classification binaire requiert généralement deux unités de sortie, tandis qu'une classification multiclasse impose un nombre de neurones strictement égal au nombre de catégories à prédire [14].

I.5.4 Fonction d’activation

La fonction d’activation constitue une opération mathématique visant à transformer la somme pondérée des signaux d'un neurone en une valeur de sortie normalisée. Sa fonction essentielle au sein d'une architecture neuronale réside dans l'introduction d'une dynamique non linéaire. Cette rupture de linéarité confère au réseau une grande capacité de représentation, lui permettant de modéliser des relations topologiques complexes et de résoudre des problèmes de haute dimension. Il existe à cet effet plusieurs types de fonctions d’activation. [15]

a Activation du sigmoïde

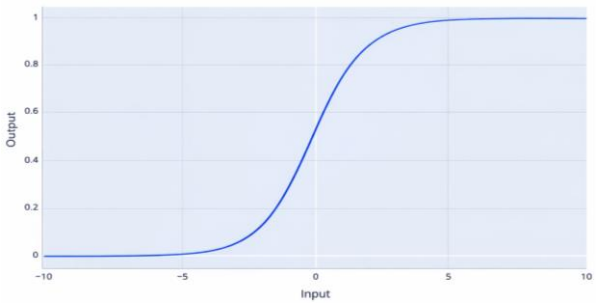

Figure I.11 : Courbe de la fonction d'activation Sigmoïde.

Étant donné que les fonctions sigmoïdes « compriment » les valeurs réelles dans un intervalle borné, elles sont parfois appelées fonctions d'écrasement (*squashing functions*). L'activation sigmoïde occupe une place historique importante dans la théorie et la pratique des réseaux de neurones. L'une des motivations de leur utilisation provient des schémas d'activation de forme sigmoïde ainsi que de l'interprétation des fonctions sigmoïdes comme des approximations mathématiquement maniables des fonctions de seuil (*step functions*). [82]

Mathématiquement, la fonction sigmoïde s'exprime par la relation suivante :

$$f(x) = \frac{1}{e^{-x}+1} \quad \text{.....(I.5)}$$

Une fonction d'activation sigmoïde est une fonction bornée, dérivable et monotone croissante, possédant un unique point d'inflexion. De manière intuitive, elle prend la forme d'une courbe lisse en « S », comme illustré à la figure I.10. Étant donné que les fonctions sigmoïdes compressent les valeurs réelles dans un intervalle borné, elles sont également appelées fonctions d'écrasement (*squashing functions*). [55]

b Activation du Tanh (tangente hyperbolique)

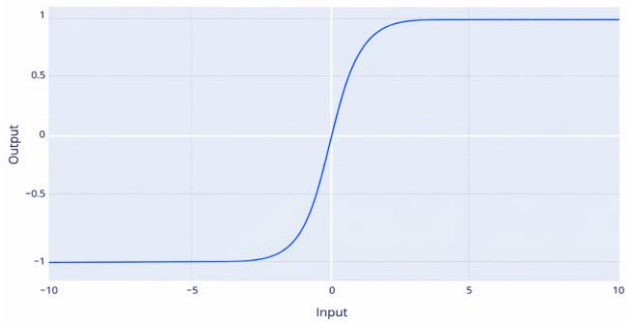

Figure I.12 : Courbe de la fonction d'activation Tanh.

Il s'agit de la fonction tangente hyperbolique (*Hyperbolic Tangent Function*). La fonction tanh est similaire à la fonction sigmoïde, mais elle est symétrique par rapport à l'origine. Cette symétrie permet d'obtenir des sorties positives et négatives, qui sont ensuite transmises comme entrées à la couche suivante du réseau. Elle peut être définie comme suit :[83]

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad \text{.....(I.6)}$$

La fonction tanh est continue et dérivable, et ses valeurs sont comprises dans l'intervalle [-1,,1]. Comparativement à la fonction sigmoïde, elle possède un gradient plus

prononcé, ce qui favorise une meilleure propagation de l'information lors de l'apprentissage. La fonction tanh est souvent préférée à la fonction sigmoïde, car ses gradients ne sont pas limités à une seule direction de variation et parce qu'elle est centrée en zéro, ce qui contribue à améliorer la convergence du réseau. [83]

c Activation de l'unité linéaire rectifiée (ReLU)

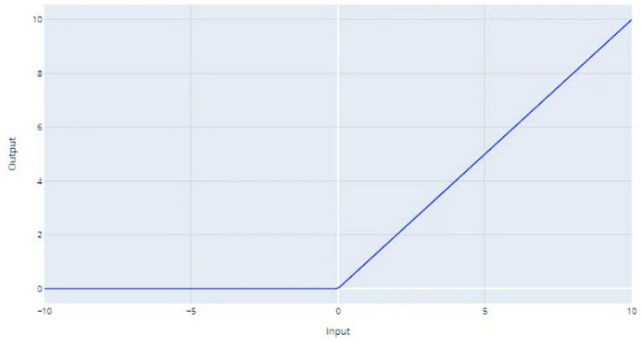

Figure I.13 : Courbe de la fonction d'activation ReLU.

La fonction ReLU (*Rectified Linear Unit*) est une fonction d'activation non linéaire largement utilisée dans les réseaux de neurones. L'un des principaux avantages de la fonction ReLU est que tous les neurones ne sont pas activés simultanément. Cela signifie qu'un neurone n'est désactivé que lorsque la sortie de la transformation linéaire est égale à zéro. Elle peut être définie mathématiquement comme suit : [83]

$$f(x) = \max(0, x) \quad \text{.....(I.7)}$$

La fonction ReLU est plus efficace que d'autres fonctions d'activation, car tous les neurones ne sont pas activés simultanément ; seul un certain nombre d'entre eux le sont à un instant donné. Dans certains cas, la valeur du gradient peut être nulle. Par conséquent, les poids et les biais ne sont pas mis à jour durant l'étape de rétropropagation (*backpropagation*) lors de l'entraînement du réseau de neurones. [83]

En ingénierie électrique, un redresseur (*rectifier*) est un dispositif qui transforme un courant alternatif en courant continu. De façon analogue, cette fonction laisse passer les valeurs d'entrée positives sans modification, tout en annulant les valeurs négatives. Autrement dit, elle convertit les entrées négatives et non négatives en sorties non négatives, comme l'illustre la figure I.13. [55]

d Activation du Softmax

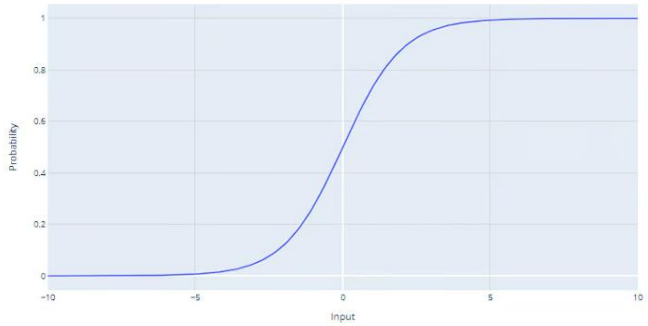

Figure I.14 : Courbe de la fonction d'activation Softmax.

La fonction **Softmax** peut être considérée comme une combinaison de plusieurs fonctions sigmoïdes. Étant donné qu'une fonction sigmoïde renvoie des valeurs comprises entre 0 et 1, celles-ci peuvent être interprétées comme des probabilités associées aux données d'une classe particulière. Ainsi, la fonction **Softmax** permet d'exprimer les sorties du réseau sous forme de probabilités pour les différentes classes. [83]

Contrairement à la fonction sigmoïde, qui est utilisée pour les problèmes de classification binaire, la fonction **Softmax** peut être employée pour les problèmes de classification multiclasse. Pour chaque donnée et pour chacune des classes considérées, elle calcule une probabilité d'appartenance. Elle peut être exprimée comme suit : [83]

$$f(xi) = \frac{e^{Xi}}{\sum_j^C e^{Xj}} \quad \text{.....(I.8)}$$

Lors de la construction d'un réseau ou d'un modèle destiné à la classification multiclasse, la couche de sortie du réseau comporte un nombre de neurones égal au nombre de classes présentes dans la cible. Chaque neurone correspond à une classe et produit une valeur représentant la probabilité associée à cette classe. [83]

I.6 Réseau de Neurones Profond DNN

I.6.1 Définition d'un Réseau de Neurones Profond DNN

Un réseau de neurones profond (DNN) est un réseau de neurones artificiels (ANN) qui comporte beaucoup plus de couches que les modèles peu profonds. Des publications récentes ont montré que les DNN sont capables d'apprendre plus facilement des correspondances plus

complexes et d'offrir de meilleures performances que les ANN à une seule couche cachée. [61]

Un réseau de neurones profond se structure autour de multiples couches de nœuds interconnectés. Chaque niveau traite l'information selon un niveau de granularité spécifique, ce qui permet ainsi au système d'extraire et d'apprendre des représentations de plus en plus abstraites et complexes des données initiales. [21]

I.6.2 Principe de Fonctionnement

L'organisation d'un réseau de neurones profond s'appuie sur une architecture complexe, modélisée selon les mécanismes de traitement de l'information du cerveau humain. Ces systèmes opèrent en propageant et en transformant les données à travers une succession de couches interconnectées. [21]

Au sein d'un réseau de neurones profond, le traitement des données s'effectue à travers une structure hiérarchique de couches interconnectées. Chaque liaison entre les unités est modulée par un poids synaptique qui en détermine l'importance relative. Afin de permettre au modèle de formaliser des motifs complexes, des fonctions d'activation sont intégrées pour introduire une dynamique non linéaire. Ainsi, l'information progresse de la couche d'entrée jusqu'à la couche de sortie, subissant des transformations successives à chaque étape du processus. [21]

Une fois la topologie d'un DNN construite, le réseau peut être alimenté par des données d'entrée d'apprentissage, et les poids associés, représentés comme des connexions entre les neurones artificiels, sont ajustés grâce au processus de rétropropagation. Cette étape correspond à la phase d'entraînement du réseau. L'entraînement est généralement réalisé une seule fois en raison du temps important qu'il nécessite, après quoi le DNN devient prêt pour la classification d'images à partir de données de test. [57]

Un réseau de neurones voit sa capacité de modélisation augmenter avec le nombre de couches, ce qui lui permet de représenter des relations plus complexes entre les données. Cependant, une profondeur excessive peut entraîner des difficultés d'entraînement, telles que le surapprentissage ou la dégradation des performances.

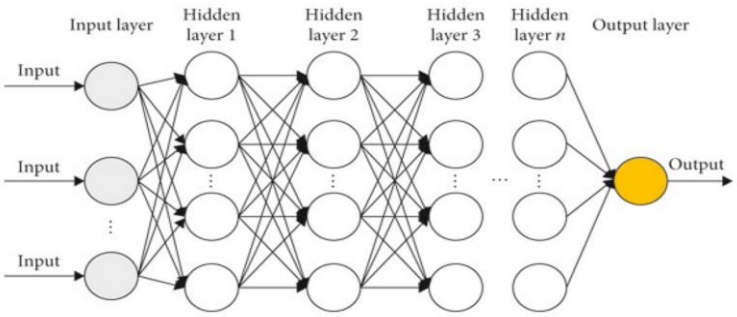

Figure I.15 : Architecture d'un réseau de neurones profond (DNN)

I.7 Réseau des Neurones Convolutifs CNN

I.7.1 Définition d'un Réseau de Neurones Convolutifs CNN

Les réseaux de neurones convolutifs, communément appelés Convolutional Neural Networks ou CNN (LeCun, 1989) [26], constituent une catégorie majeure de réseaux neuronaux dédiés au domaine de l'apprentissage profond. Ces architectures ont marqué un progrès significatif dans le champ de la reconnaissance d'images. Principalement exploités pour l'analyse de données visuelles, les CNN s'imposent aujourd'hui comme les moteurs fondamentaux des tâches de classification d'images.

Les réseaux de neurones convolutifs (CNN) se caractérisent par l'intégration de couches de convolution spécifiquement développées pour le traitement de données structurées en grille, à l'instar des images matricielles composées de pixels.

Les CNN ont obtenu des résultats remarquables dans différents domaines au cours des dix dernières années. Ces performances ont poussé les chercheurs et les développeurs à recourir à de grands modèles pour résoudre des tâches complexes, ce qui n'était pas possible avec les réseaux ANN classiques [58].

I.7.2 Origines conceptuelles des CNN et comparaison avec le système visuel humain

Le développement des réseaux neuronaux convolutifs s'inscrit dans une approche bio-inspirée reposant sur l'étude des mécanismes de traitement visuel chez l'être humain. Une mise en parallèle de ces deux systèmes permet d'identifier des correspondances structurelles.

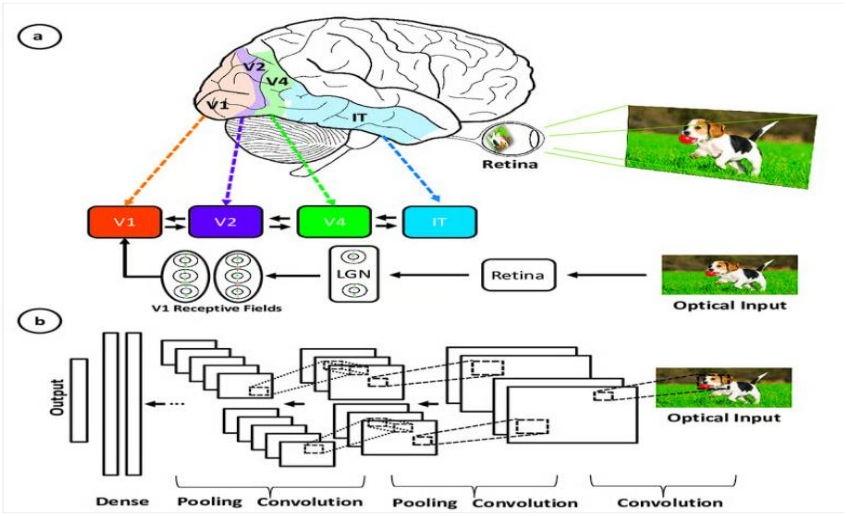

Figure I.16 : Origine bio-inspirée des réseaux de neurones convolutifs (CNN).

Le système visuel humain suit une organisation hiérarchique où les caractéristiques simples sont progressivement combinées pour former des représentations complexes. Les réseaux de neurones convolutifs reproduisent ce principe en apprenant, couche par couche, des caractéristiques de plus en plus abstraites. Cette structure leur permet d'être performants dans la reconnaissance et la classification des données.

I.7.3 Principe de fonctionnement des CNN

Une forme courante de réseaux de neurones profonds (DNN) est celle des réseaux de neurones convolutionnels (CNN), composés de plusieurs couches de convolution (CONV). Dans ce type de réseau, chaque couche produit une abstraction de plus en plus élevée des données d'entrée, appelée carte de caractéristiques (*feature map* ou *fmap*), qui conserve des informations essentielles et distinctives. Les CNN modernes parviennent à atteindre des performances supérieures grâce à l'utilisation d'une hiérarchie de couches très profonde. [59]

Un réseau de neurones convolutif peut intégrer des dizaines, voire des centaines de couches, chacune étant configurée pour détecter des caractéristiques distinctes au sein d'une image. Au cours de la phase d'apprentissage, des filtres aux résolutions variées sont appliqués sur chaque image, la sortie de chaque opération de convolution servant ensuite d'entrée à la couche suivante. Initialement, ces filtres ciblent des attributs rudimentaires, tels que la luminosité et les contours, avant de se complexifier progressivement pour capturer des caractéristiques spécifiques et discriminantes propres à l'objet analysé. [23]

Les représentations extraites sont ensuite transmises aux couches finales, généralement entièrement connectées, afin de produire la sortie du modèle sous forme de classification.

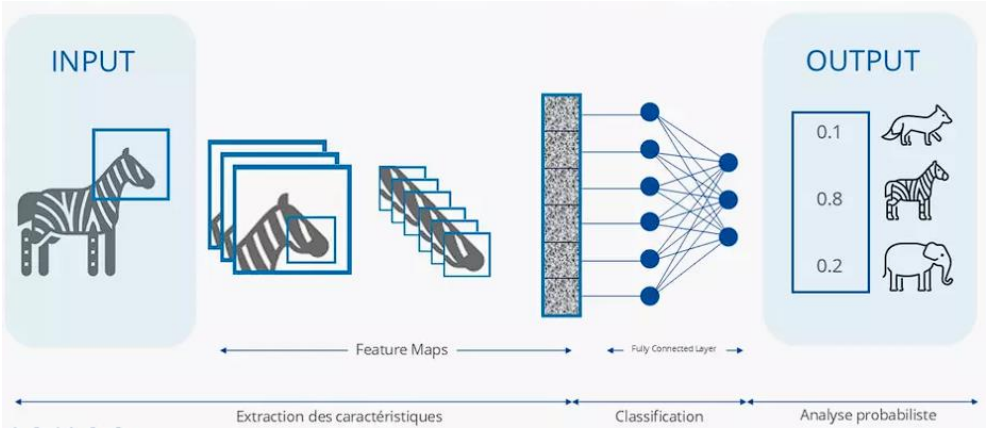

Figure I.17 : Schéma conceptuel du traitement hiérarchique et de l'extraction de caractéristiques dans CNN.

Dans cet exemple, l'image d'un animal est analysée par le réseau afin d'identifier sa catégorie. Après traitement, le modèle attribue des probabilités à chaque classe possible (renard, zèbre, éléphant), la valeur la plus élevée correspondant à la prédiction finale. Cette représentation probabiliste permet d'exprimer le degré de confiance du modèle dans sa décision.

I.7.4 Architecture des CNN

L'architecture fondamentale d'un réseau de neurones convolutif repose sur cinq étapes distinctes. Au sein de cette structure, chaque couche remplit une fonction unitaire et spécifique, leur action conjuguée permettant de transformer les images brutes en scores de classification. Ces différentes strates constituent le socle commun de la majorité des architectures de CNN, des configurations les plus superficielles aux modèles les plus profonds. [24]

Dans les réseaux de neurones convolutifs, chaque couche joue le rôle d'un filtre chargé de détecter la présence de caractéristiques ou de motifs particuliers dans les données d'origine. Les premières couches du réseau identifient des caractéristiques relativement simples à reconnaître et à interpréter. Les couches suivantes détectent progressivement des caractéristiques plus abstraites. Quant à la dernière couche du réseau convolutif, elle est capable d'effectuer une classification très spécifique en combinant l'ensemble des caractéristiques détectées auparavant dans les données d'entrée. [60]

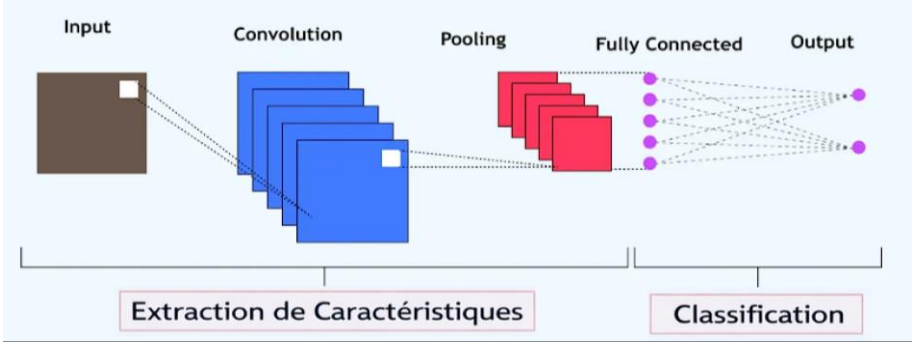

Figure I.18 : Structure globale d'une architecture CNN.

a. Couche d'entrées (Input)

La couche d'entrée constitue le point de départ du réseau, dont le rôle est de recevoir les données brutes sans traitement préalable.

b. Couche de Convolution

La couche convolutive constitue le composant fondamental de toute architecture de CNN. Elle intègre un ensemble de filtres qui subissent une opération de convolution avec l'image d'entrée à N dimensions, permettant ainsi de générer une carte de caractéristiques (*feature map*). [25]

L'objectif principal de l'étape de convolution réside dans l'extraction des caractéristiques de l'image d'entrée, cette couche constituant systématiquement le point d'entrée de tout CNN. En analysant les données d'entrée par le biais de sous-matrices de dimensions réduites, la convolution parvient à préserver les relations spatiales entre les pixels. Sur le plan formel, il s'agit d'une opération mathématique impliquant deux entrées distinctes. [26]

Dans l'architecture des CNN, il est courant qu'une couche convolutive soit suivie par une couche de pooling.

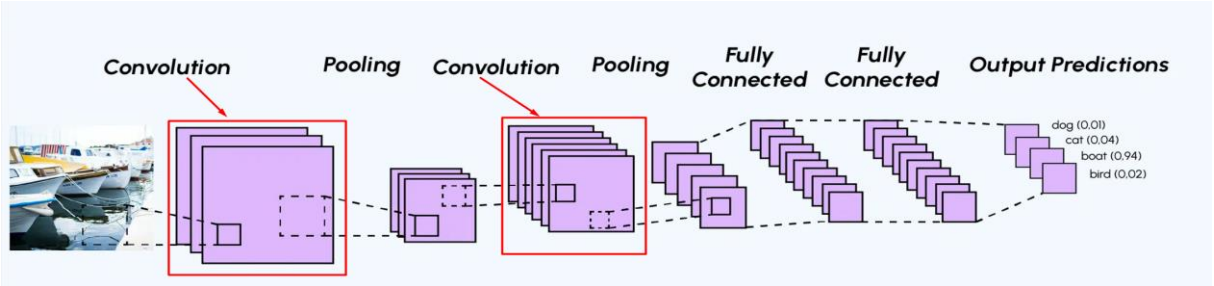

Figure I.19 : Identification des couches de convolution au sein d'un CNN.

c. Couche de Pooling

Après chaque couche convolutive, une couche de *pooling* peut être ajoutée. [60]

La couche de pooling (ou de sous-échantillonnage) réduit la dimensionnalité des cartes de caractéristiques générées par les couches précédentes du réseau de neurones convolutif. En sélectionnant les valeurs les plus significatives au sein de régions locales restreintes, elle élimine les données redondantes. Ce processus permet de préserver les attributs essentiels tout en atténuant le bruit et les informations superflues, ce qui aboutit à un modèle plus léger et computationalement plus efficient, sans altérer les caractéristiques fondamentales nécessaires à l'apprentissage. [24]

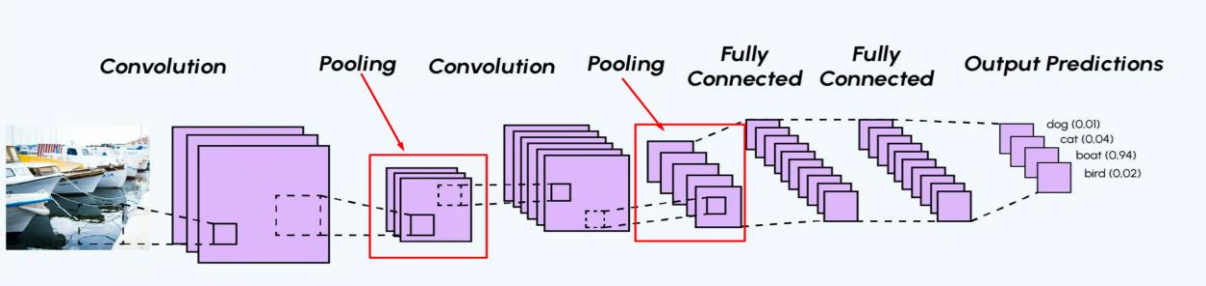

Figure I.20 : Identification des couches de Pooling au sein d'un CNN.

d. Couche entièrement connectée (Fully-Connected)

Après plusieurs couches de convolution et de *pooling*, le raisonnement de haut niveau dans le réseau neuronal s'effectue grâce à des couches entièrement connectées. [60]

Une fois les caractéristiques extraites et affinées par les couches précédentes du CNN, le réseau requiert un mécanisme d'intégration afin de consolider ces informations en une représentation globale. Cette transition est assurée par la couche entièrement connectée (Fully Connected layer). Le processus débute par l'aplatissement (flattening) de l'ensemble des cartes de caractéristiques en un vecteur unidimensionnel unique, lequel synthétise l'intégralité des attributs appris au cours des étapes antérieures. [24]

Le vecteur aplati est ensuite transmis à des unités denses, au sein desquelles chaque neurone est connecté à l'intégralité des composantes du vecteur. Ces connexions permettent au modèle d'appréhender les corrélations complexes entre les différentes caractéristiques extraites. Tandis que les premières couches du CNN se focalisent sur la détection des contours et des formes géométriques, cette strate terminale adopte une perspective globale, dédiée à

l'identification sémantique de l'image, telle que la reconnaissance d'un chiffre, d'un objet ou d'un visage. [24]

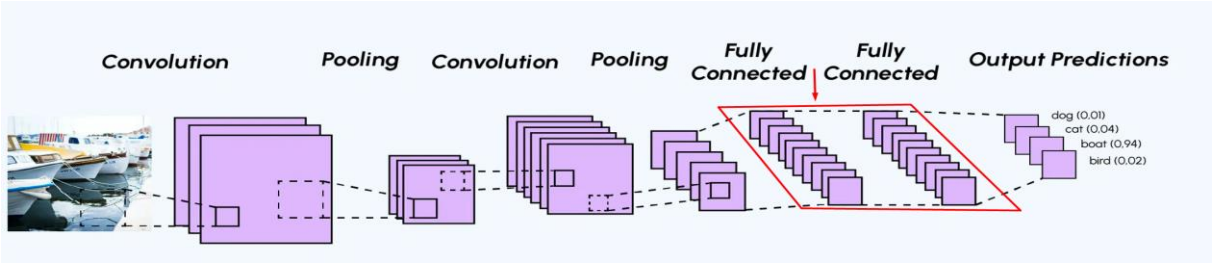

Figure I.21 : Identification des couches de Fully-Connected au sein d'un CNN.

e. Couche de Sortie (Output)

La couche de sortie constitue l'étape terminale de l'architecture d'un réseau de neurones convolutif, niveau auquel s'effectue la prise de décision du modèle. À la suite du traitement des caractéristiques par la couche entièrement connectée, la couche de sortie convertit ces valeurs intermédiaires en probabilités explicites, définissant ainsi le degré d'appartenance de l'image d'entrée à chacune des classes prédéfinies. [24]

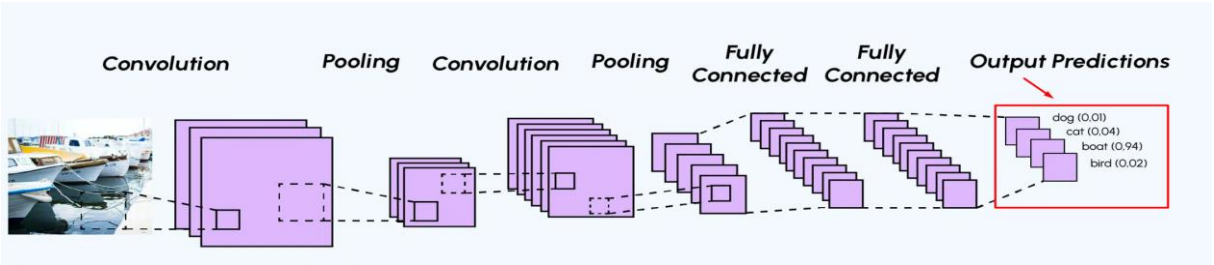

Figure I.22 : Identification des couches de Output au sein d'un CNN.

I.8 État de l'art sur la classification des sons environnementaux

La classification des sons environnementaux (ESC) est un domaine en pleine expansion au carrefour du traitement du signal et de l'intelligence artificielle. Caractérisés par leur grande variabilité et leur nature non stationnaire, ces signaux restent complexes à traiter mais cruciaux pour diverses applications (sécurité, surveillance). Ce domaine a fortement évolué, passant des méthodes classiques d'extraction manuelle vers des architectures d'apprentissage profond basées sur les transformations temps-fréquence. Cette section propose ainsi une revue de la littérature afin de situer notre étude dans ce paysage scientifique.

I.8.1 Les pionniers : CNN sur spectrogrammes

Étude de Piczak — Premier CNN 2D pour l'ESC (2015)

Piczak [70] a été le pionnier de l'application des réseaux convolutifs 2D à la classification des sons environnementaux, posant ainsi les bases d'une nouvelle ère pour l'ESC. Son approche consiste à traiter les représentations temps-fréquence du signal audio comme des images d'entrée pour un CNN, avec les éléments suivants :

- Entrée **:** log-Mel spectrogramme et son delta
- Architecture : deux couches convolutionnelles + deux couches entièrement connectées
- Résultats obtenus :
    - ESC-50 → 64,5 % d'accuracy, surpassant la forêt aléatoire de +20,6 points de pourcentage
- Apports fondamentaux :
    - Première démonstration de la capacité des CNN à apprendre des représentations pertinentes directement à partir de spectrogrammes, sans ingénierie de caractéristiques manuelle
    - Travail fondateur ayant ouvert la voie à l'ensemble des recherches architecturales ultérieures en ESC

I.8.2 Transfert de modèles pré-entraînés sur ImageNet

Étude de Boddapati et al. — Transfer Learning avec AlexNet et GoogLeNet (2017)

Boddapati, Petef, Rasmusson et al. [63] ont été parmi les premiers à explorer le transfert d'apprentissage depuis ImageNet pour la classification des sons environnementaux (ESC), en convertissant les signaux audio en représentations visuelles exploitables par des CNN. Leur étude compare deux architectures sur trois datasets (ESC-50, ESC-10, UrbanSound8K) et trois types de représentations :

- Représentations testées : Spectrogramme, MFCC, Cross Recurrence Plot
- Architectures : AlexNet vs GoogLeNet
- Meilleurs résultats (GoogLeNet + spectrogramme) : 73 % sur ESC-50 · 91 % sur ESC-10 · 93 % sur UrbanSound8K
- Conclusion : Les réseaux récurrents convolutifs n'apportent pas nécessairement de gain dans ce contexte

Étude de Ashurov et al. — Comparaison systématique de modèles et d'optimiseurs (2022)

Ashurov, Zhou, Shi, Zhao et Liu [64] ont mené une analyse approfondie de l'impact combiné du choix d'architecture et d'optimiseur sur les performances de l'ESC par transfert d'apprentissage. Utilisant des log-Mel spectrogrammes comme entrée sur le dataset UrbanSound8K, ils ont évalué plusieurs configurations :

- Architectures testées : VGG, ResNet, Inception, DenseNet
- Optimiseurs testés : Adam, Adamax, RMSprop
- Meilleurs résultats :
    - DenseNet201 → 97,25 % d'accuracy
    - ResNet50V2 → 95,5 % d'accuracy
- Facteurs clés identifiés : le taux d'apprentissage et le nombre d'époques influencent significativement la convergence et la généralisation du modèle

I.8.3 Architectures CNN spécialisées et régularisation

Étude de Mushtaq et Su — DCNN régularisé avec augmentation de données (2021)

Mushtaq et Su [65] ont conçu un réseau convolutionnel profond (DCNN) spécifiquement adapté à l'ESC, en s'affranchissant des couches de max-pooling classiques et en combinant plusieurs stratégies pour contrer le surapprentissage sur les petits datasets. Leur approche repose sur l'utilisation du log-Mel spectrogramme comme entrée, associée à :

- Augmentation de données hors ligne : bruit blanc, étirement temporel, décalage de hauteur
- Régularisation : L2
- Résultats obtenus :
    - ESC-10 → 94,94 % d'accuracy
    - ESC-50 → 89,28 % d'accuracy
    - UrbanSound8K → 95,37 % d'accuracy

Étude de Fang et al. — RACNN : un modèle léger pour environnements embarqués (2022)

Fang, Yin, Du et Huang [66] ont proposé le RACNN (Resource Adaptive Convolutional Neural Network), une architecture légère orientée vers les contraintes matérielles des dispositifs embarqués, sans sacrifier les performances de classification. Le modèle s'appuie sur le log-Mel spectrogramme et intègre plusieurs innovations architecturales :

- Module RAC : génère des cartes de caractéristiques redondantes à faible coût computationnel
- Mécanismes intégrés : attention sur les canaux + connexions résiduelles
- Résultats obtenus :
    - ESC-50 → 86,65 % (± 0,25 %) avec un nombre de paramètres réduit
    - UrbanSound8K → 97,51 % avec seulement 145,7 K paramètres, soit ~7 % de la taille du modèle Two-Stream le plus compact
- Conclusion : démontre la faisabilité d'une ESC efficace sur des systèmes à ressources limitées

I.8.4 Approches end-to-end et CNN 1D

Étude de Abdoli et al. — CNN 1D end-to-end sur signal audio brut (2019)

Abdoli, Cardinal et Koerich [67] ont proposé une approche radicalement différente des méthodes classiques, en éliminant toute étape de transformation préalable du signal audio. Leur modèle CNN 1D apprend directement à partir du signal audio brut dans une architecture entièrement end-to-end, avec plusieurs innovations notables :

- Architecture : plusieurs couches convolutionnelles 1D capturant la structure temporelle fine du signal
- Initialisation spécifique : première couche convolutionnelle initialisée avec un banc de filtres Gammatone, modélisant la réponse cochléaire humaine
- Résultats obtenus :
    - UrbanSound8K → 89 % d'accuracy, surpassant la majorité des méthodes contemporaines basées sur des caractéristiques manuelles ou des représentations 2D
- Avantages clés :
    - Réduction significative du prétraitement nécessaire

- Découverte automatique de filtres pertinents directement à partir des données
- Nombre de paramètres réduit par rapport aux architectures 2D classiques

I.8.5 Synthèse comparative des principales contributions

Le tableau suivant récapitule de manière synthétique les travaux les plus représentatifs sur la classification des sons environnementaux. Il met en évidence l'évolution des performances selon les jeux de données et les méthodes employées, depuis les approches CNN classiques jusqu'aux modèles plus avancés basés sur le transfert d'apprentissage.

Auteurs	Année	Dataset	Méthode	Accuracy
Piczak	2015	ESC-50	CNN 2D (log-Mel)	64,5 %
Boddapati et al.	2017	ESC-50	GoogLeNet (spectrogramme)	73 %
Abdoli et al.	2019	UrbanSound8K	CNN 1D end-to-end	89 %
Mushtaq et Su	2021	ESC-50	DCNN régularisé (log-Mel)	89,28 %
Fang et al.	2022	ESC-50	RACNN (log-Mel)	86,65 %
Ashurov et al.	2022	UrbanSound8K	DenseNet201 (log-Mel)	97,25 %

Tableau I.3 : Récapitulatif des principaux travaux sur la classification des sons environnementaux

I.9 Conclusion

Ce chapitre a permis de poser les bases théoriques nécessaires à notre étude sur la classification des sons environnementaux. Nous y avons présenté les notions essentielles liées à l'intelligence artificielle, à l'apprentissage automatique, à l'apprentissage profond, ainsi qu'aux réseaux de neurones artificiels et convolutifs, qui constituent le socle méthodologique de ce travail.

Dans le chapitre suivant, nous passerons à l'aspect expérimental en nous intéressant à la base de données ESC-50, à sa préparation, ainsi qu'aux étapes de prétraitement et d'augmentation des données. Nous y présenterons également les deux modèles pré-entraînés retenus pour notre étude, afin de préparer le terrain aux tests et aux comparaisons de performances qui seront développés par la suite.

Chapitre II

Chapitre 2 : Base de données et modèles de classification.

II.1 Introduction

Dans ce chapitre, nous nous intéressons aux deux piliers fondamentaux de tout système d'apprentissage supervisé : les données et les modèles. Depuis la construction de la base de données jusqu'au choix des modèles de classification. Nous mettons en évidence les étapes essentielles qui ont permis de préparer les données et de définir les conditions expérimentales de cette étude.

II.2 Description de la base de données

La base de données primaire utilisée dans ce travail est constituée initialement d'enregistrements audio au format WAV (Waveform Audio File Format), un format non compressé offrant une haute fidélité sonore, parfaitement adapté aux traitements numériques du signal audio.

Ces enregistrements couvrent 26 types de bruits sonores environnementaux distincts, issus de divers contextes sonores urbains, naturels et domestiques. Chaque classe contient 40 fichiers audio, ce qui porte le nombre total d'enregistrements à 1 040 fichiers WAV.

La base de données utilisée est ESC-50 (Environmental Sound Classification), constituée de 2000 enregistrements audio répartis en 50 classes. [12]

Dans le cadre de notre étude, nous avons sélectionné 26 classes pertinentes correspondant aux bruits environnementaux urbains ciblés.

II.3 Prétraitement sur la base de données

Afin de rendre nos signaux audios exploitables par des modèles de vision par ordinateur, nous avons procédé à leur transformation en images via les deux types de représentations temps-fréquence, donnant lieu à deux bases de données visuelles indépendantes : la première constituée de spectrogrammes, et la seconde de Mel-spectrogrammes.

Chaque fichier audio a été converti en une image de 224 × 224 pixels au format RGB. Cette résolution a été choisie afin d'assurer une compatibilité optimale avec les modèles de classification pré-entraînés utilisés dans ce travail, qui requièrent une taille d'entrée standardisée. Chacune des deux bases de données regroupe ainsi l'ensemble des

représentations visuelles générées, organisées selon les 26 classes de bruits environnementaux constituant notre corpus.

Ces représentations ont ensuite été soumis à un ensemble de traitements et de transformations telles que techniques d'augmentation de données qui seront détaillés dans les sections suivantes.

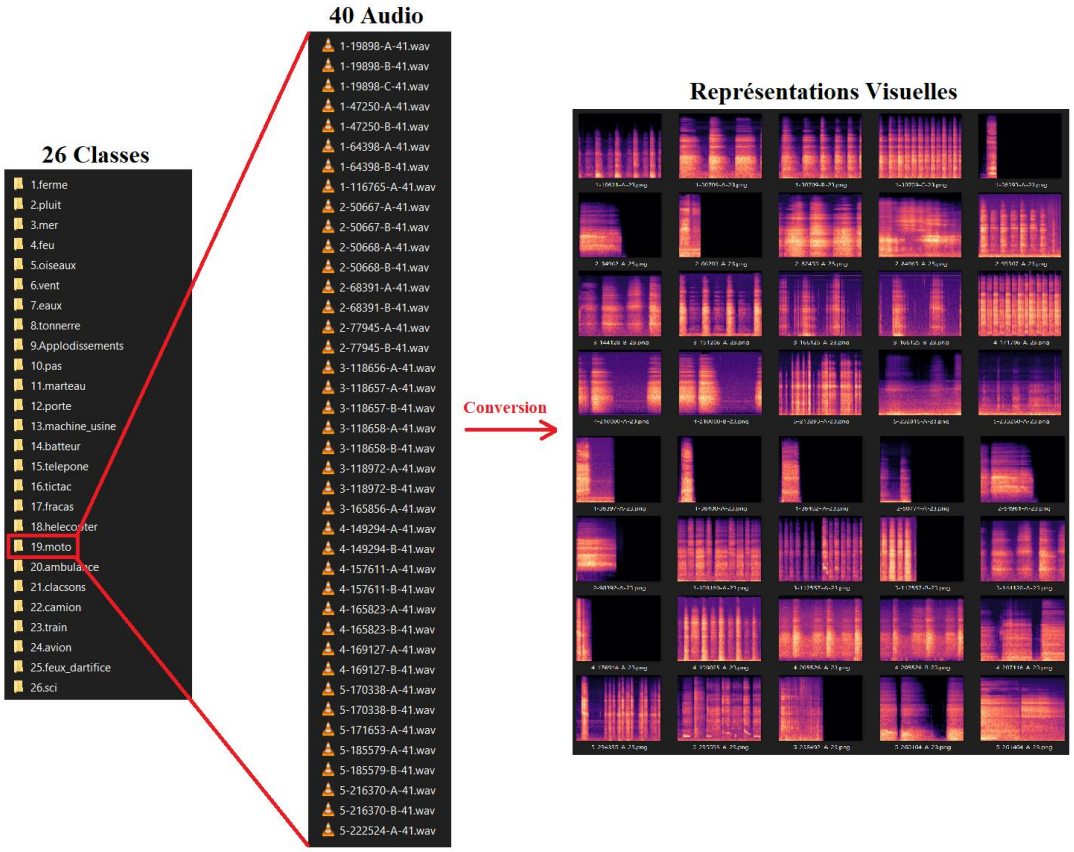

Figure II.1 Pipeline de conversion de la base de données audio en représentations visuelles

II.3.1 Spectrogramme (Spectrogram) :

a Transformée de Fourier à court terme (STFT)

Le spectrogramme est fondé sur la transformée de Fourier à court terme (STFT, Short-Time Fourier Transform), qui décompose un signal temporel en segments courts afin d'analyser son contenu fréquentiel local. Soit x[n] un signal discret, w[n] une fenêtre d'analyse de longueur N (par exemple Hamming ou Hann), et H le pas d'avancement (hop size). La STFT discrète est définie par l'équation suivante [68] :

$$X(m,k) = \sum_{n=0}^{N-1} x[n] \cdot w[n-mH] \cdot e^{-j2\pi kn/N} \quad \text{.....(II.1)}$$

où m indexe la trame temporelle, k le bin fréquentiel, et j est l'unité imaginaire. La fenêtre w[n] isole un segment temporel sur lequel on applique la transformée de Fourier discrète (DFT). Le choix de N régit le compromis entre résolution temporelle et résolution fréquentielle : une fenêtre courte offre une meilleure résolution temporelle, tandis qu'une fenêtre longue améliore la résolution fréquentielle [69].

b Définition du spectrogramme

Le spectrogramme est généré par le découpage du signal audio en segments temporels de courte durée, généralement de l'ordre de quelques millisecondes, suivis de l'application successive de la transformée de Fourier discrète sur chacun d'eux afin d'en extraire le contenu fréquentiel. Les spectres ainsi obtenus sont ensuite disposés chronologiquement le long de l'axe temporel pour composer la représentation finale du spectrogramme. [8]

Le spectrogramme est le carré du module de la STFT. Il fournit une représentation bidimensionnelle de la densité spectrale de puissance du signal en fonction du temps et de la fréquence[68] :

$$S[m,k] = |X[m,k]|^2 \quad \text{.....(II.2)}$$

Le spectrogramme constitue la représentation temps-fréquence la plus couramment utilisée en classification des sons environnementaux, car il permet de capturer les variations spectrales caractéristiques de chaque classe sonore [70]. Toutefois, il présente une résolution linéaire en fréquence, ce qui ne reflète pas la perception auditive humaine, plus fine dans les basses fréquences que dans les hautes fréquences.

Dans notre travail, nous avons utilisé le spectrogramme comme première représentation visuelle des signaux sonores. Il représente le signal de manière bidimensionnelle, où l'axe horizontal correspond au temps, l'axe vertical à la fréquence (en Hz), et l'intensité des couleurs à l'amplitude (en dB). Chaque signal audio de notre base de données a été converti en une image de **224 × 224 pixels.**

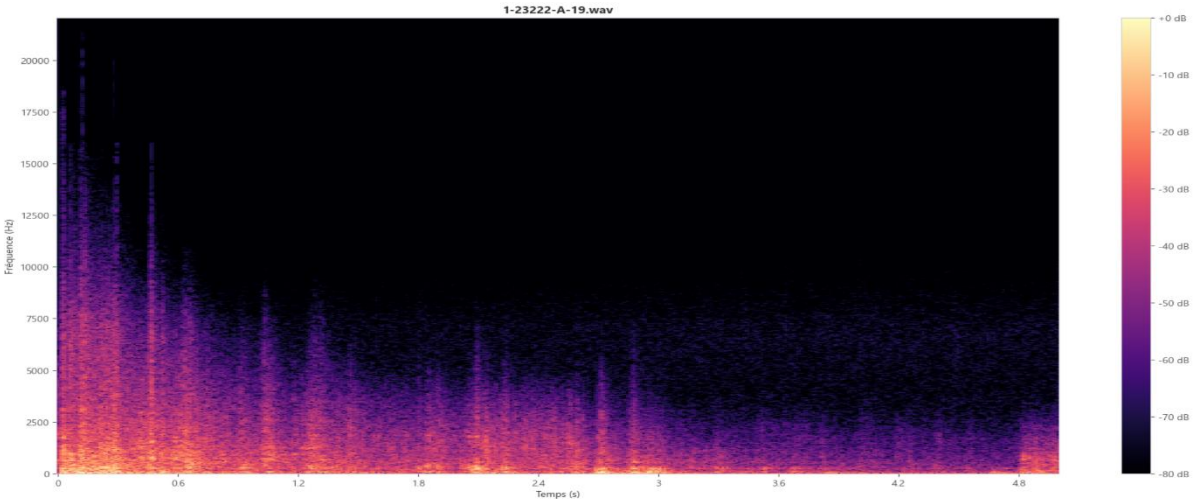

Figure II.2 Exemple de spectrogramme généré à partir d'un fichier audio de notre base de données

II.3.2 Mel-Spectrogramme (Mel-Spectrogram)

II.3.2.a Échelle mel

L'échelle mel est une échelle psychoacoustique de hauteur, proposée initialement par Stevens, Volkmann et Newman (1937)[71], telle que des intervalles de hauteur égaux en mels correspondent à des distances perceptives égales pour l'auditeur humain. La conversion de la fréquence en hertz (f) vers l'échelle mel (m) est donnée par la formule couramment adoptée dans le standard HTK[72] :

$$\mathrm{m} = 2595 \cdot log_{10}(1 + \frac{f}{700}) \quad \text{.....(II.3)}$$

La transformation inverse, permettant de convertir une valeur mel en fréquence linéaire, s'écrit :

$$f = 700 \cdot (10^{\frac{m}{2595}} - 1) \quad \text{.....(II.4)}$$

On remarque que cette échelle est approximativement linéaire en dessous de 1 kHz et logarithmique au-delà, conformément aux observations psychoacoustiques sur la perception humaine des hauteurs[71].

b Banc de filtres mel

Un banc de filtres mel est un ensemble de M filtres triangulaires espacés uniformément sur l'échelle mel, chacun capturant l'énergie spectrale dans une bande fréquentielle donnée[72]. Le m'-ième filtre triangulaire est défini par :

$$H_{m'}[\mathrm{k}] = \max\left(0{,}1 - \frac{|f[k]-f_c(m')|}{\Delta \mathrm{f}(\mathrm{m}')}\right) \text{ ……..…….(II.5)}$$

où f[k] est la fréquence du k-ième bin spectral, fc(m') est la fréquence centrale du filtre, et Δf(m') est la largeur de la bande passante du filtre. Ce banc de filtres réalise une compression non linéaire du spectre, en concentrant la résolution dans les basses fréquences et en réduisant la résolution dans les hautes fréquences, mimant ainsi la sensibilité de l'oreille humaine[73].

c Définition du mel-spectrogramme

Le spectrogramme Mel constitue une variante spécifique du spectrogramme, largement exploitée dans les domaines du traitement de la parole et de l'apprentissage automatique. À l'instar d'une représentation temps-fréquence classique, il retrace l'évolution du contenu fréquentiel d'un signal acoustique au cours du temps, tout en transposant les données sur une échelle de fréquences distincte. [8]

Le mel-spectrogramme est obtenu en appliquant le banc de filtres mel au spectrogramme de puissance. Pour chaque trame temporelle m, l'énergie dans chaque bande mel m' est calculée par sommation pondérée [72] :

$$M_{mel}[m, m'] = \sum_{k=0}^{K-1} \mathrm{S}[\mathrm{m}, \mathrm{k}] \cdot H_{m'}[\mathrm{k}] \text{ ……..……..……..(II.6)}$$

En pratique, on utilise généralement le logarithme du mel-spectrogramme de puissance (log-mel-spectrogramme) afin de compresser la plage dynamique et de se rapprocher de la perception auditive logarithmique de l'intensité [74] :

$$logS_{mel}[m, m'] = \log\left(M_{mel}[m, m'] + \in\right) \text{ ……..…….(II.7)}$$

où ε est une petite constante ajoutée pour éviter le logarithme de zéro. Le mel-spectrogramme constitue aujourd'hui la représentation de référence en classification des sons environnementaux par réseaux de neurones convolutifs [70], car il offre une représentation compacte et perceptuellement pertinente du signal audio.

Nous avons également adopté le Mel-spectrogramme comme deuxième représentation, qui constitue une variante du spectrogramme dans laquelle nous convertissons l'axe des fréquences vers l'échelle de Mel, une échelle perceptuelle non linéaire modélisant la sensibilité de l'oreille humaine, offrant ainsi une meilleure représentation des caractéristiques

acoustiques de nos signaux. images générées présentent également des dimensions de **224 × 224 pixels.**

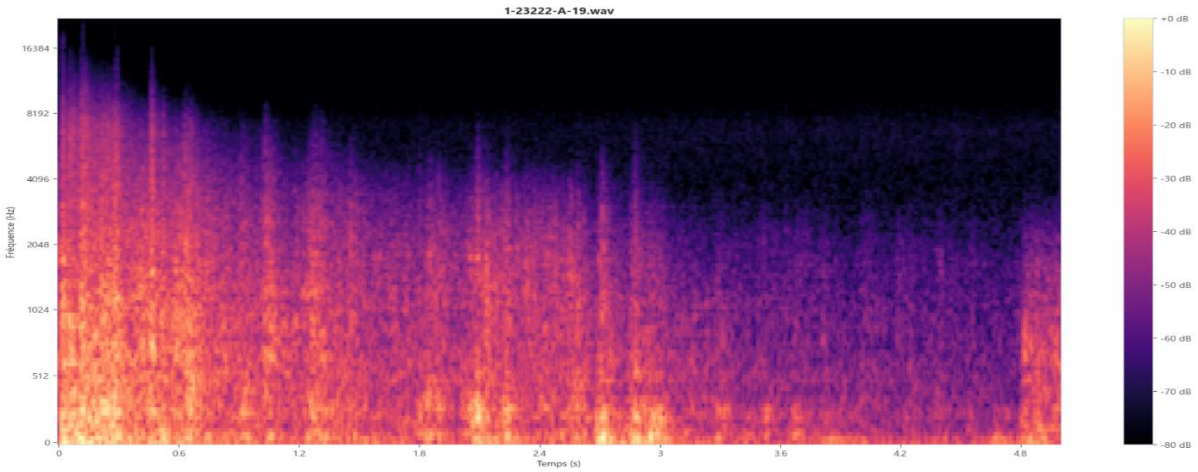

Figure II.3 Exemple de Mel-Spectrogramme généré à partir d'un fichier audio de notre base de données

II.4 Présentation des classes de la base de données

Dans cette section, nous présentons les **26 classes** de bruits sonores environnementaux constituant notre base de données. Pour chaque classe, nous illustrons trois représentations : la forme d'onde temporelle (1D), le spectrogramme et le Mel-spectrogramme correspondants.

Classe 1 : Ferme

Sons issus d'un environnement agricole, incluant les bruits d'animaux de la ferme et les activités rurales.

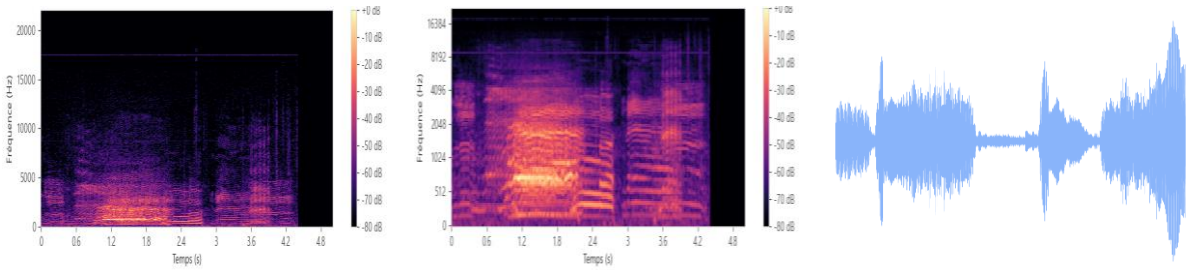

Figure II.4: Représentations de la classe Ferme — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 2 : Pluie

Sons de précipitations atmosphériques, caractérisés par un bruit continu et large bande.

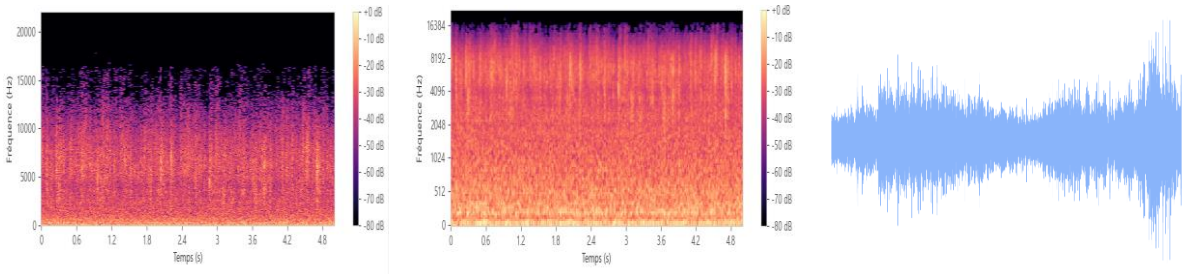

Figure II.5 : Représentations de la classe Pluie — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 3 : Mer

Sons des vagues marines et du bord de mer, présentant un caractère rythmique et répétitif.

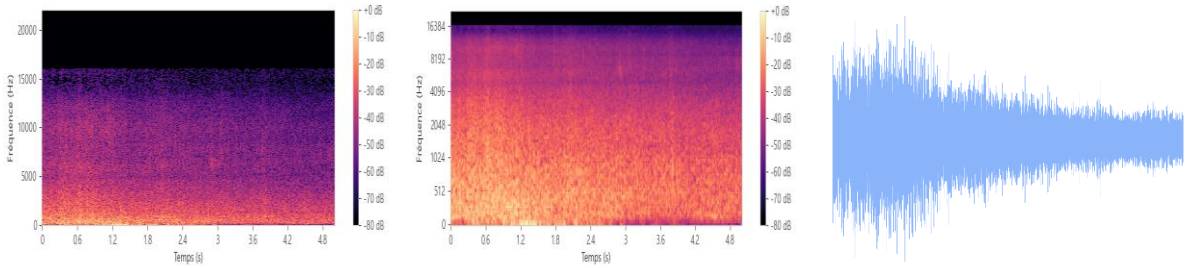

Figure II.6 : Représentations de la classe Mer — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 4 : Feu

Sons de combustion et de crépitement du feu, caractérisés par des impulsions irrégulières.

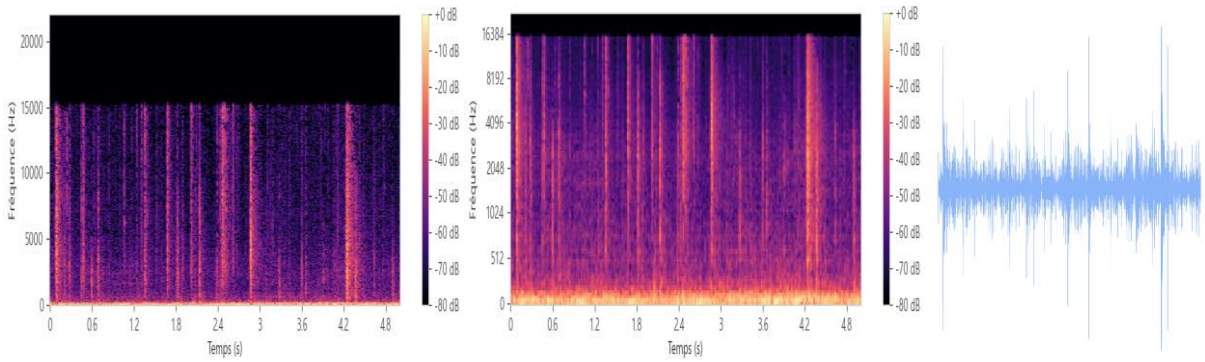

Figure II.7 : Représentations de la classe Feu — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 5 : Oiseaux

Sons de chants et vocalisations d'oiseaux, présentant des motifs fréquentiels riches et variés.

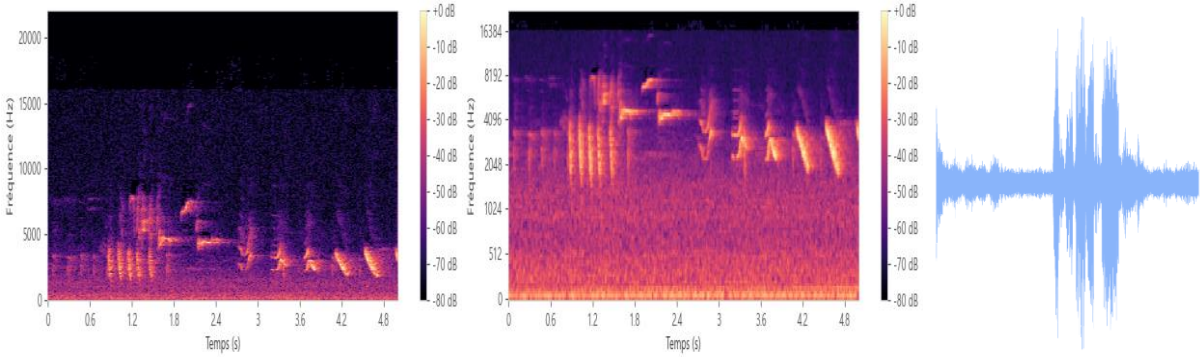

Figure II.8 : Représentations de la classe Oiseaux — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 6 : Vent

Sons du vent en mouvement, caractérisés par un bruit de fond à basses fréquences dominant.

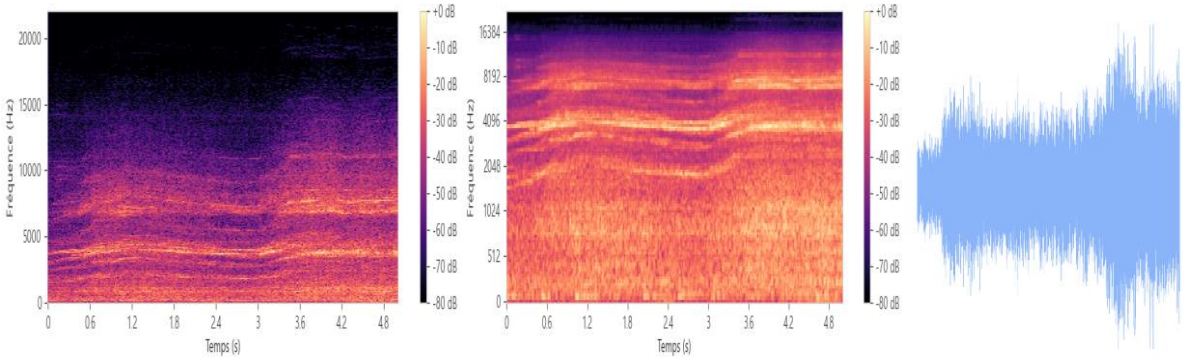

Figure II.9 : Représentations de la classe Vent — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 7 : Eaux

Sons liés à l'eau dans un contexte quotidien, tels que l'écoulement d'un robinet.

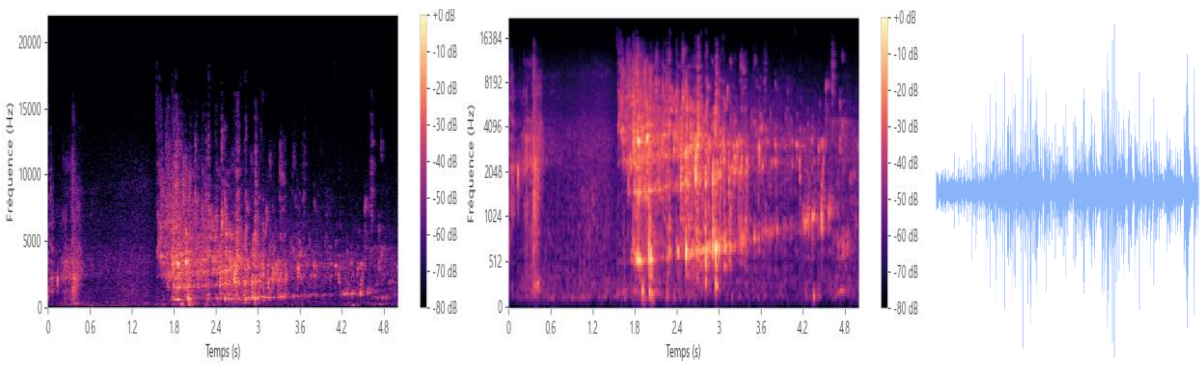

Figure II.10 : Représentations de la classe Eux — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 8 : Tonnerre

Sons d'orages et de tonnerres, caractérisés par de fortes impulsions à large spectre fréquentiel.

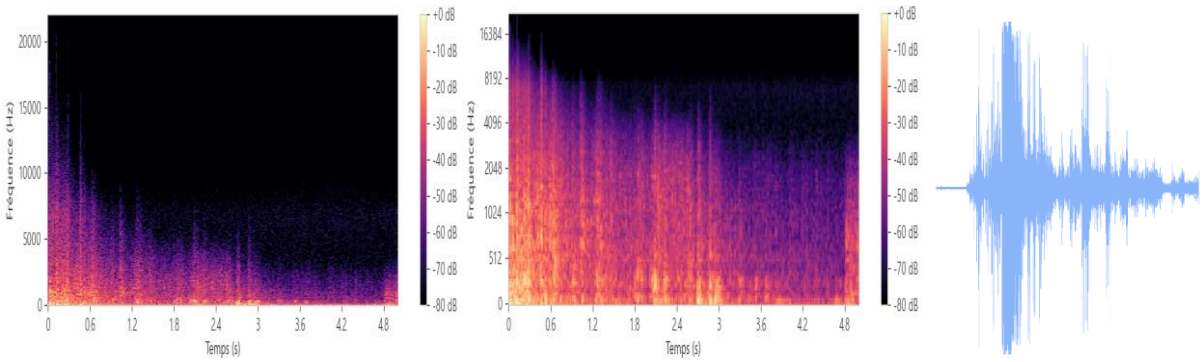

Figure II.11 : Représentations de la classe Tonnerre — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 9 : Applaudissements

Sons d'applaudissements humains, présentant des impulsions percussives répétitives.

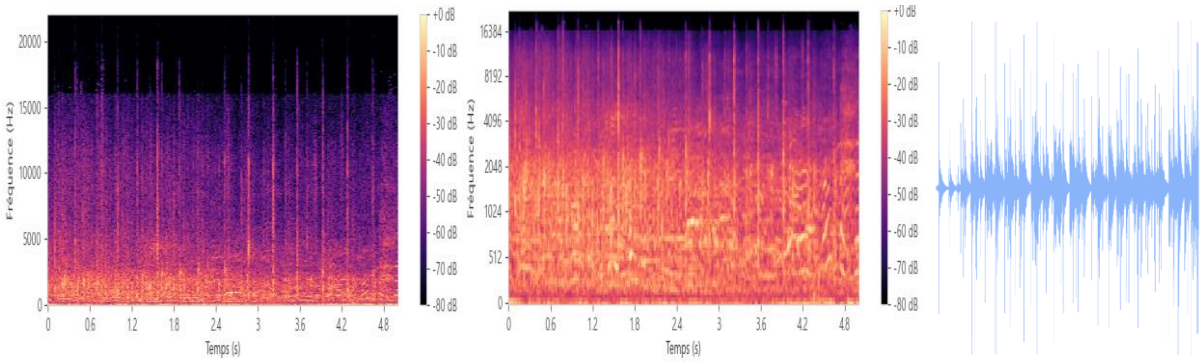

Figure II.12 : Représentations de la classe Applaudissements — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 10 : Pas

Sons de pas réguliers, formés d'impulsions périodiques sur différentes surfaces.

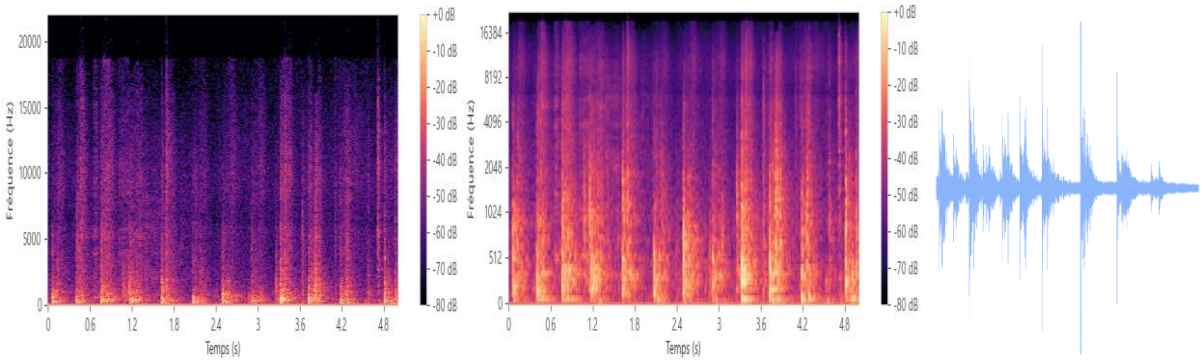

Figure II.13 : Représentations de la classe Pas — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 11 : Marteau

Sons percussives répétitives sur surfaces solides, a impulsions brèves et rythmiques.

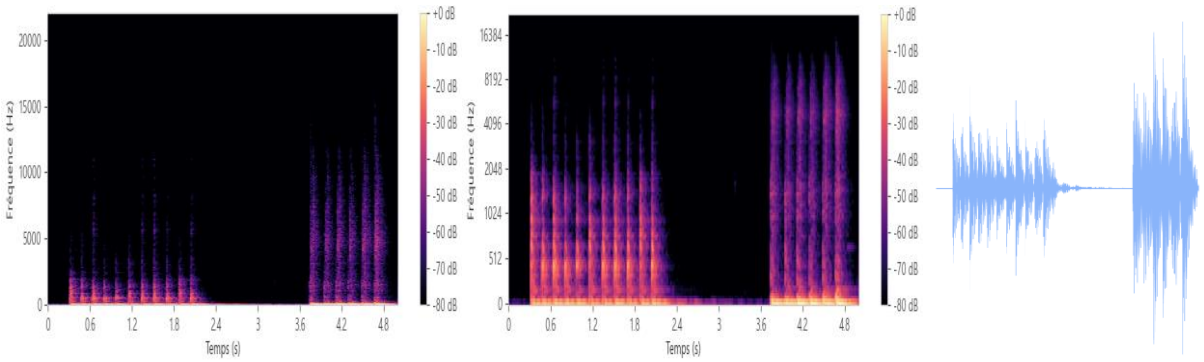

Figure II.14 : Représentations de la classe Marteau — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 12 : Porte

Sons d'ouverture, de fermeture et de claquement de portes, de nature impulsive et brève.

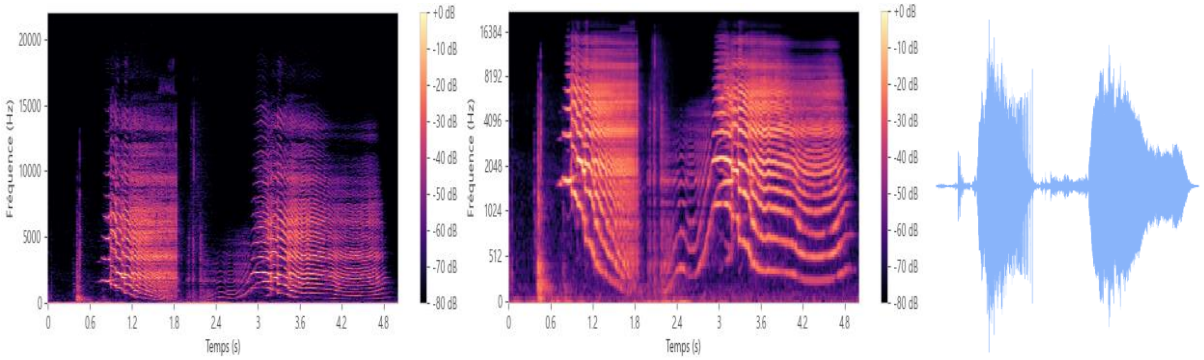

Figure II.15 : Représentations de la classe Porte — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 13 : Machine usine

Sons de machines industrielles, caractérisés par des vibrations mécaniques continues.

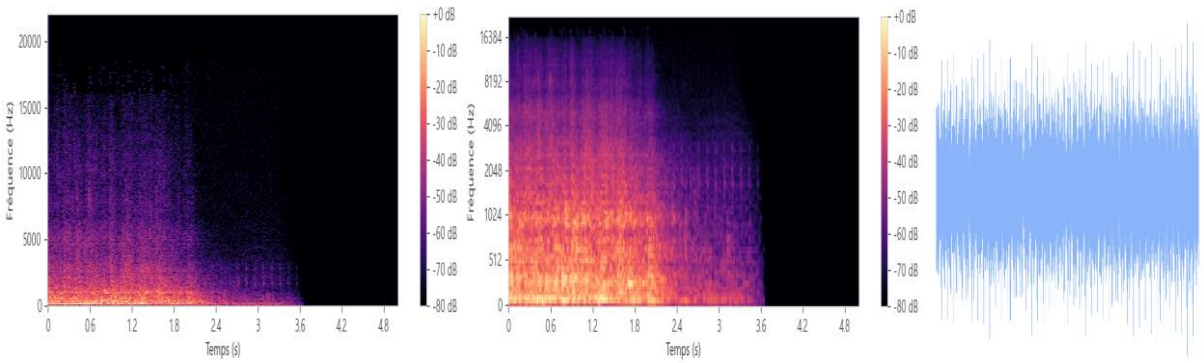

Figure II.16 : Représentations de la classe Machine Usine — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 14 : Batteur

Sons mécaniques rotatifs, continus et rythmiques.

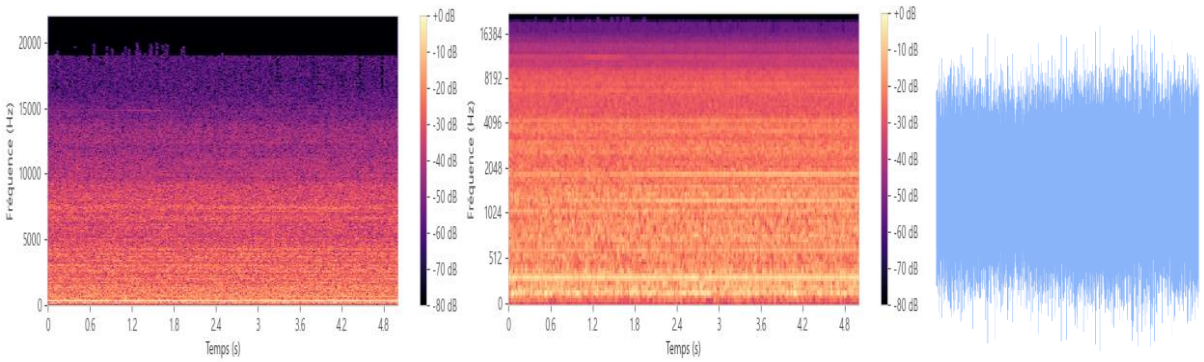

Figure II.17 : Représentations de la classe Batteur — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 15 : Téléphone

Sons de sonneries téléphoniques, caractérisés par des tonalités répétitives et régulières.

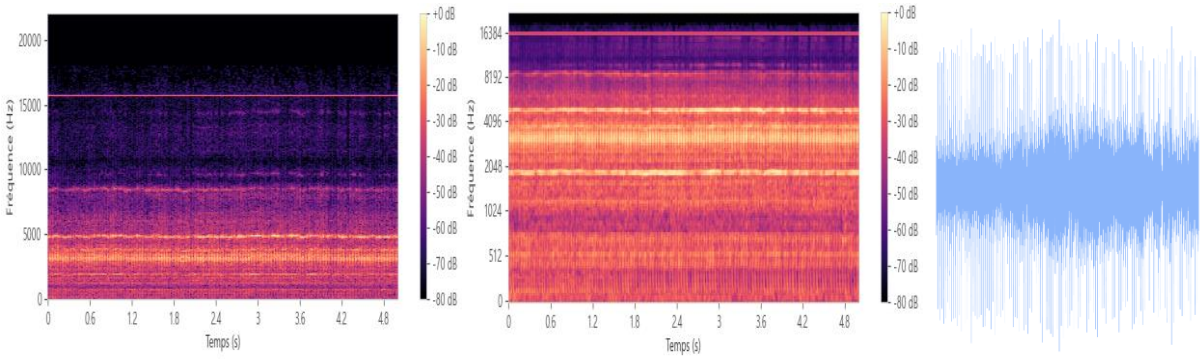

Figure II.18 : Représentations de la classe Téléphone — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 16 : Tic-tac

Sons d'horloges mécaniques, caractérisés par des impulsions régulières et répétitives.

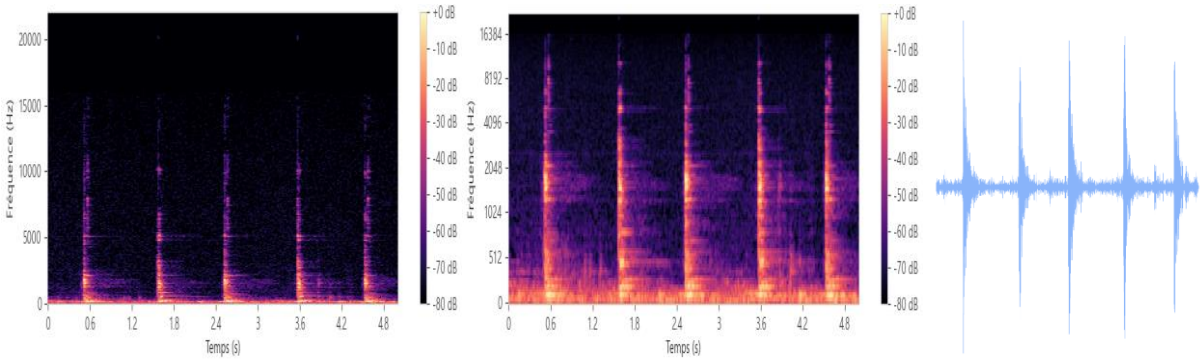

Figure II.19 : Représentations de la classe Tic-Tac — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 17 : Fracas

Sons de chocs et de fractures d'objets, présentant des impulsions larges bandes soudaines.

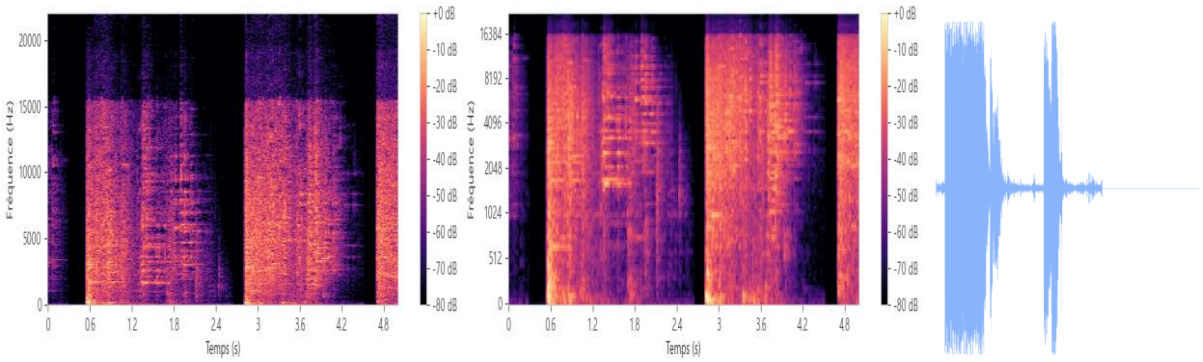

Figure II.20 : Représentations de la classe Fracas — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 18 : Hélicoptère

Sons de rotors d'hélicoptère, caractérisés par un bruit mécanique régulier et répétitif.

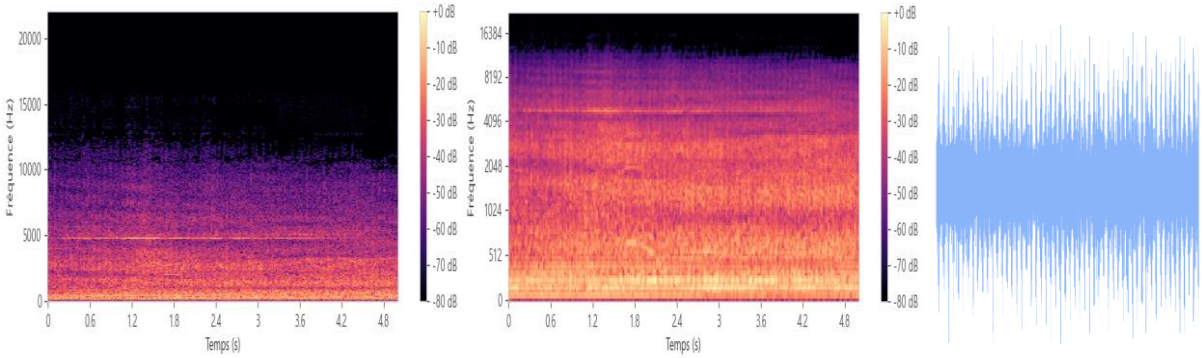

Figure II.21 : Représentations de la classe Helicoptère — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 19 : Moto

Sons de moteurs de motos à basses et moyennes fréquences.

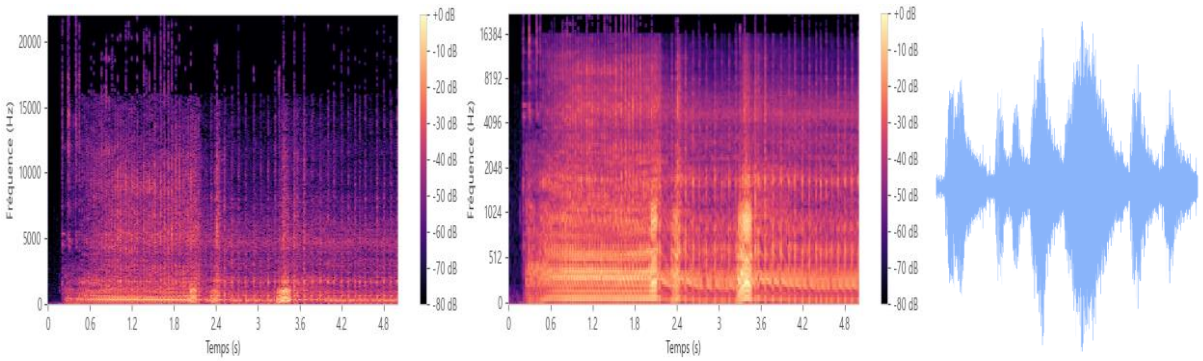

Figure II.22 : Représentations de la classe Moto — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 20 : Ambulance

Sons de sirènes et d'alarmes, caractérisés par une modulation de fréquence répétitive.

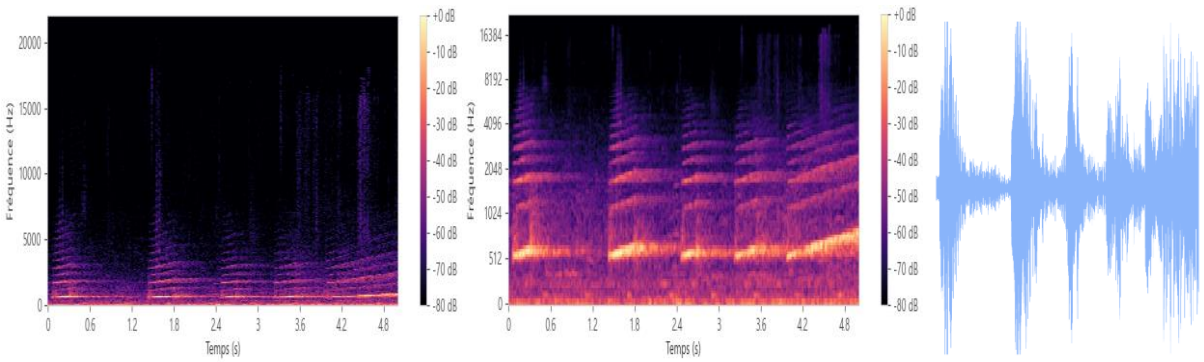

Figure II.23 : Représentations de la classe Ambulance — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 21 : Klaxons

Sons de klaxons de véhicules, courts et intenses à fréquences moyennes.

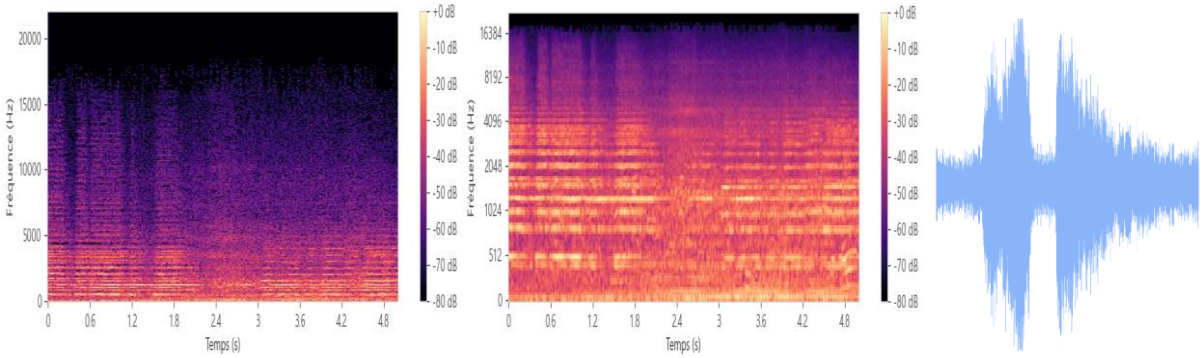

Figure II.24 : Représentations de la classe Klaxons — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 22 : Camion

Sons de moteurs de véhicules, caractérisés par des vibrations continues à basses fréquences.

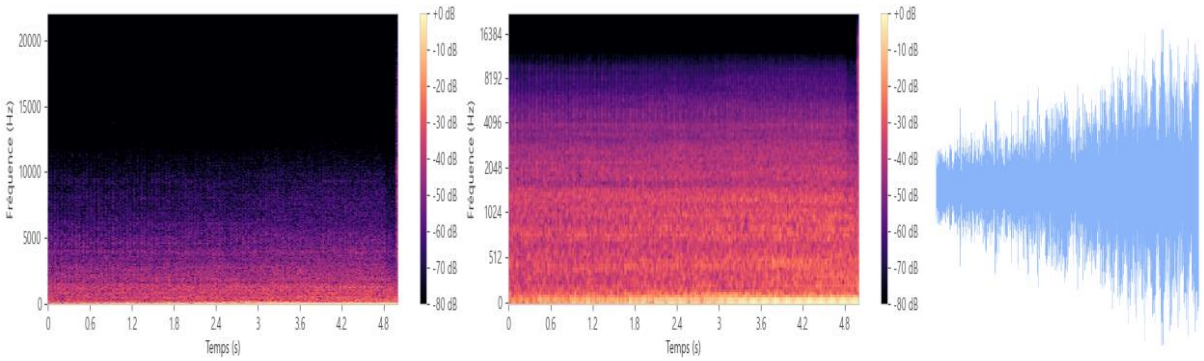

Figure II.25 : Représentations de la classe Camion — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 23 : Train

Sons de trains en circulation, bruit mécanique continu aux modulations rythmiques.

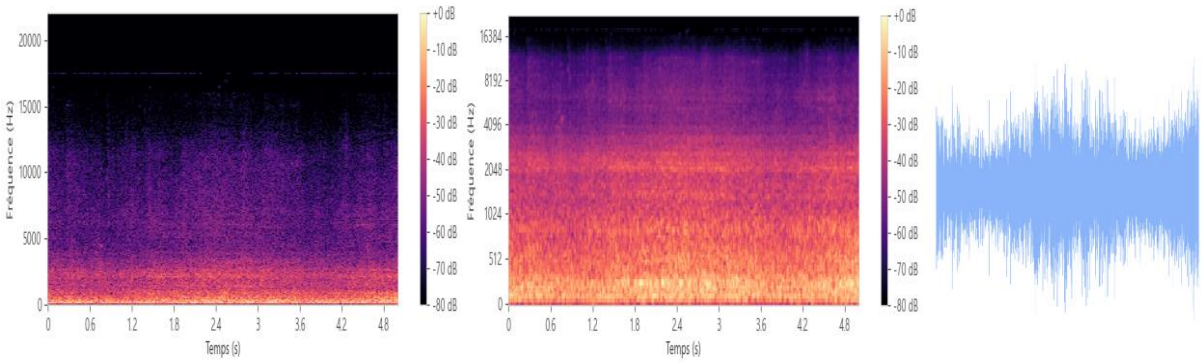

Figure II.26 : Représentations de la classe Train — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 24 : Avion

Sons de réacteurs d'avion, caractérisés par un bruit large bande continu à haute intensité.

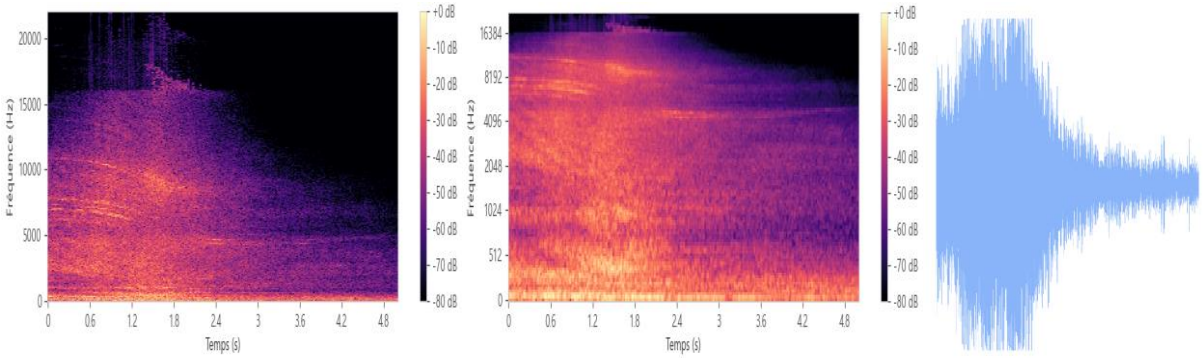

Figure II.27 : Représentations de la classe Avion — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 25 : Feux d'artifice

Sons d'explosions de feux d'artifice, impulsions soudaines et intenses à large spectre.

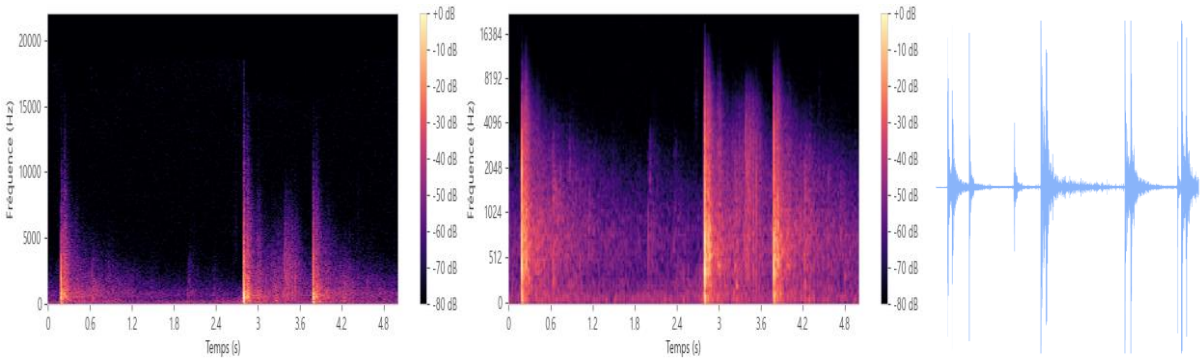

Figure II.28 : Représentations de la classe Faux d'artifice — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

Classe 26 : Scie

Sons de scie manuelle, frottements répétitifs et rythmiques sur surface solide.

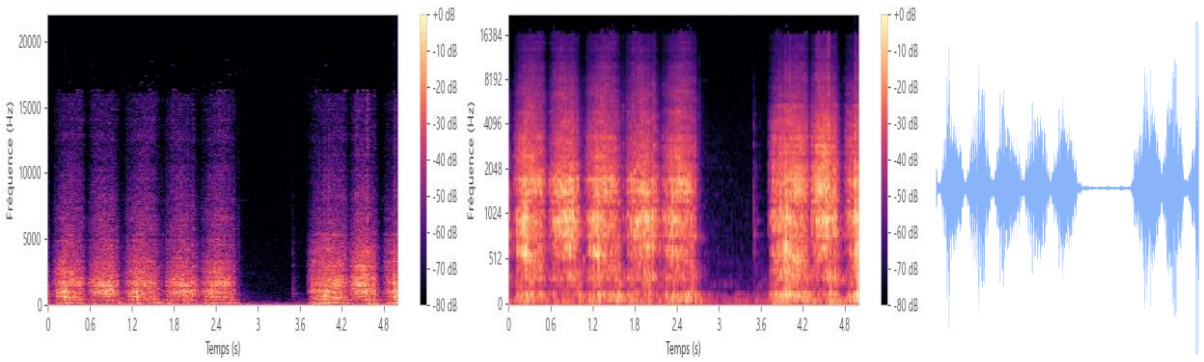

Figure II.29 : Représentations de la classe Scie — (a) Spectrogramme, (b) Mel-Spectrogramme, (c) Forme d'onde 1D

II.5 Augmentation de la base de données

Afin d'enrichir notre base de données et d'améliorer la capacité de généralisation de nos modèles, nous avons appliqué des techniques d'augmentation à deux niveaux distincts : au niveau des signaux audio bruts (WAV's), puis au niveau des représentations visuelles obtenues (Spectrogrammes et Mel-Spectrogrammes).

II.5.1 Augmentation au niveau audio

Dans un premier temps, nous avons appliqué une augmentation directement sur les fichiers audio au format WAV. Pour chaque enregistrement original, nous avons généré une version augmentée par l'ajout d'un bruit artificiel (ajout de bruit gaussien) d'une intensité de 10 dB, simulant ainsi des conditions d'enregistrement réelles et dégradées. Cette opération a permis de doubler le nombre de fichiers audio par classe, passant de 40 fichiers originaux à 80 fichiers audio par classe, soit un total de 2 080 fichiers WAV.

Afin d'illustrer l'effet de cette augmentation, la figure ci-dessous présente un exemple d'enregistrement audio, représenté sous forme de spectrogramme et de Mel-spectrogramme, accompagné de leurs versions augmentées après ajout de bruit.

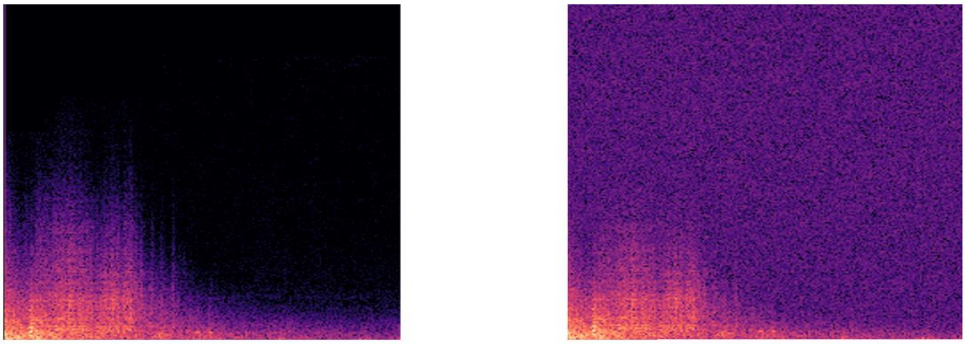

Figure II.30 : Effet de l'ajout de bruit sur le spectrogramme

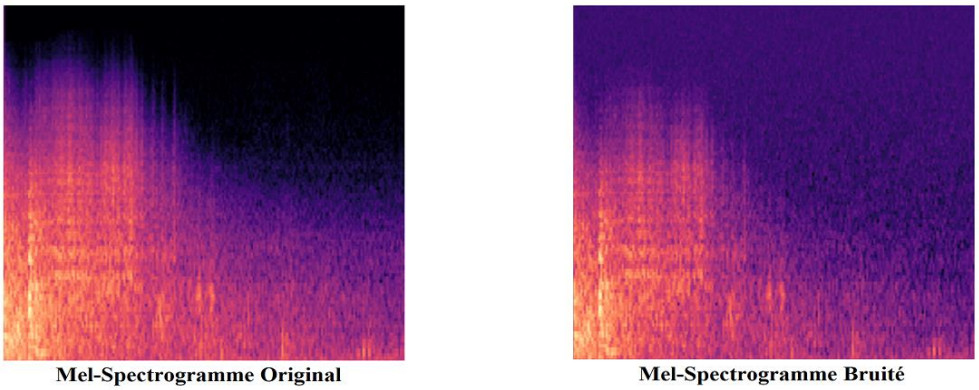

Figure I.31 : Effet de l'ajout de bruit sur le Mel-spectrogramme

On observe clairement l'effet du bruit ajouté sur les deux représentations, se manifestant par une perturbation visible de l'intensité spectrale, notamment dans les zones à faible énergie.

II.5.2 Augmentation au niveau image

Afin d'accroître davantage la taille et la diversité de notre base de données, nous avons appliqué une seconde augmentation directement sur les images obtenues. Pour chaque image, nous avons généré 4 versions transformées selon les opérations suivantes :

- Assombrissement (Darkening) — réduction de la luminosité de l'image de 10%
- Éclaircissement (Brightening) — augmentation de la luminosité de l'image de 10%
- Floutage (Blurring) — application d'un filtre de flou gaussien sur l'image
- Ajout de bruit (Noise Addition) — injection aléatoire de bruit gaussien dans les pixels de l'image

Chaque image originale a ainsi donné lieu à 4 images augmentées supplémentaires, portant le total à 80 × 4 = 320 images augmentées, auxquelles s'ajoutent les 80 images originales, pour atteindre 400 images par classe.

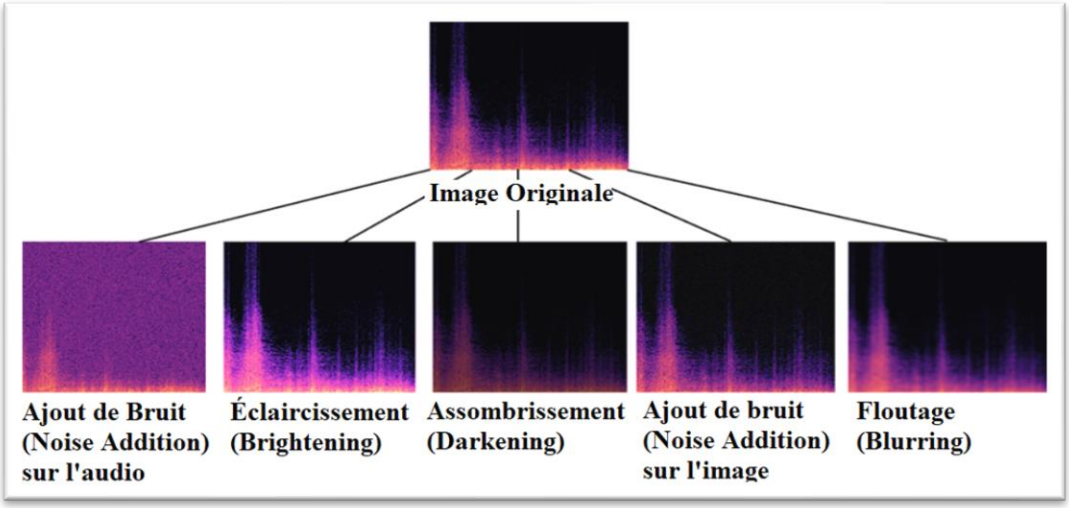

Figure II.32 : Représentations des Techniques d'augmentation de données appliquées aux images

II.6 Bilan final de la base de données

À l'issue de l'ensemble des opérations de préparation, de transformation et d'enrichissement des données, chacune de nos deux bases de données — la base de spectrogrammes et la base de Mel-spectrogrammes — se compose de 10 400 images, réparties équitablement en 400 images par classe sur les 26 classes.

II.7 Les modèles CNN utilisés

Les réseaux de neurones convolutifs sont largement exploités en vision par ordinateur, domaine au sein duquel ils se sont imposés comme la référence pour de nombreuses tâches, notamment la classification d'images. Ces architectures ont également démontré leur efficacité en traitement automatique du langage naturel, en particulier pour les applications liées à la classification de textes. [13]

Dans le cadre de ce travail, et après plusieurs expérimentations, nous avons retenu deux modèles pré-entraînés : EfficientNetV2S et ConvNeXtBase. Ces deux architectures ont été utilisées en exploitant des connaissances acquises lors d'un entraînement préalable sur un large ensemble d'images, afin de les adapter à notre tâche de classification des sons environnementaux.

II.7.1 EfficientNetV2S

EfficientNetV2-S constitue une architecture avancée de réseau de neurones convolutif, particulièrement reconnue pour son efficacité computationnelle et sa capacité d'échelonnage optimale, permettant ainsi d'atteindre des performances élevées tout en limitant le coût des calculs. Ce modèle est ensuite affiné à partir d'un jeu de données préalablement prétraité et équilibré. Notre démarche méthodologique s'articule autour de plusieurs phases clés : le prétraitement des données, la sélection et l'ajustement fin (*fine-tuning*) du modèle, ainsi que l'évaluation rigoureuse de ses performances. [17]

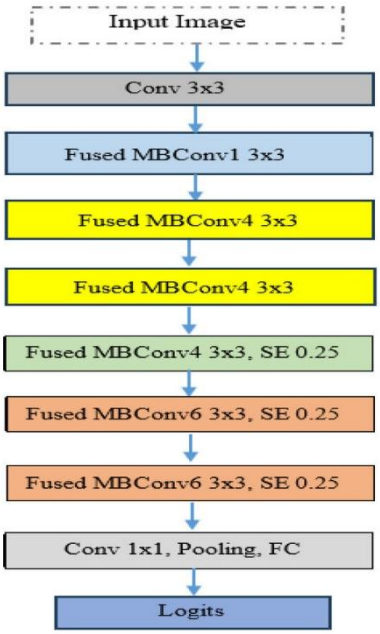

Figure II.33 : Structure des étapes (Stages) de l'architecture EfficientNetV2S

Couche Stem : La phase initiale repose sur une couche de convolution 3x3, couplée à une normalisation par lots (*batch normalization*) et à une fonction d'activation Swish. Cette étape préliminaire assure le conditionnement de l'image d'entrée en vue de son traitement par les strates ultérieures.

Blocs et étapes (*Stages*) : Le réseau présente une structure séquentielle articulée en plusieurs étapes, chacune intégrant un nombre déterminé de blocs *fused-MBConv* et *MBConv*. Ces différentes phases se caractérisent par des variations de la profondeur des canaux et de la typologie des blocs sous-jacents, optimisant ainsi l'extraction hiérarchique des caractéristiques à différents niveaux d'abstraction.

Couches finales : Au terme de la succession des blocs *MBConv* et *fused-MBConv*, l'architecture converge vers une couche de pooling moyen global (*global average pooling*), suivie d'une couche entièrement connectée (*Fully Connected*), pour se conclure par une fonction d'activation Softmax dédiée à la classification finale. [18]

II.7.2 ConvNeXtBase

Le modèle ConvNeXt-Base constitue la configuration la plus volumineuse et la plus évoluée exploitée au sein de cette étude. Il a été spécifiquement conçu pour le traitement de volumineux ensembles de données caractérisés par des attributs complexes et de forte dimensionnalité. Son architecture profonde, soutenue par une quantité substantielle de paramètres et de couches, lui confère l'aptitude d'apprendre des corrélations et des propriétés subtiles. Cette spécificité en fait un choix particulièrement idoine pour les applications exigeant un haut degré de précision et une robustesse accrue.

Bien que ce modèle implique des ressources computationnelles conséquentes ainsi que des phases d'apprentissage prolongées, sa capacité élevée à capturer des motifs tant locaux que globaux justifie son déploiement lorsque l'exactitude et la fiabilité demeurent prioritaires. ConvNeXt-Base incarne ainsi le seuil de performance le plus élevé parmi les variantes étudiées, se distinguant de manière optimale dès lors que la dimension des données et la puissance de calcul ne représentent pas des facteurs limitants. [47]

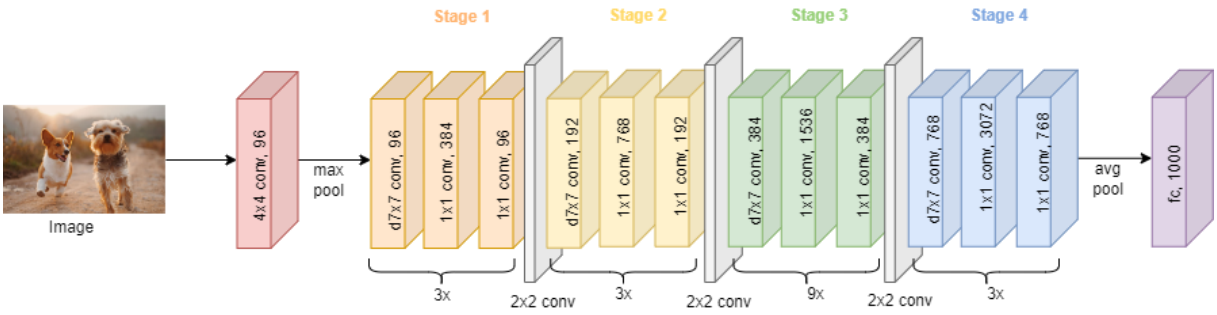

Figure II.34 : Structure des étapes (Stages) de l'architecture ConvNeXtBase

II.8 Conclusion

Ce chapitre a été consacré principalement à la constitution et à l'exploitation de la base de données utilisée dans cette étude. Nous avons d'abord présenté la base ESC-50, en précisant le choix des 26 classes retenues, puis nous avons expliqué le passage des signaux audio bruts vers des représentations visuelles adaptées, à savoir les spectrogrammes et les

Mel-spectrogrammes. Nous avons également détaillé les différentes techniques d'augmentation appliquées aux données, aussi bien au niveau audio qu'au niveau image, afin d'enrichir la base et d'améliorer la robustesse des modèles. Par ailleurs, nous avons présenté les deux architectures pré-entraînées retenues pour ce travail, à savoir ConvNeXtBase et EfficientNetV2S, qui ont été adaptées à notre tâche de classification. Les résultats obtenus à partir de ces bases seront présentés dans le chapitre suivant.

Chapitre III

Chapitre 3 : Classification des bruits de l’environnement en utilisant les CNN.

III.1 Introduction

Ce chapitre présente la démarche expérimentale adoptée pour la classification des bruits environnementaux à l'aide des deux modèles pré-entraînés, ConvNeXtBase et EfficientNetV2S, appliqués sur les deux représentations visuelles issues du dataset ESC-50, à savoir les spectrogrammes et les Mel-spectrogrammes. Une approche complémentaire est également explorée, consistant à entraîner un modèle CNN 1D directement sur les données audio brutes, afin d'évaluer l'apport des représentations temps-fréquence par rapport au traitement du signal original. Après avoir défini l'environnement de travail et les outils utilisés, nous mettons en place les conditions nécessaires pour comparer les performances obtenues et identifier la configuration la plus pertinente.

III.2 Environnement de développement

L'environnement de développement utilisé dans ce travail permet l'entraînement et l'évaluation des modèles de manière efficace et accessible.

III.2.1 Google colab

L'approche d'enseignement et d'apprentissage du deep learning basée sur Python via Google Colab constitue un outil particulièrement efficace dans le cadre académique, car elle offre une méthode pratique et facile à utiliser pour exploiter le deep learning et le machine learning, avec un accès gratuit aux GPU. [44]

Google Colaboratory, connu sous le nom de « Colab », est une plateforme en ligne mise à disposition gratuitement par Google, offrant un accès à des ressources matérielles performantes telles que le GPU Nvidia K80 disposant de 12 Go de VRAM cadencé à 0,82 GHz, ainsi qu'une unité de traitement tensoriel (TPU) destinée à accélérer les tâches d'entraînement des modèles et d'apprentissage profond. La plateforme dispose également de 2 cœurs de processeur et de 12 Go de mémoire RAM. [27]

Dans le cadre de ce projet, Google Colab a été utilisé comme environnement principal d'entraînement, en tirant profit de ses ressources GPU gratuites pour accélérer l'apprentissage des modèles. Son intégration avec Google Drive a facilité l'accès au dataset ESC-50 tout au long des expérimentations.

III.3 Langages de programmation

Le langage de programmation constitue un élément fondamental dans la mise en œuvre de tout projet d'apprentissage profond. Dans ce travail, Python a été choisi comme langage principal en raison de sa popularité et de la richesse de ses bibliothèques dédiées au Machine Learning.

III.3.1 Python

Python constitue un langage de programmation moderne doté d'une expressivité et d'une puissance considérables. Bien qu'il partage certaines caractéristiques syntaxiques avec Fortran — l'un des pionniers des langages de programmation de haut niveau — il surpasse largement ce dernier en termes de flexibilité et de fonctionnalités. Python adopte un typage dynamique implicite, dispensant ainsi le programmeur de toute déclaration explicite des variables, et s'appuie sur l'indentation comme mécanisme fondamental de structuration du flux de contrôle. Par ailleurs, contrairement à des langages tels que Java qui imposent une programmation orientée objet obligatoire, Python offre une approche multi-paradigme permettant au développeur de recourir à la définition de classes selon les exigences du contexte, sans y être contraint. [45]

Dans le cadre de ce projet, Python a été utilisé comme langage principal pour l'implémentation des modèles de classification audio. L'utilisation de bibliothèques telles que TensorFlow et Keras a permis de simplifier le développement et l'entraînement des réseaux de neurones appliqués au dataset ESC-50.

III.4 Bibliothèques utilisées

La mise en œuvre des modèles d'apprentissage profond dans ce travail repose sur l'utilisation de bibliothèques spécialisées. Ces outils permettent de simplifier le développement, l'entraînement et l'évaluation des réseaux de neurones convolutifs.

III.4.1 TensorFlow

TensorFlow est un framework open source conçue par l'équipe Google Brain, dédié à l'implémentation de solutions d'apprentissage profond reposant sur les réseaux de neurones artificiels. Il intègre une large gamme de modèles et d'algorithmes couvrant aussi bien le Machine Learning que l'apprentissage profond. Ses interfaces de programmation, basées sur le langage Python, permettent de construire, d'entraîner et d'exécuter des réseaux de neurones

pour des tâches de classification et de reconnaissance au sein de la base de données utilisée. [34]

III.4.2 Keras

Keras est une bibliothèque open source dédiée à l'apprentissage profond, développée en Python. Elle offre une intégration transparente avec des frameworks tels que TensorFlow, Theano et Microsoft CNTK, favorisant ainsi une expérimentation rapide tout en allégeant la complexité de développement. Keras met à disposition plusieurs architectures de réseaux de neurones pré-entraînés, parmi lesquelles deux ont été retenues dans ce travail : ConvNeXtBase, EfficientNetV2S. Ces architectures ont été initialement développées dans le cadre de compétitions de vision par ordinateur pour la classification d'images sur la base ImageNet. Keras les propose aux développeurs pour des tâches de classification d'images, d'extraction de caractéristiques et d'adaptation à différents ensembles de classes. [33]

III.5 Les hyper-paramètres fixés pour l'apprentissage

Un hyper-paramètre d'un modèle est une configuration externe du modèle pour améliorer ses performances. Ces valeurs sont définies après une analyse empirique, c'est à dire en trouvant la meilleure valeur à travers plusieurs essais pour améliorer la précision finale. [36]

Ces hyper-paramètres ne sont pas indépendants les uns des autres ; leurs effets sont interdépendants et varient en fonction des données d'entraînement ainsi que de l'architecture du modèle employé. [37] Dans ce travail, les hyper-paramètres ont été configurés spécifiquement pour le dataset ESC-50.

III.5.1 La taille du lot (Batch size)

Le batch size désigne le nombre d'échantillons traités simultanément par le modèle lors d'une itération d'entraînement avant la mise à jour des poids du réseau. Un batch size de **16** a été utilisé dans ce travail, ce qui représente un bon compromis entre la stabilité de la convergence et l'efficacité du calcul, tout en tenant compte des limitations de mémoire de l'environnement GPU utilisé.

III.5.2 Epoque (Epoch)

L'époque représente un passage complet de l'ensemble des données d'entraînement à travers le modèle, utilisée comme critère d'arrêt de l'apprentissage. Nous l'avons fixée à 50 époques dans ce travail.

III.5.3 Optimiseurs (Optimizers)

L'optimiseur est l'algorithme responsable de la mise à jour des poids du réseau de neurones afin de minimiser la fonction de perte lors de l'entraînement. L'optimiseur **Adam** a été utilisé dans ce travail, en raison de sa rapidité de convergence et de son efficacité sur les datasets de taille moyenne.

III.5.4 Taux d'apprentissage (Learning rate)

Le taux d'apprentissage est un hyper-paramètre qui contrôle la taille du pas effectué lors de la mise à jour des poids du réseau à chaque itération d'entraînement. Un taux trop élevé risque de dépasser le minimum de la fonction de perte, tandis qu'un taux trop faible ralentit la convergence. Dans ce travail, un taux d'apprentissage de **0.0001** a été utilisé, une valeur plus petite que celle par défaut, choisie afin d'assurer une convergence plus fine et plus stable lors de l'affinage (fine-tuning) des modèles pré-entraînés.

III.5.5 Taux d'arrêt (Early Stopping)

L'early stopping est une technique de régularisation qui consiste à interrompre l'entraînement du modèle dès que la performance sur l'ensemble de validation cesse de s'améliorer, afin d'éviter le sur-apprentissage. Dans ce travail, cette technique a été appliquée en surveillant la perte de validation, avec une patience de 10 époques avant l'arrêt définitif de l'entraînement.

III.5.6 Fonction de Perte

La fonction de perte mesure l'écart entre les prédictions du modèle et les valeurs réelles lors de l'entraînement, et son objectif est d'être minimisée afin d'améliorer les performances du modèle. Dans ce travail, la fonction Categorical Crossentropy a été utilisée, étant la plus adaptée aux problèmes de classification multi-classes tel que la classification des sons du dataset ESC-50.

III.5.7 Fonction d'activation (Sortie)

La fonction d'activation de sortie transforme les valeurs produites par la dernière couche du réseau en probabilités interprétables. La fonction **Softmax** a été utilisée dans ce travail, car elle convertit les sorties en une distribution de probabilités sur les 26 classes du dataset ESC-50, permettant ainsi d'identifier la classe la plus probable.

III.5.8 Nombre de Classes

Le nombre de classes représente le nombre total de catégories distinctes que le modèle doit apprendre à reconnaître et à classifier. Dans ce travail, le nombre de classes a été fixé à 26, correspondant aux 26 catégories de sons environnementaux sélectionnées à partir du dataset ESC-50.

III.6 Évaluation des performances

Pour mesurer l'efficacité d'un modèle de réseau de neurones, on peut recourir à plusieurs indicateurs que l'on calcule puis interprète. Parmi les approches utilisées figurent notamment l'estimation par maximum de vraisemblance ainsi que l'algorithme de descente de gradient. Avant de préciser ces indicateurs, il convient d'introduire la matrice de confusion, qui correspond à un tableau mettant en relation les valeurs observées et celles prédites. La classe positive désigne la catégorie ciblée, tandis que la classe négative regroupe toutes les autres. Cette matrice permet de distinguer quatre types de résultats. [38]

Dans le cadre de ce travail, nous avons utilisé plusieurs indicateurs pour évaluer les performances de nos modèles, lesquels sont présentés dans les sous-sections suivantes.

III.6.1 Matrice de confusion (Confusion Matrix)

Dans le domaine de l'apprentissage automatique et plus précisément du problème de la classification statistique, une matrice de confusion, également appelée matrice d'erreur. [40]

Est une disposition spécifique de tableau qui permet de visualiser la performance d'un algorithme, généralement un algorithme d'apprentissage supervisé (dans l'apprentissage non supervisé, on l'appelle habituellement une matrice de correspondance). Chaque ligne de la matrice représente les instances d'une classe prédite, tandis que chaque colonne représente les instances d'une classe réelle (ou inversement). [39]

Elle identifie quatre catégories de résultats :

- La classe réelle est prédite positive : c'est un vrai positif (TP)
- La classe réelle est prédite négative : c'est un vrai négatif (TN)
- La classe réelle est positive mais prédite négative : c'est un faux négatif (FN)
- La classe réelle est négative mais prédite positive : c'est un faux positif (FP)

Voici un exemple de matrice de confusion

	Réel positif	**Réel négatif**
Prédite positive	TP	FP
Prédite négative	FN	TN

Tableau III.1 : Exemple illustratif d'une matrice de confusion

III.6.2 Exactitude (Accuracy)

La métrique d'exactitude mesure à quel point les prédictions de votre modèle sont correctes en calculant la proportion de résultats corrects parmi le nombre total de résultats. [41]

Elle est définie mathématiquement comme suit :

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad \text{.....(III.1)}$$

III.6.3 Precision

La précision ou confiance (comme elle est appelée en fouille de données) désigne la proportion des cas positifs prédits qui sont réellement positifs. C'est sur cela que se concentrent l'apprentissage automatique, la fouille de données et la recherche d'information. [42] La précision se calcule selon la formule suivante :

$$\text{Precision} = \frac{TP}{\text{TP+FP}} \quad \text{.....(III.2)}$$

III.6.4 Rappel (Recall)

Le rappel est de plus considéré comme primordial, car l'objectif est d'identifier tous les cas réellement positifs. [42] Il est défini comme le nombre de vrais positifs divisé par la somme des vrais positifs et des faux négatifs. Le Recall est calculé à l'aide de la formule suivante :

$$\text{Recall} = \frac{TP}{\text{TP+FN}} \quad \text{(III.3)}$$

III.6.5 F1-score

Le score F1 calculé à partir des matrices de confusion a été (et demeure) parmi les métriques les plus populaires adoptées dans les tâches de classification binaire. Cependant, ces mesures statistiques peuvent présenter des résultats dangereusement surestimés et excessivement optimistes, en particulier sur des ensembles de données déséquilibrés. [43]

Le score F1 (*F1-Score*) correspond à la moyenne harmonique de la précision (*precision*) et du rappel (*recall*), offrant ainsi une mesure équilibrée de ces deux indicateurs. Il est particulièrement utile lorsque la distribution des classes est déséquilibrée. [75]

Il représente une mesure de performance d'un modèle de classification. Il combine les métriques de précision et de rappel, et il est défini par :

$$\text{F1\_score} = \frac{TP}{\text{TP}+(\frac{1}{2})(\text{FP+FN})} \quad \text{(III.4)}$$

III.6.6 Perte (Loss)

La fonction de perte mesure l'écart entre les probabilités prédites par le modèle et les vraies étiquettes de classe lors de l'entraînement. Dans le cadre d'une classification multi-classes avec activation Softmax, la fonction utilisée est l'Entropie Croisée Catégorielle (Categorical Cross-Entropy), définie par : [84]

$$L = -1/N \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \cdot \log(\hat{y}_{i,c}) \quad \text{(III.5)}$$

où :

- N : le nombre d'échantillons
- C : le nombre de classes (26 dans notre travail)
- $y_{i,c}$: vaut 1 si l'échantillon i appartient à la classe c, 0 sinon (*encodage one-hot*)
- $\hat{y}_{i,c}$: la probabilité prédite par le modèle pour la classe c de l'échantillon i

Plus la valeur de la perte est faible, meilleure est la performance du modèle.

III.7 Création des labels

La base de données a été divisée en trois sous-ensembles : 70% pour l'entraînement du modèle, 15% pour la validation et 15% pour le test. Chaque partie contient les 26 classes, et chaque classe contient 400 images issues de la conversion et de l'augmentation des fichiers

audio WAV. À partir des sous-dossiers portant les noms des classes, les labels ont été créés pour chaque image puis convertis en matrice binaire (one-hot encoding) afin d'être exploités par les modèles. L'ensemble de ce processus de division et de préparation des données a été réalisé dans l'environnement Google Colaboratory (Google Colab), qui offre un accès gratuit aux ressources de calcul GPU, facilitant ainsi le traitement des données.

III.8 Prétraitement des images

Les 02 modèles pré-entraînés sélectionnés utilisent leur propre prétraitement d'image, que nous pouvons l'utiliser directement grâce à une fonction prédéfinie. Les signaux audio sont représentés sous forme de spectrogrammes et de Mel-spectrogrammes de taille **224×224 pixels**. Le résultat du prétraitement de chaque modèle est illustré dans les figures suivantes :

III.8.1 ConvNeXtBase — Spectrogramme

Le prétraitement du modèle ConvNeXtBase s'attend à ce que les images d'entrée (Spectrogrammes) soient des tenseurs flottants de pixels avec des valeurs comprises entre [0-255]. La normalisation est directement intégrée de manière interne dans l'architecture du modèle.

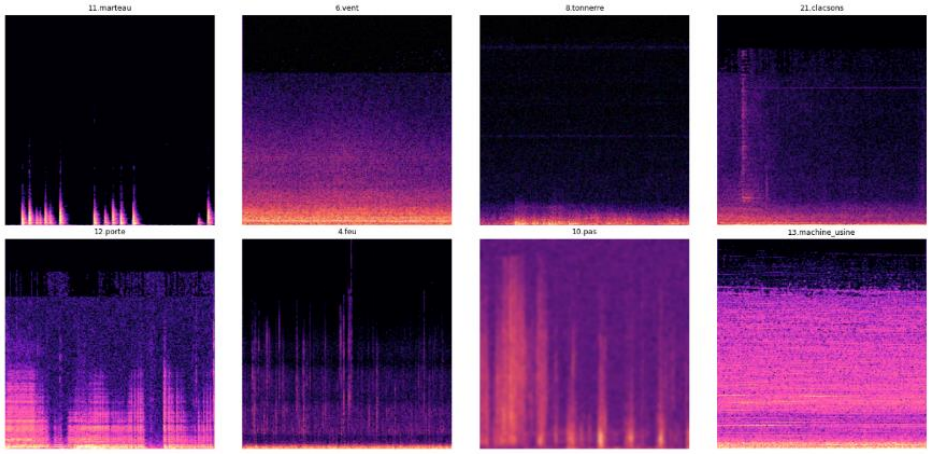

Figure III.1 : Exemple de prétraitement des images avec l'architecture ConvNeXtBase (Spectrogramme).

III.8.2 ConvNeXtBase — Mel-Spectrogramme

Le même prétraitement est appliqué ici. Le modèle ConvNeXtBase s'attend à ce que les entrées (Mel-Spectrogrammes) soient des tenseurs flottants de pixels avec des valeurs comprises entre [0-255].

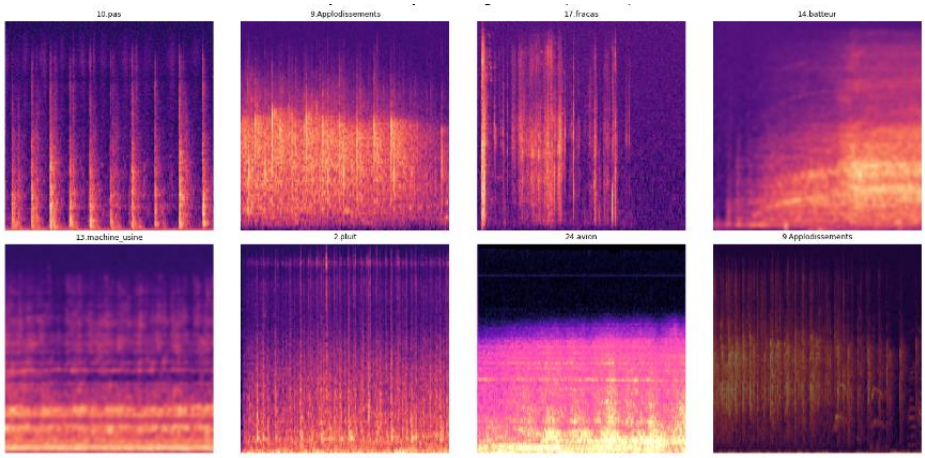

Figure III.2 : Exemple de prétraitement des images avec l'architecture ConvNeXtBase (Mel-spectrogramme).

III.8.3 EfficientNetV2S — Spectrogramme

Les modèles de la famille EfficientNetV2 s'attendent à ce que leurs entrées (Spectrogrammes) soient des tenseurs flottants de pixels avec des valeurs comprises entre [0-255]. La mise à l'échelle est gérée nativement par l'architecture.

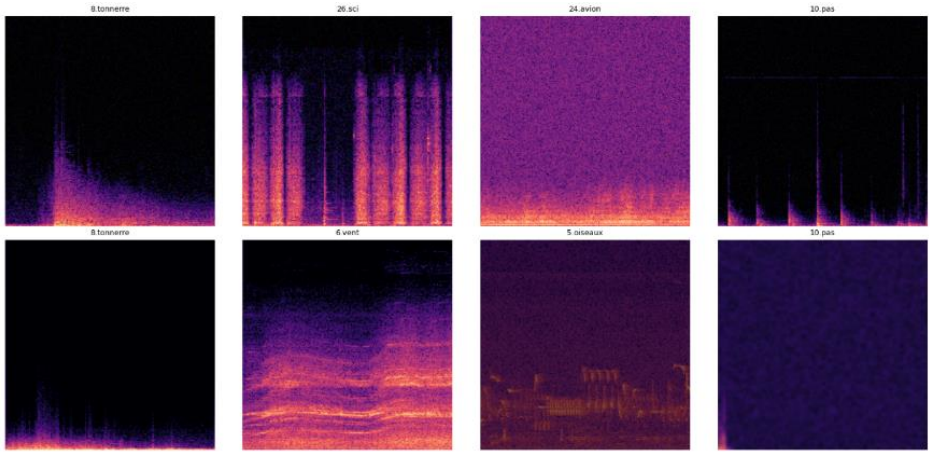

Figure III.3 : Exemple de prétraitement des images avec l'architecture EfficientNetV2S (Spectrogramme).

III.8.4 EfficientNetV2S — Mel-Spectrogramme

Le même prétraitement que pour le spectrogramme standard. Le modèle EfficientNetV2S s'attend à ce que ses entrées (Mel-Spectrogrammes) soient des tenseurs flottants de pixels avec des valeurs comprises entre [0-255].

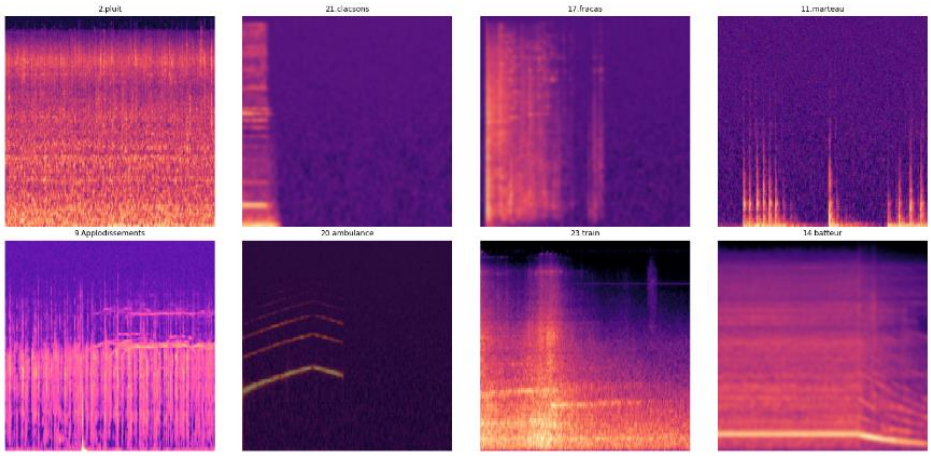

Figure III.4 : Exemple de prétraitement des images avec l'architecture EfficientNetV2S (Mel-spectrogramme).

III.9 Architecture Globale des Modèles

La figure suivante illustre l'architecture globale du modèle proposé. L'entrée est d'abord traitée sous forme d'image à 03 canaux (Spectrogramme ou Mel-Spectrogramme), puis transmise au modèle CNN pré-entraîné (ConvNeXtBase ou EfficientNetV2S). La sortie est ensuite acheminée vers une couche de Global Average Pooling, qui calcule la valeur moyenne de chaque canal afin de réduire le nombre de paramètres du réseau. Cette couche est suivie de couches entièrement connectées (Dense) activées par la fonction ReLU et de couches Dropout pour désactiver certains neurones et éviter le sur-apprentissage. La dernière couche Dense est activée par la fonction SoftMax, contenant 26 neurones correspondant aux 26 classes de sons environnementaux.

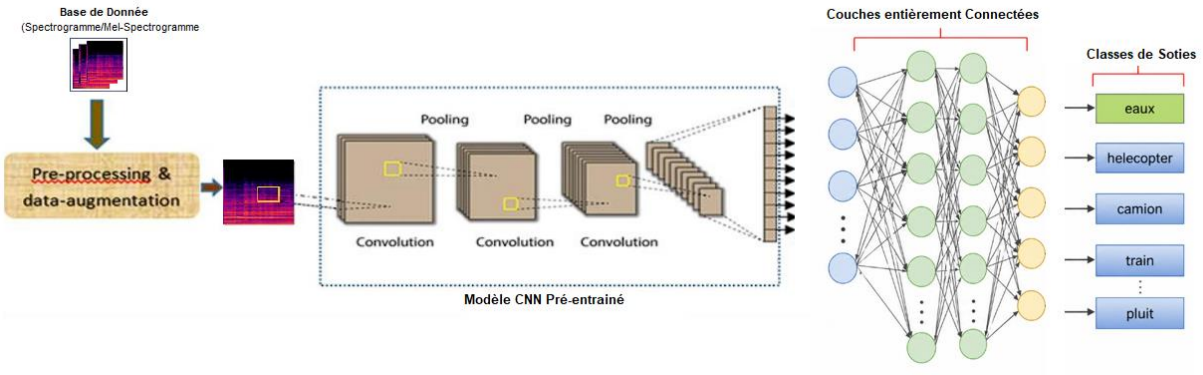

Figure III.5 : Architecture globale du système de classification basé sur les CNN

Les 02 tableaux suivantes présentent l'architecture globale des modèles CNN pré-entraînés utilisés dans cette étude, à savoir ConvNeXtBase et EfficientNetV2S. Chaque architecture repose sur un backbone pré-entraîné sur ImageNet pour l'extraction automatique

des caractéristiques à partir des spectrogrammes et mel-spectrogrammes, suivi d'une tête de classification composée des couches GlobalAveragePooling2D, Dense et Dropout afin de prédire les différentes classes de bruits sonores environnementaux.

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
convnext_base (Functional)	(None, 7, 7, 1024)	87,566,464
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1024)	0
dense (Dense)	(None, 1024)	1,049,600
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 512)	524,800
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 26)	13,338

Tableau III.2 : Architecture du premier modèle utilisant ConvNeXtBase.

- **Total params:** 89,154,202 (340.10 MB)
- **Trainable params:** 1,587,738 (6.06 MB)
- **Non-trainable params:** 87,566,464 (334.04 MB)

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
efficientnetv2-s (Functional)	(None, 7, 7, 1280)	20,331,360
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dense (Dense)	(None, 1024)	1,311,744
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 512)	524,800
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 26)	13,338

Tableau III.3 : Architecture du modèle utilisant EfficientNetV2S.

- **Total params:** 89,154,202 (340.10 MB)
- **Trainable params:** 1,587,738 (6.06 MB)
- **Non-trainable params:** 87,566,464 (334.04 MB)

III.10 Résultats d'entrainement du Modèle

Dans cette partie, nous présentons les courbes d'entraînement obtenues par les deux modèles pré-entraînés (ConvNeXtBase et EfficientNetV2S) appliqués sur les deux types de représentations audio (Spectrogramme et Mel-spectrogramme), illustrant la variation de la Loss, de l'Accuracy et du Recall au fil des époques.

III.10.1 Les Graphes d'Apprentissage et de Validation :

Les graphes d'apprentissage et de validation permettent de suivre l'évolution des performances des modèles au cours des époques d'entraînement. Ils illustrent la variation de la perte (Loss), de la précision (Accuracy) et du rappel (Recall), afin d'analyser le comportement des modèles durant la phase d'entraînement. Les figures suivantes présentent ces courbes pour les quatre expériences réalisées :

III.10.1.a ConvNeXtBase — Spectrogramme :

Les figures III.6, III.7 et III.8 illustrent l'évolution des courbes d'apprentissage et de validation du modèle ConvNeXtBase entraîné sur les spectrogrammes, au fil des 35 époques d'entraînement. Elles représentent respectivement la variation de la perte (Loss), de la précision globale (Accuracy) et du rappel (Recall), où la courbe bleue correspond à l'entraînement et la courbe orange en pointillés correspond à la validation.

Les trois courbes présentent une progression cohérente et stable tout au long des 35 époques d'entraînement, avec des tendances similaires entre les courbes d'entraînement et de validation. La loss de validation se stabilise autour de 0.25, tandis que l'accuracy et le recall de validation atteignent respectivement environ 0.92, contre des valeurs proches de 1.0 en entraînement. Un léger écart entre les deux courbes est observable à partir de l'époque 20, sans toutefois traduire un surapprentissage significatif.

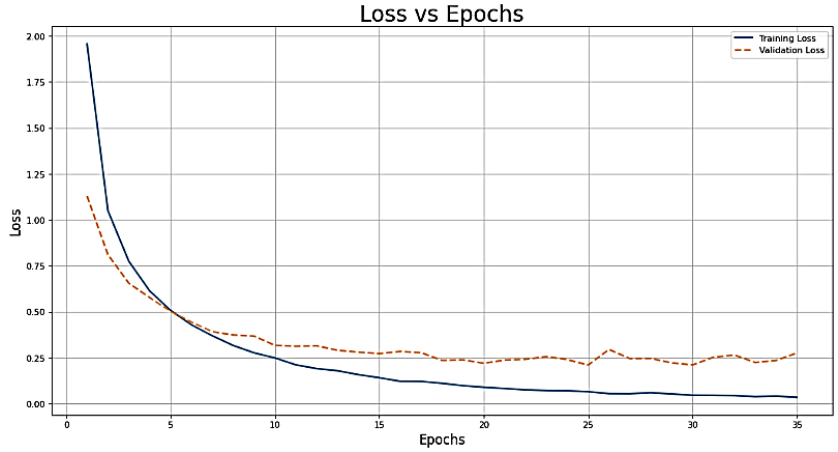

Figure III.6 : Variation de la Loss vs Epochs — ConvNeXtBase (Spectrogramme)

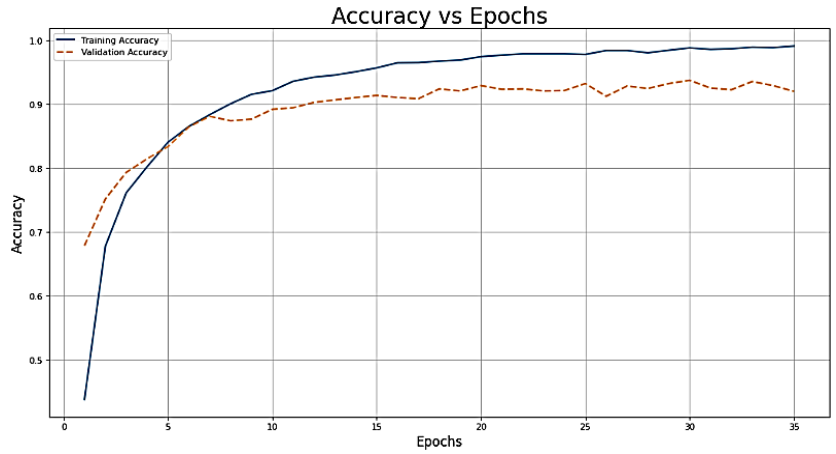

Figure III.7 : Variation de l'Accuracy vs Epochs — ConvNeXtBase (Spectrogramme)

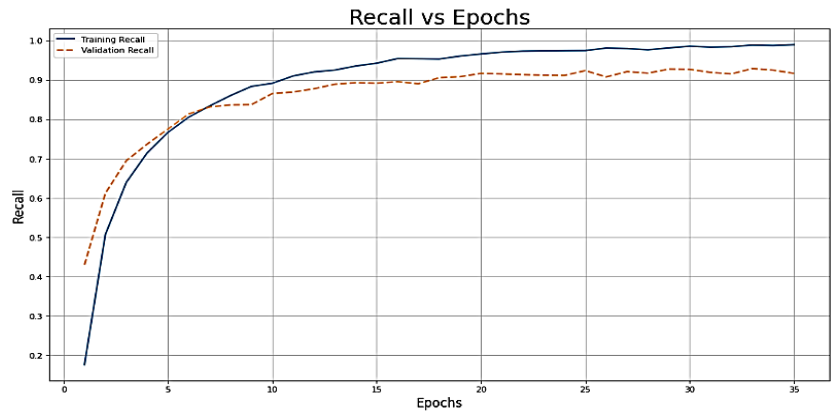

Figure III.8 : Variation du Recall vs Epochs — ConvNeXtBase (Spectrogramme)

Cette configuration confirme la viabilité de ConvNeXtBase pour le traitement des données acoustiques. Le Spectrogramme linéaire fournit una richesse fréquentielle brute suffisante pour assurer un apprentissage stable et une bonne capacité de Généralisation (Generalization) face aux bruits environnementaux.

III.10.1.b ConvNeXtBase — Mel-Spectrogramme :

Les figures III.9, III.10 et III.11 illustrent l'évolution des courbes d'apprentissage et de validation du modèle ConvNeXtBase entraîné sur les Mel-spectrogrammes, au fil des 40 époques d'entraînement. Elles représentent respectivement la variation de la perte (Loss), de la précision globale (Accuracy) et du rappel (Recall).

Les trois courbes présentent une progression cohérente et stable tout au long des 40 époques d'entraînement. La loss de validation se stabilise autour de 0.10, tandis que l'accuracy et le recall de validation atteignent respectivement environ 0.96 et 0.96, contre des valeurs proches de 1.0 en entraînement. Un léger écart entre les deux courbes est observable à partir de l'époque 20, sans toutefois traduire un surapprentissage significatif.

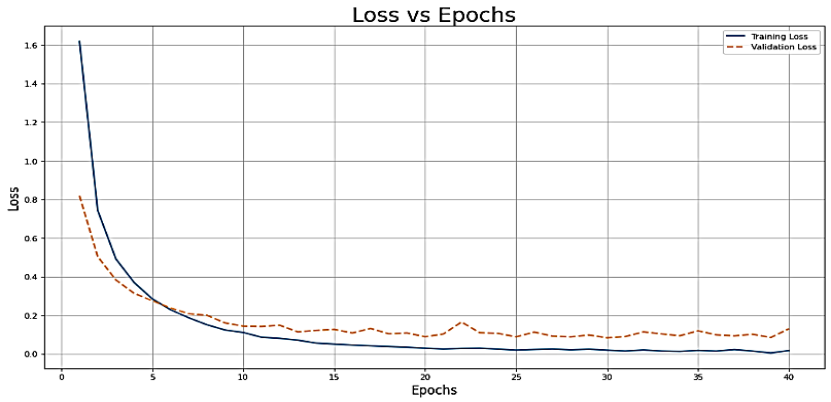

Figure III.9 : Variation de la Loss vs Epochs — ConvNeXtBase (Mel-Spectrogramme)

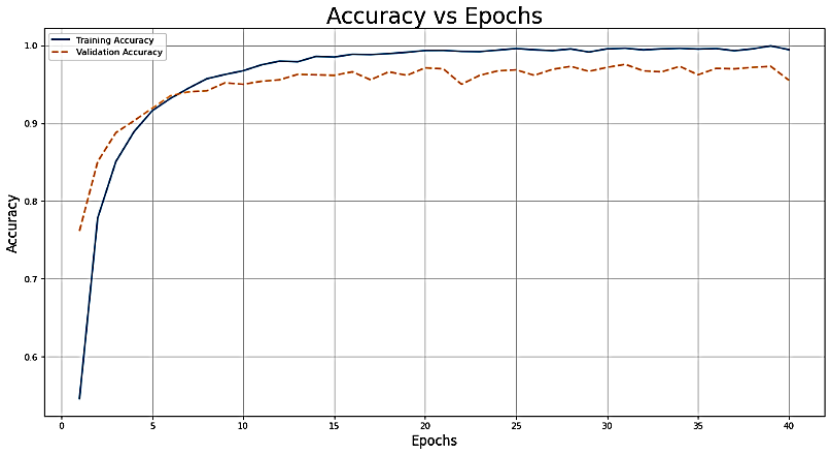

Figure III.10 : Variation de l'Accuracy vs Epochs — ConvNeXtBase (Mel-Spectrogramme)

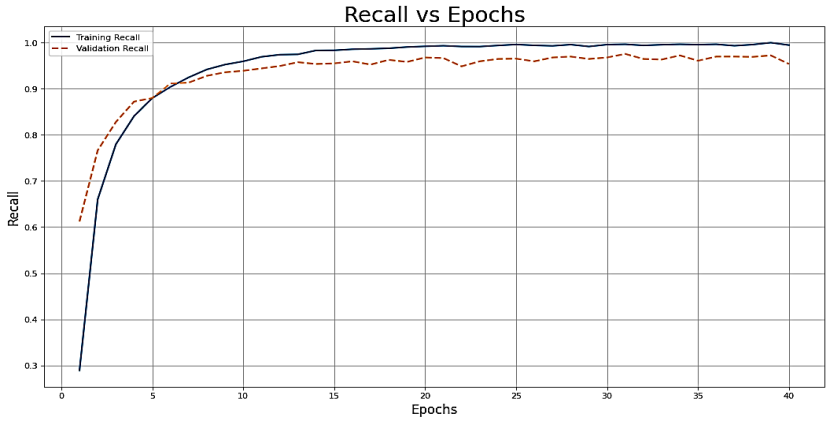

Figure III.11 : Variation du Recall vs Epochs — ConvNeXtBase (Mel-Spectrogramme)

Cette approche évalue le comportement de ConvNeXtBase avec le Mel-Spectrogramme. En adaptant les fréquences à l'audition humaine, ce descripteur réduit le bruit informationnel superflu, favorisant ainsi une convergence efficace de l'apprentissage.

III.10.1.c EfficientNetV2S — Spectrogramme :

Les figures III.12, III.13 et III.14 illustrent l'évolution des courbes d'apprentissage et de validation du modèle EfficientNetV2S entraîné sur les spectrogrammes, au fil des 50 époques d'entraînement. Elles représentent respectivement la variation de la perte (Loss), de la précision globale (Accuracy) et du rappel (Recall).

Les trois courbes présentent une progression cohérente et stable tout au long des 50 époques d'entraînement. La loss de validation se stabilise autour de 0.15, tandis que l'accuracy et le recall de validation atteignent respectivement environ 0.95 et 0.95, contre des valeurs proches de 1.0 en entraînement. Un léger écart entre les deux courbes est observable à partir de l'époque 20, sans toutefois traduire un surapprentissage significatif.

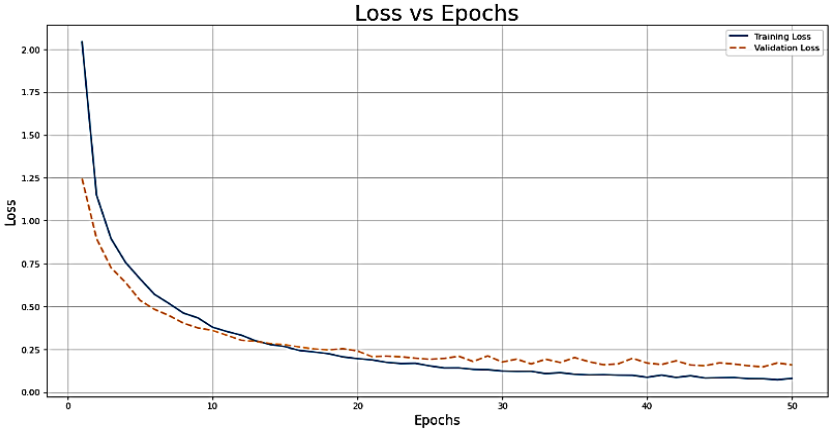

Figure III.12 : Variation de la Loss vs Epochs — EfficientNetV2S (Spectrogramme)

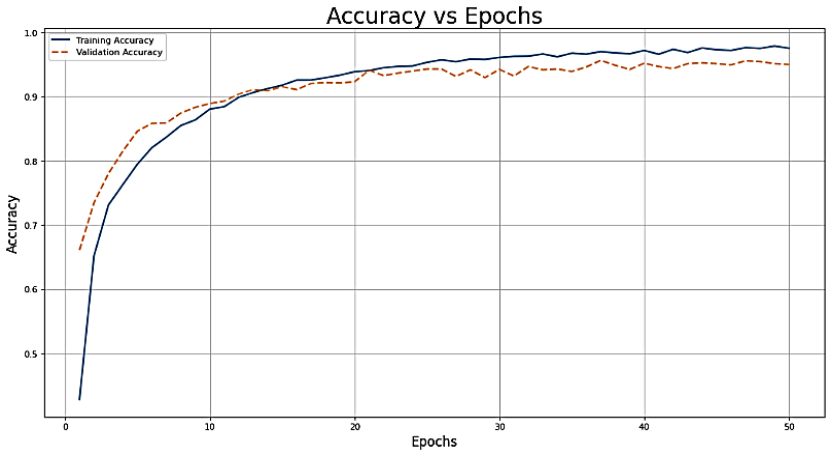

Figure III.13 : Variation de l'Accuracy vs Epochs — EfficientNetV2S (Spectrogramme)

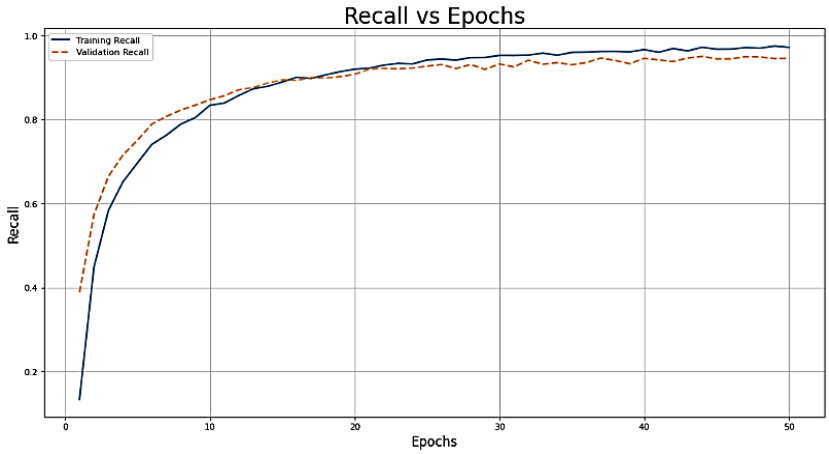

Figure III.14 : Variation du Recall vs Epochs — EfficientNetV2S (Spectrogramme)

Ce cas analyse l'utilisation d'EfficientNetV2S sur des Spectrogrammes linéaires. Sa structure permet de capturer les détails acoustiques essentiels tout en minimisant le risque de Surapprentissage (Overfitting) au fil des époques.

III.10.1.d EfficientNetV2S — Mel-Spectrogramme :

Les figures III.15, III.16 et III.17 illustrent l'évolution des courbes d'apprentissage et de validation du modèle EfficientNetV2S entraîné sur les Mel-spectrogrammes, au fil des 42 époques d'entraînement. Elles représentent respectivement la variation de la perte (Loss), de la précision globale (Accuracy) et du rappel (Recall).

Les trois courbes présentent une progression cohérente et stable tout au long des 42 époques d'entraînement. La loss de validation se stabilise autour de 0.07, tandis que l'accuracy et le recall de validation atteignent respectivement environ 0.97 et 0.97, contre des

valeurs proches de 1.0 en entraînement. Un léger écart entre les deux courbes est observable à partir de l'époque 20, sans toutefois traduire un surapprentissage significatif.

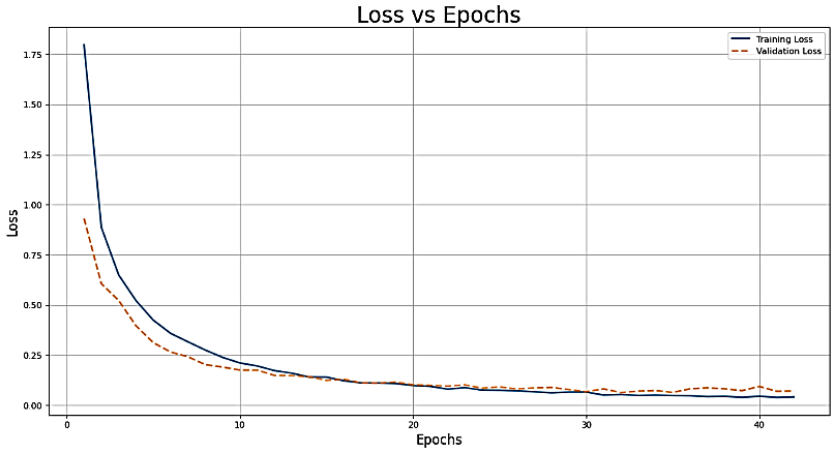

Figure III.15 : Variation de la Loss vs Epochs — EfficientNetV2S (Mel-Spectrogramme)

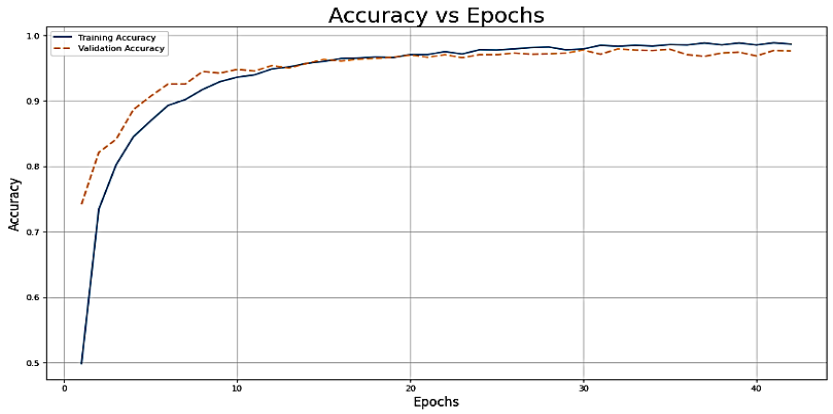

Figure III.16 : Variation de l'Accuracy vs Epochs — EfficientNetV2S (Mel-Spectrogramme)

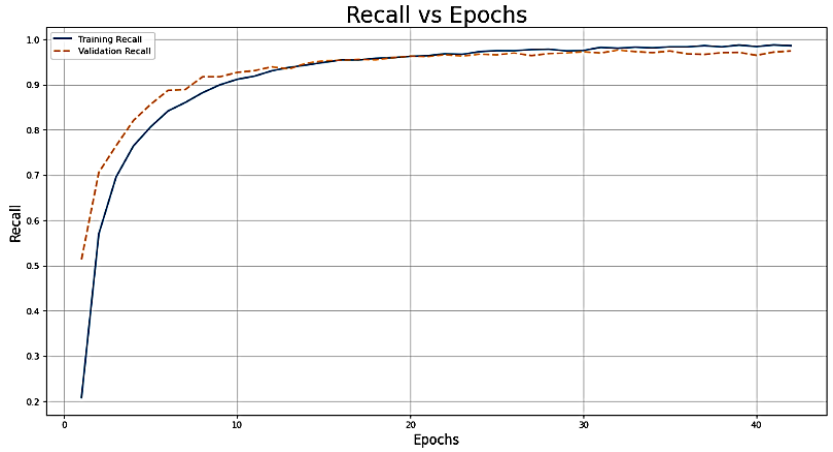

Figure III.17 : Variation du Recall vs Epochs — EfficientNetV2S (Mel-Spectrogramme)

Ce scénario associe l'architecture EfficientNetV2S aux Mel-Spectrogrammes. L'interaction entre les caractéristiques de l'échelle de Mel et la structure du modèle assure la stabilité des courbes et une forte capacité de Généralisation (Generalization).

III.11 Résultats de Classfication du Modèle

Dans cette partie, nous présentons les résultats de classification obtenus par les deux modèles pré-entraînés (ConvNeXtBase et EfficientNetV2S) appliqués sur les deux types de représentations audio (Spectrogramme et Mel-spectrogramme) à des fins de comparaison.

III.11.1 Matrice de confusion

La matrice de confusion constitue un outil d'évaluation permettant de représenter les performances d'un modèle de classification en comparant les classes réelles aux classes prédites sur l'ensemble des données de test. [46]

Les figures suivantes illustrent les matrices de confusion des quatre expériences réalisées, à savoir ConvNeXtBase et EfficientNetV2S entraînés respectivement sur les spectrogrammes et les Mel-spectrogrammes.

III.11.1.a ConvNeXtBase — Spectrogramme :

La figure suivante présente la matrice de confusion obtenue par le modèle ConvNeXtBase appliqué aux spectrogrammes. Elle permet d'évaluer la capacité du modèle à distinguer correctement les différentes classes sonores et à repérer les éventuelles confusions entre elles.

La matrice de confusion montre une performance remarquable avec une diagonale dominante ‹attestant d'une classification quasi parfaite pour la majorité des classes. Les catégories comme « oiseaux », « pluie » et « fracas » sont identifiées sans erreur, tandis que de rares confusions apparaissent entre les bruits d'avion, d'hélicoptère et de vent en raison de leurs similitudes fréquentielles. En somme, ce modèle démontre une robustesse élevée et une grande précision dans l'extraction des caractéristiques à partir des spectrogrammes.

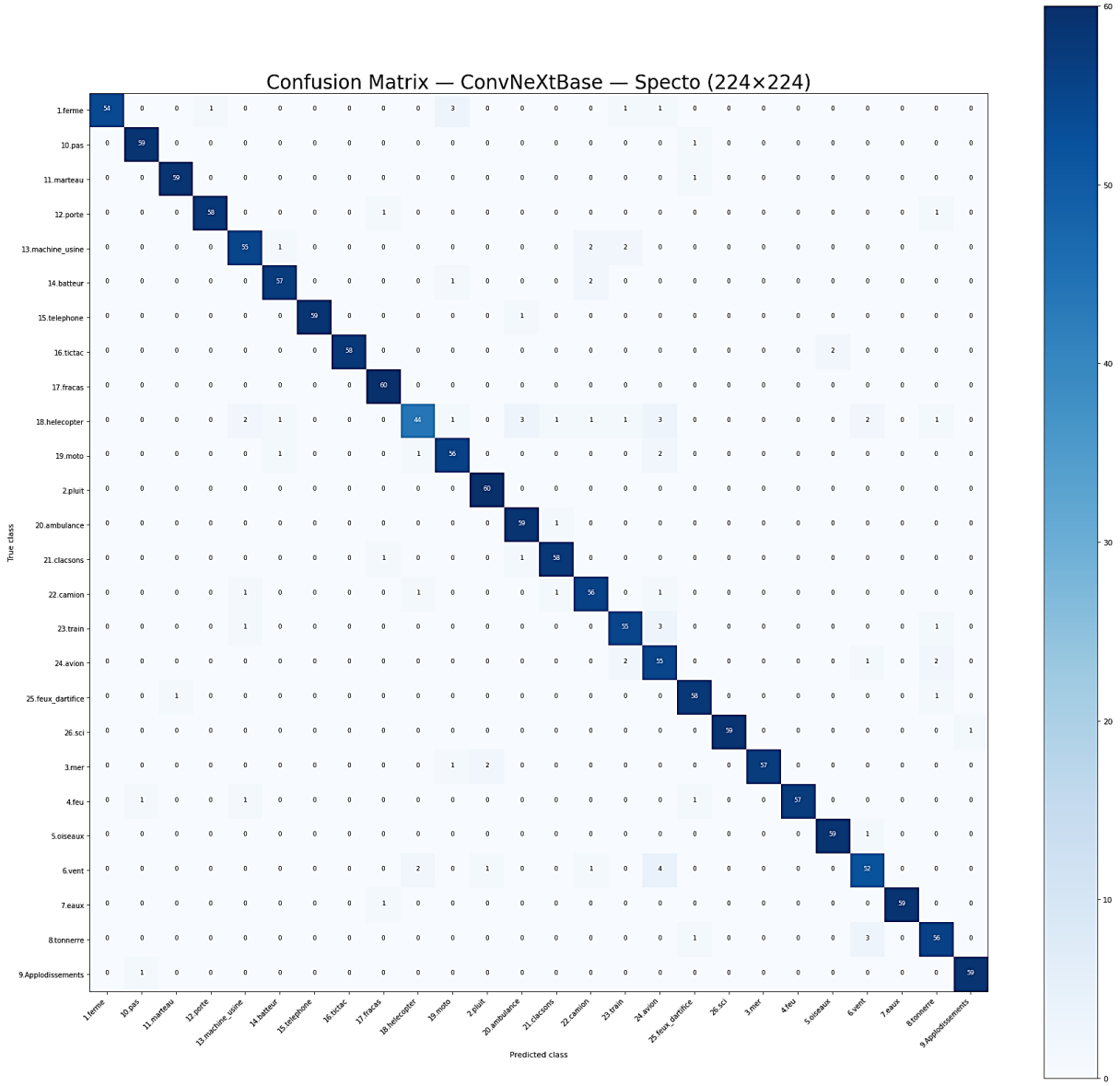

Figure III.18 : Matrice de confusion du modèle ConvNeXtBase (Spectrogramme)

III.11.1.b ConvNeXtBase — Mel-Spectrogramme :

La figure suivante présente la matrice de confusion obtenue par le modèle ConvNeXtBase appliqué aux Mel-spectrogrammes. Elle permet d'évaluer la capacité du modèle à reconnaître correctement les différentes classes sonores et à identifier les éventuelles confusions entre elles.

L'utilisation des Mel-spectrogrammes avec le modèle ConvNeXtBase accentue la précision globale, avec une diagonale quasi parfaite pour des classes telles que « avion », « tictac » et « pluie ». Comparativement aux spectrogrammes classiques, cette approche réduit significativement les confusions, bien que des erreurs mineures subsistent entre « hélicoptère

» et « avion ». En conclusion, l'échelle de Mel semble optimiser l'extraction des traits acoustiques, offrant ainsi une performance légèrement supérieure et plus stable pour la classification des bruits environnementaux.

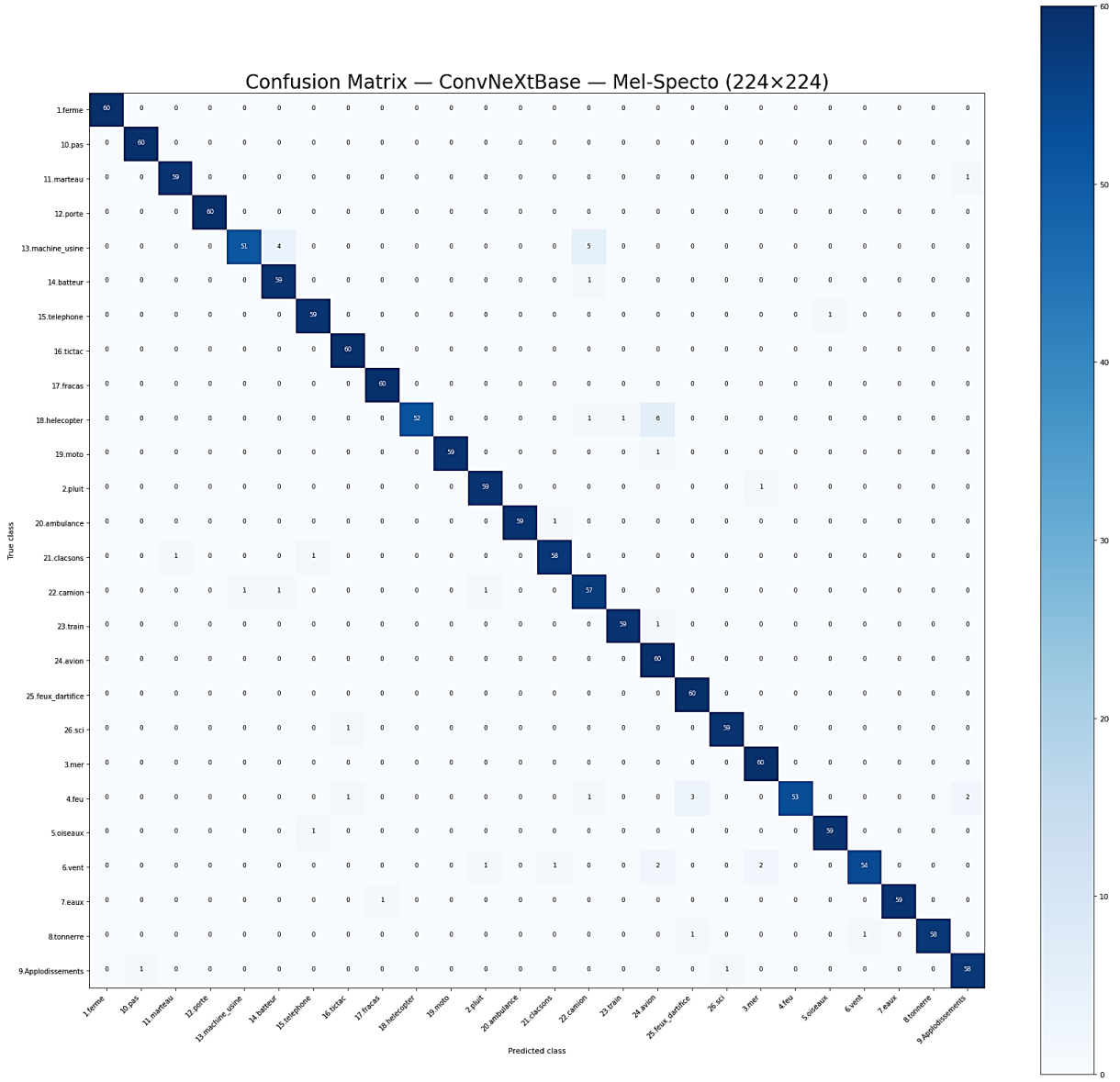

Figure III.19 : Matrice de confusion du modèle ConvNeXtBase (Mel-Spectrogramme)

III.11.1.c EfficientNetV2S — Spectrogramme :

La figure suivante présente la matrice de confusion obtenue par le modèle EfficientNetV2S appliqué aux spectrogrammes. Elle permet d'évaluer la capacité du modèle à classer correctement les différentes classes sonores et à repérer les éventuelles confusions entre elles.

Le passage à l'architecture EfficientNetV2S avec les spectrogrammes montre une précision stable, particulièrement pour les classes « ferme », « marteau » et « téléphone ». Bien que ce modèle soit plus léger que ConvNeXt, il maintient une performance compétitive, malgré une dispersion légèrement plus marquée dans la classification des bruits de moteurs (hélicoptère et avion). En comparaison avec les résultats précédents, EfficientNetV2S offre un excellent compromis entre complexité architecturale et robustesse de classification pour les signaux environnementaux.

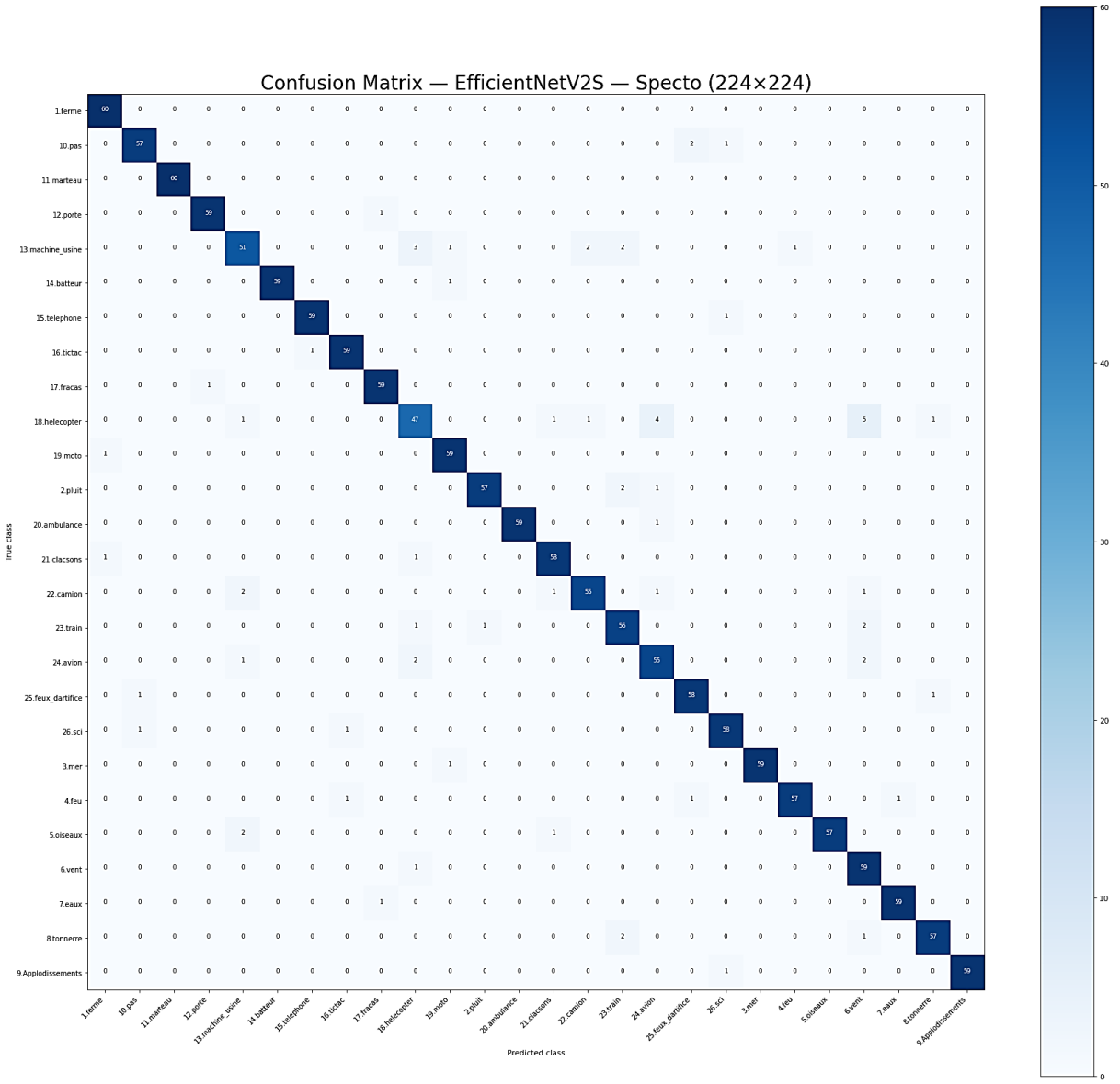

Figure III.20 : Matrice de confusion du modèle EfficientNetV2S (Spectrogramme)

III.11.1.d EfficientNetV2S — Mel-Spectrogramme :

La figure suivante présente la matrice de confusion obtenue par le modèle EfficientNetV2S appliqué aux Mel-spectrogrammes. Elle permet d'évaluer la capacité du modèle à reconnaître correctement les différentes classes sonores et à identifier les éventuelles confusions entre elles.

L'intégration des Mel-spectrogrammes avec l'architecture EfficientNetV2S aboutit aux résultats les plus performants, avec une quasi-absence d'erreurs sur la majorité des classes acoustiques. La matrice de confusion montre une concentration maximale sur la diagonale principale, illustrant une distinction parfaite pour des catégories complexes comme « moto », « ambulance » et « tonnerre ». En conclusion, cette configuration représente le meilleur compromis de l'étude, surpassant les spectrogrammes classiques en termes de précision tout en confirmant que l'échelle de Mel est mieux adaptée à la perception et à la classification des sons environnementaux.

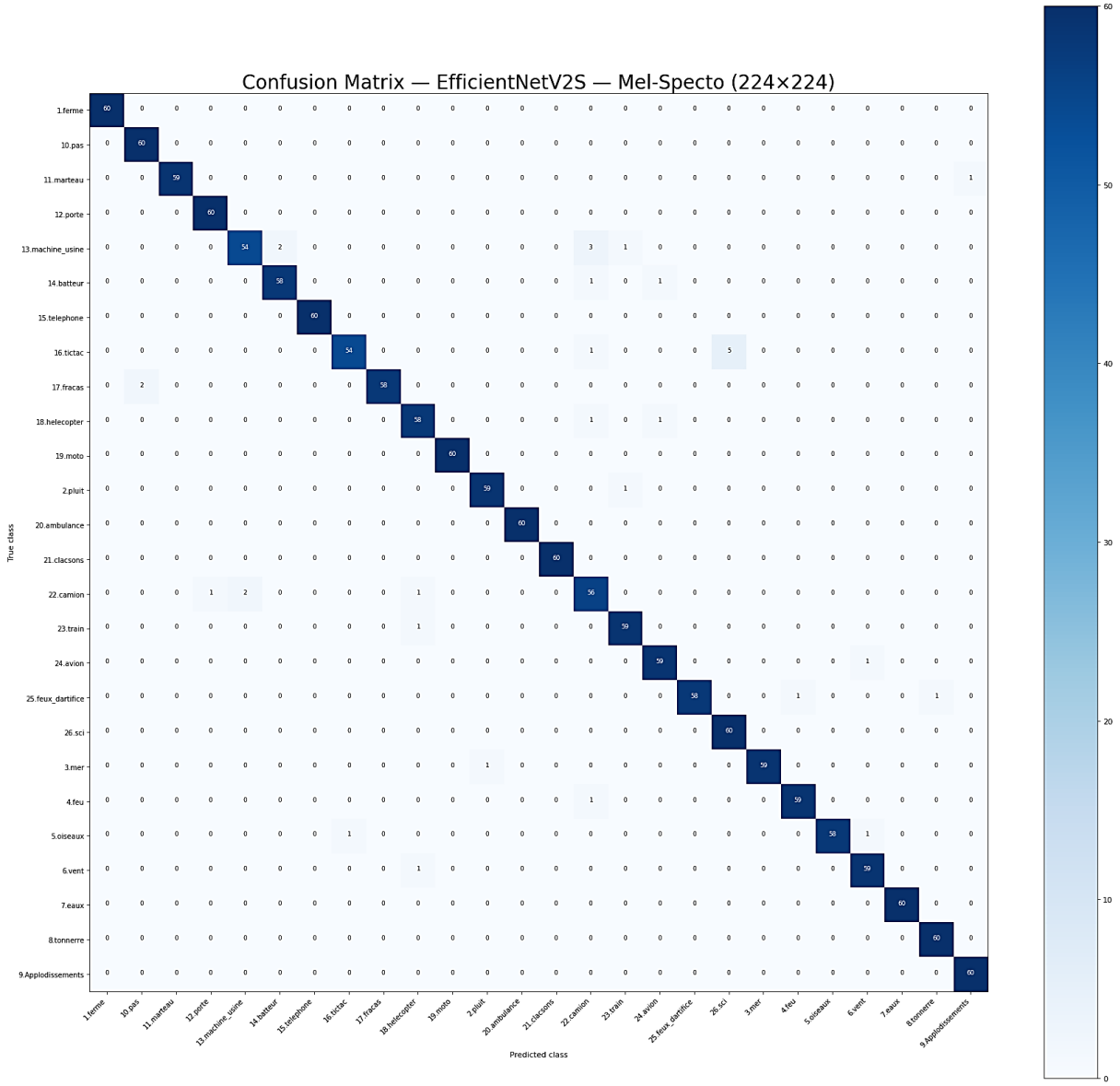

Figure III.21 : Matrice de confusion du modèle EfficientNetV2S (Mel-Spectrogramme)

III.12 Résultats de classification

Les figures suivantes représentent les résultats de classification des deux modèles utilisés : ConvNeXtBase et EfficientNetV2S, entraînés respectivement sur les spectrogrammes et les Mel-spectrogrammes.

III.12.1 ConvNeXtBase — Spectrogramme

La figure suivante présente le rapport de classification détaillé obtenu par le modèle ConvNeXtBase appliqué aux spectrogrammes. Elle permet d'évaluer les performances du modèle pour chaque classe sonore à travers les mesures de précision, de rappel, de F1-score et de support.

Le modèle affiche une performance globale solide avec une accuracy de 95 %, illustrant une excellente capacité de généralisation sur les spectrogrammes. Les catégories « telephone », « scie » et « eaux » atteignent des scores F1-score quasi parfaits (0,99), tandis que la classe « helicoptere » présente le score le plus faible (0,81). En somme, cette configuration confirme l'efficacité de l'architecture ConvNeXtBase pour la discrimination précise des bruits environnementaux, malgré des difficultés mineures sur certains signaux aéronautiques.

	precision	recall	f1-score	support
1.ferme	1.00	0.90	0.95	60
10.pas	0.97	0.98	0.98	60
11.marteau	0.98	0.98	0.98	60
12.porte	0.98	0.97	0.97	60
13.machine_usine	0.92	0.92	0.92	60
14.batteur	0.95	0.95	0.95	60
15.telephone	1.00	0.98	0.99	60
16.tictac	1.00	0.97	0.98	60
17.fracas	0.95	1.00	0.98	60
18.helecopter	0.92	0.73	0.81	60
19.moto	0.90	0.93	0.92	60
2.pluit	0.95	1.00	0.98	60
20.ambulance	0.92	0.98	0.95	60
21.clacsons	0.95	0.97	0.96	60
22.camion	0.90	0.93	0.92	60
23.train	0.90	0.92	0.91	60
24.avion	0.80	0.92	0.85	60
25.feux_dartifice	0.94	0.97	0.95	60
26.sci	1.00	0.98	0.99	60
3.mer	1.00	0.95	0.97	60
4.feu	1.00	0.95	0.97	60
5.oiseaux	0.97	0.98	0.98	60
6.vent	0.88	0.87	0.87	60
7.eaux	1.00	0.98	0.99	60
8.tonnerre	0.90	0.93	0.92	60
9.Applodissements	0.98	0.98	0.98	60
accuracy			0.95	1560
macro avg	0.95	0.95	0.95	1560
weighted avg	0.95	0.95	0.95	1560

Figure III.22 : Rapport de classification détaillé du modèle ConvNeXtBase — Spectrogramme

III.12.2 ConvNeXtBase — Mel-Spectrogramme

La figure suivante présente le rapport de classification détaillé obtenu par le modèle ConvNeXtBase appliqué aux Mel-spectrogrammes. Elle permet d'évaluer les performances du modèle pour chaque classe sonore à travers les mesures de précision, de rappel, de F1-score et de support.

L'adoption de l'échelle de Mel avec ConvNeXtBase porte l'accuracy globale à 97 %, surpassant ainsi les résultats obtenus avec les spectrogrammes classiques. Des scores parfaits (F1-score de 1,00) sont enregistrés pour les classes « ferme » et « porte », tandis que les catégories « pas », « fracas » et « eaux » atteignent 0,99. Les classes « machine_usine » et « camion » marquent les scores les plus bas (0,91), bien que la performance reste très équilibrée sur l'ensemble des 26 classes. En conclusion, le Mel-spectrogramme optimise nettement la séparation des classes, confirmant la supériorité de ce prétraitement pour la classification acoustique environnementale.

	precision	recall	f1-score	support
1.ferme	1.00	1.00	1.00	60
10.pas	0.98	1.00	0.99	60
11.marteau	0.98	0.98	0.98	60
12.porte	1.00	1.00	1.00	60
13.machine_usine	0.98	0.85	0.91	60
14.batteur	0.92	0.98	0.95	60
15.telephone	0.97	0.98	0.98	60
16.tictac	0.97	1.00	0.98	60
17.fracas	0.98	1.00	0.99	60
18.helecopter	1.00	0.87	0.93	60
19.moto	1.00	0.98	0.99	60
2.pluit	0.97	0.98	0.98	60
20.ambulance	1.00	0.98	0.99	60
21.clacsons	0.97	0.97	0.97	60
22.camion	0.88	0.95	0.91	60
23.train	0.98	0.98	0.98	60
24.avion	0.86	1.00	0.92	60
25.feux_dartifice	0.94	1.00	0.97	60
26.sci	0.98	0.98	0.98	60
3.mer	0.95	1.00	0.98	60
4.feu	1.00	0.88	0.94	60
5.oiseaux	0.98	0.98	0.98	60
6.vent	0.98	0.90	0.94	60
7.eaux	1.00	0.98	0.99	60
8.tonnerre	1.00	0.97	0.98	60
9.Applodissements	0.95	0.97	0.96	60
accuracy			0.97	1560
macro avg	0.97	0.97	0.97	1560
weighted avg	0.97	0.97	0.97	1560

Figure III.23 : Rapport de classification détaillé du modèle ConvNeXtBase — Mel-Spectrogramme

III.12.3 EfficientNetV2S — Spectrogramme

La figure suivante présente le rapport de classification détaillé obtenu par le modèle EfficientNetV2S appliqué aux spectrogrammes. Elle permet d'évaluer, pour chaque classe sonore, les valeurs de précision, de rappel, de F1-score et de support.

L'architecture EfficientNetV2S appliquée aux spectrogrammes atteint une précision globale (accuracy) de 96 %. Le modèle démontre une efficacité remarquable pour la classe « marteau » avec un F1-score parfait de 1,00, suivie de près par les classes « batteur », « ambulance » et « mer » à 0,99. En revanche, des performances plus modérées sont notées pour la classe « helicoptere » (0,82) et « machine_usine » (0,87). En conclusion, ce modèle offre un haut niveau de fiabilité, prouvant que cette structure légère est capable de capturer des détails acoustiques complexes tout en maintenant une excellente stabilité de classification.

	precision	recall	f1-score	support
1.ferme	0.97	1.00	0.98	60
10.pas	0.97	0.95	0.96	60
11.marteau	1.00	1.00	1.00	60
12.porte	0.98	0.98	0.98	60
13.machine_usine	0.89	0.85	0.87	60
14.batteur	1.00	0.98	0.99	60
15.telephone	0.98	0.98	0.98	60
16.tictac	0.97	0.98	0.98	60
17.fracas	0.97	0.98	0.98	60
18.helecopter	0.85	0.78	0.82	60
19.moto	0.95	0.98	0.97	60
2.pluit	0.98	0.95	0.97	60
20.ambulance	1.00	0.98	0.99	60
21.clacsons	0.95	0.97	0.96	60
22.camion	0.95	0.92	0.93	60
23.train	0.90	0.93	0.92	60
24.avion	0.89	0.92	0.90	60
25.feux_dartifice	0.95	0.97	0.96	60
26.sci	0.95	0.97	0.96	60
3.mer	1.00	0.98	0.99	60
4.feu	0.98	0.95	0.97	60
5.oiseaux	1.00	0.95	0.97	60
6.vent	0.84	0.98	0.91	60
7.eaux	0.98	0.98	0.98	60
8.tonnerre	0.97	0.95	0.96	60
9.Applodissements	1.00	0.98	0.99	60
accuracy			0.96	1560
macro avg	0.96	0.96	0.96	1560
weighted avg	0.96	0.96	0.96	1560

Figure III.24 : Rapport de classification détaillé du modèle EfficientNetV2S — Spectrogramme

III.12.4 EfficientNetV2S — Mel-Spectrogramme

La figure suivante présente le rapport de classification détaillé obtenu par le modèle EfficientNetV2S appliqué aux Mel-spectrogrammes. Elle permet d'évaluer, pour chaque classe sonore, les valeurs de précision, de rappel, de F1-score et de support.

Cette configuration atteint le niveau de performance le plus élevé avec une accuracy globale de 98 %. Le modèle affiche une précision parfaite (F1-score de 1,00) pour de nombreuses classes, notamment « ferme », « telephone », « moto », « ambulance », « clacsons » et « eaux ». La classe « camion » enregistre le score le plus bas de la série avec un F1-score de 0,91, ce qui reste une performance très satisfaisante. En conclusion, l'association de l'architecture EfficientNetV2S et des Mel-spectrogrammes constitue l'approche la plus efficace de cette étude, offrant une discrimination quasi optimale des bruits environnementaux.

	precision	recall	f1-score	support
1.ferme	1.00	1.00	1.00	60
10.pas	0.97	1.00	0.98	60
11.marteau	1.00	0.98	0.99	60
12.porte	0.98	1.00	0.99	60
13.machine_usine	0.96	0.90	0.93	60
14.batteur	0.97	0.97	0.97	60
15.telephone	1.00	1.00	1.00	60
16.tictac	0.98	0.90	0.94	60
17.fracas	1.00	0.97	0.98	60
18.helecopter	0.95	0.97	0.96	60
19.moto	1.00	1.00	1.00	60
2.pluit	0.98	0.98	0.98	60
20.ambulance	1.00	1.00	1.00	60
21.clacsons	1.00	1.00	1.00	60
22.camion	0.89	0.93	0.91	60
23.train	0.97	0.98	0.98	60
24.avion	0.97	0.98	0.98	60
25.feux_dartifice	1.00	0.97	0.98	60
26.sci	0.92	1.00	0.96	60
3.mer	1.00	0.98	0.99	60
4.feu	0.98	0.98	0.98	60
5.oiseaux	1.00	0.97	0.98	60
6.vent	0.97	0.98	0.98	60
7.eaux	1.00	1.00	1.00	60
8.tonnerre	0.98	1.00	0.99	60
9.Applodissements	0.98	1.00	0.99	60
accuracy			0.98	1560
macro avg	0.98	0.98	0.98	1560
weighted avg	0.98	0.98	0.98	1560

Figure III.25 : Rapport de classification détaillé du modèle EfficientNetV2S — Mel-Spectrogramme

III.13 Analyse Comparative des Résultats

Le tableau ci-dessous présente une synthèse comparative des résultats obtenus par les deux modèles pré-entraînés (ConvNeXtBase et EfficientNetV2S) appliqués sur les deux types de représentations audio (Spectrogramme et Mel-spectrogramme), en termes d'Accuracy, de Precision, de Recall et de Loss.

Ces résultats montrent que la combinaison la plus performante est obtenue avec EfficientNetV2S et le Mel-spectrogramme, qui offrent les meilleures valeurs d'accuracy, de precision, de recall et la loss la plus faible.

*(Note : Les métriques de performance ont été converties en pourcentages pour une meilleure lisibilité, tandis que la **Loss** conserve sa valeur décimale d'origine).*

Modèle	**Représentation**	**Accuracy**	**Precision**	**Recall**	**Loss**
ConvNeXtBase	Spectrogramme	94.74%	95.38%	94.04%	0.1836
ConvNeXtBase	Mel-Spectrogramme	96.86%	97.04%	96.60%	0.0980
EfficientNetV2S	Spectrogramme	95.64%	96.75%	95.38%	0.1278
EfficientNetV2S	Mel-Spectrogramme	97.88%	98.01%	97.76%	0.0615

Tableau III.4 : Synthèse comparative des performances globales des modèles évalués

La figure suivante présente un histogramme comparatif des performances du modèle ConvNeXtBase selon les deux représentations audio utilisées, à savoir le spectrogramme et le Mel-spectrogramme, en termes d'accuracy, de precision, de recall et de loss.

Le graphique confirme la supériorité du Mel-Spectrogramme pour ConvNeXtBase. Il accroît l'Exactitude (Accuracy), la Précision (Precision) et le Rappel (Recall) d'environ 0.95 à 0.97, et réduit la Perte (Loss) de moitié (de 0.18 à 0.10). On en déduit que cette représentation capture mieux les caractéristiques acoustiques, assurant une capacité de Généralisation (Generalization) optimale.

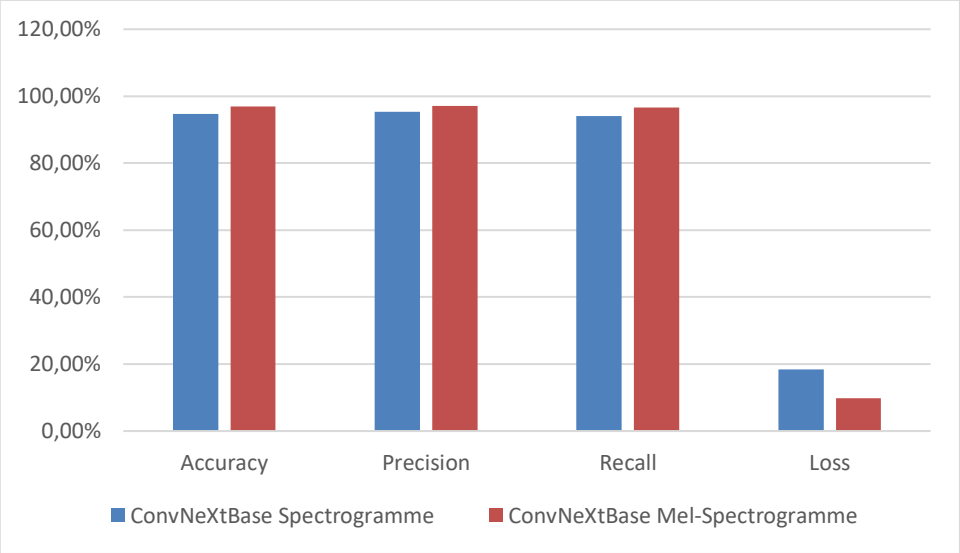

Figure III.26 : Histogramme comparatif des métriques d'évaluation pour le modèle ConvNeXtBase

Le graphique ci-dessous compare les performances du modèle EfficientNetV2S selon les deux représentations audio utilisées, à savoir le spectrogramme et le Mel-spectrogramme, en termes d'accuracy, de precision, de recall et de loss.

Le graphique illustre la supériorité du Mel-Spectrogramme pour l'architecture EfficientNetV2S. L'Exactitude (Accuracy), la Précision (Precision) et le Rappel (Recall) progressent d'environ 0.96 à près de 0.98, tandis que la Perte (Loss) est presque divisée par deux (de 0.13 à 0.07). On en déduit que cette représentation maximise l'extraction des caractéristiques acoustiques, offrant la meilleure capacité de Généralisation (Generalization) globale de l'étude.

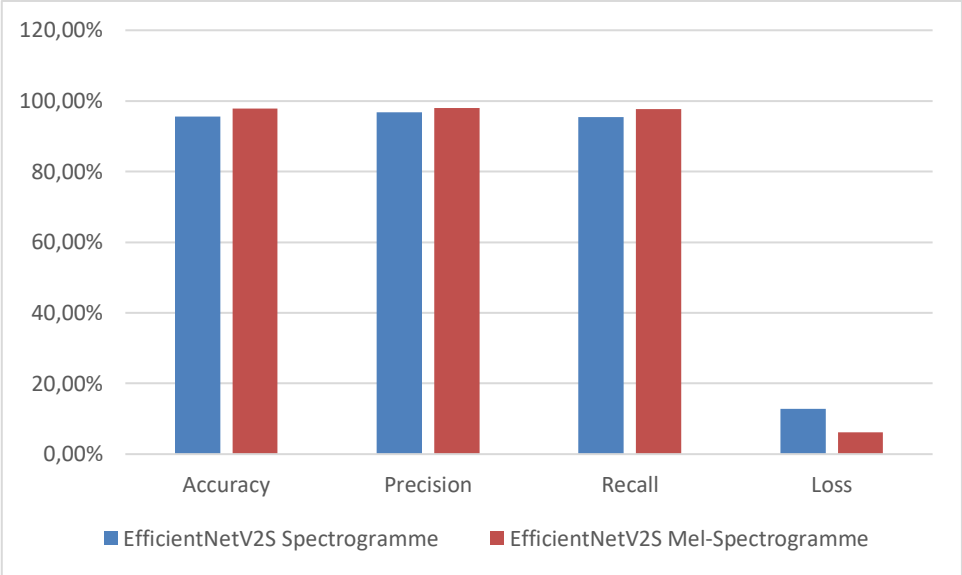

Figure III.27 : Histogramme comparatif des métriques d'évaluation pour le modèle EfficientNetV2S

En comparant les deux modèles optimisés, EfficientNetV2S s'avère nettement supérieur à ConvNeXtBase. Il offre la meilleure capacité de Généralisation (Generalization), maximisant la Précision (Accuracy) et le Rappel (Recall) à près de 0.98, tout en enregistrant la Perte (Loss) la plus faible (0.07). Il s'impose ainsi comme l'architecture la plus performante et la plus robuste de cette étude.

III.14 Étude comparative : Classification par représentations temps-fréquence et signal audio brut

Pour la première approche, nous avons utilisé les modèles pré-entraînés ConvNeXtBase et EfficientNetV2S sur les représentations temps-fréquence afin d'évaluer leur efficacité dans la classification des sons. En parallèle, une autre approche a été testée en utilisant un 1D CNN entraîné directement sur les fichiers audio bruts. Cette méthode permet d'évaluer la capacité du modèle à apprendre à partir du signal original, sans passer par une transformation en image ou en représentation spectrale.

Nous présenterons ensuite les résultats obtenus pour l'audio brut afin d'évaluer ses performances respectives. Puis, nous comparerons les résultats du traitement des sons bruts avec ceux obtenus à partir des représentations temps-fréquence, (Spectrogramme et Mel-Spectrogramme), afin de mettre en évidence la méthode la plus efficace pour cette tâche de classification.

III.14.1 Résultats obtenus sur les données audio brutes

Le tableau ci-dessous présente les résultats obtenus sur les données audio brutes à l'aide des deux modèles pré-entraînés, ConvNeXtBase et EfficientNetV2S, en termes de test loss, test accuracy, test recall et precision.

On constate qu'EfficientNetV2S surpasse légèrement ConvNeXtBase, avec une accuracy de 69.23% contre 67.95% et un recall de 56.41% contre 51.60%.

Modèle	**Test Loss**	**Test Accuracy**	**Test Recall**	**Precision**
ConvNeXtBase	1.0307	67.95%	51.60%	54.60%
EfficientNetV2S	1.0290	**69.23%**	**56.41%**	**59.41%**

Tableau III.5 : Résultats obtenus avec ConvNeXtBase et EfficientNetV2S sur les données audio brutes

La figure suivante illustre l'évolution des performances du modèle ConvNeXtBase sur les données audio brutes au cours des époques d'entraînement, en montrant la perte d'entraînement (training loss) et de validation (validation loss), la précision d'entraînement (training accuracy) et de validation (validation accuracy), ainsi que le rappel d'entraînement (training recall) et de validation (validation recall).

Les courbes montrent une amélioration progressive des performances du modèle ConvNeXtBase sur les données audio brutes, avec une baisse de la loss et une hausse de l'accuracy et du recall au fil des époques.

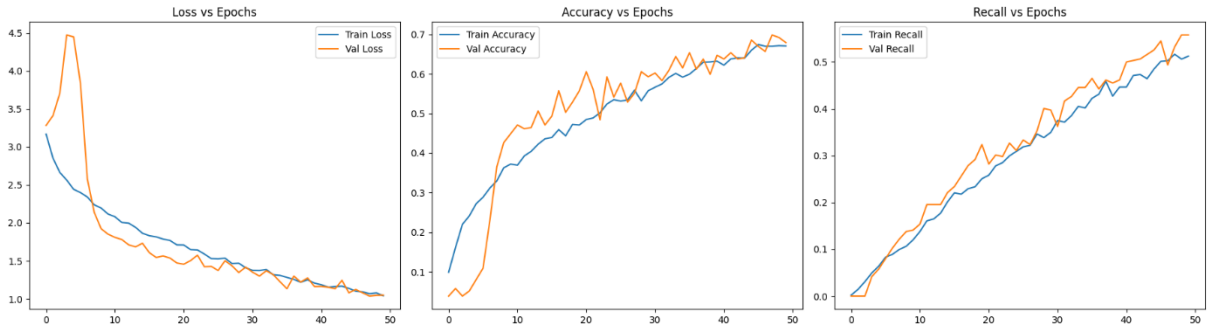

Figure III.28 : Évolution des performances du modèle ConvNeXtBase sur les données audio brutes

Le figure suivante illustre l'évolution des performances du modèle EfficientNetV2S sur les données audio brutes au cours des époques d'entraînement, en montrant la perte d'entraînement (training loss) et de validation (validation loss), la précision d'entraînement

(training accuracy) et de validation (validation accuracy), ainsi que le rappel d'entraînement (training recall) et de validation (validation recall).

Les courbes montrent une amélioration progressive des performances du modèle EfficientNetV2S sur les données audio brutes, avec une baisse de la loss et une hausse de l'accuracy et du recall au fil des époques.

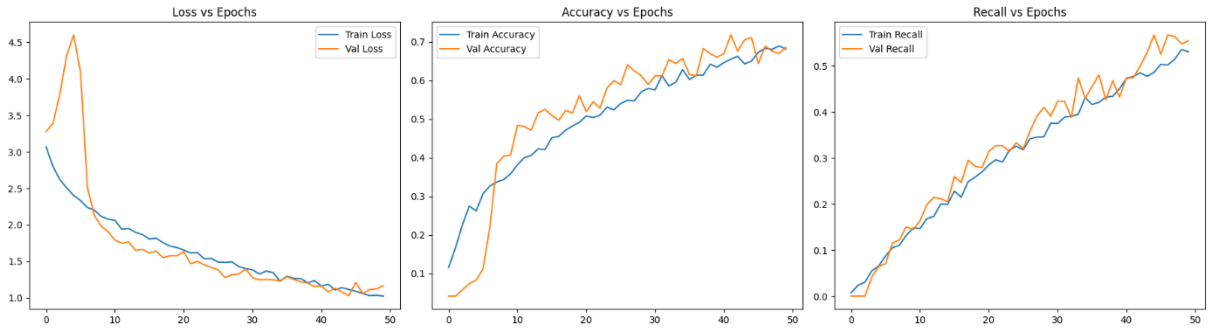

Figure III.29 : Évolution des performances du modèle EfficientNetV2S sur les données audio brutes

Ces résultats démontrent que le traitement direct des signaux acoustiques bruts limite la capacité des architectures CNN à extraire des descripteurs profonds. Cela justifie pleinement la nécessité de passer par des représentations temps-fréquence bidimensionnelles pour optimiser l'apprentissage et la Généralisation (Generalization) des modèles.

III.14.2 Comparaison des résultats obtenus sur les données audio brutes

Le tableau suivant présente une synthèse comparative des performances obtenues avec les modèles entraînés sur les représentations temps-fréquence et sur les données audio brutes. Cette comparaison permet de mettre en évidence l'écart de performance entre les deux types d'approches.

Modèle	Représentation	Accuracy	Precision	Recall	Loss
ConvNeXtBase	Spectrogramme	94.74%	95.38%	94.04%	0.1836
ConvNeXtBase	Mel-Spectrogramme	96.86%	97.04%	96.60%	0.0980
EfficientNetV2S	Spectrogramme	95.64%	96.75%	95.38%	0.1278
EfficientNetV2S	Mel-Spectrogramme	97.88%	98.01%	97.76%	0.0615
ConvNeXtBase	Audio brut	67.95%	54.60%	51.60%	1.0307
EfficientNetV2S	Audio brut	69.23%	59.41%	56.41%	1.0290

Tableau III.6 : Comparaison des performances globales

L'analyse du tableau met en évidence une divergence de performance nette entre les deux approches, soulignant la supériorité des représentations temps-fréquence. Ce constat indique que cette modélisation bidimensionnelle permet au réseau d'exploiter optimalement les indices acoustiques latents, facilitant ainsi la discrimination des classes sonores. À l'inverse, l'exploitation du signal brut génère des performances plus limitées, ce qui démontre que les caractéristiques acoustiques fondamentales du son y sont moins accessibles pour l'architecture. Par conséquent, la transformation spectrale du signal s'impose comme une étape cruciale pour optimiser le processus de classification.

Ce diagramme à barres compare les métriques d'évaluation des modèles ConvNeXtBase et EfficientNetV2S selon la représentation du signal, en mettant en évidence l'accuracy (accuracy), la précision (precision), le rappel (recall) et la perte (loss) pour les deux types d'entrées, spectrogramme et audio brut.

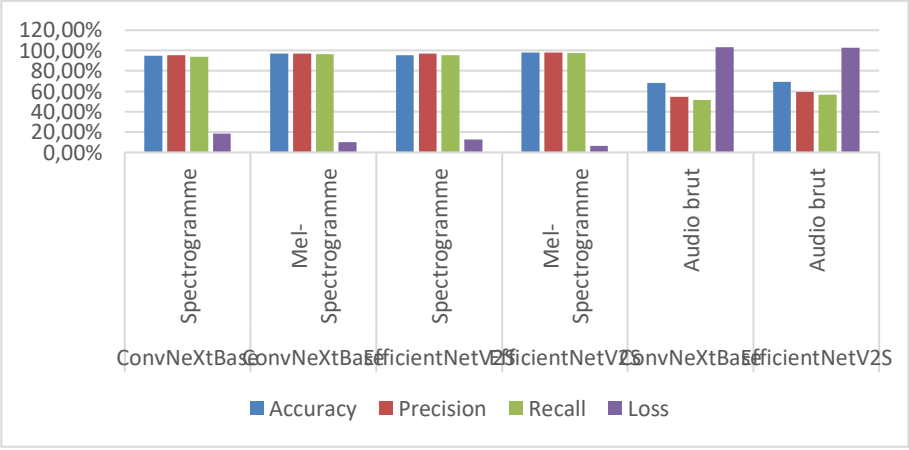

Figure III.30 : Histogramme comparatif des métriques d'évaluation selon la représentation du signal

Cette figure montre que les représentations temps-fréquence permettent d'obtenir de meilleurs résultats que le traitement direct du signal audio brut, aussi bien en termes d'accuracy que de recall et de precision. En revanche, l'utilisation du signal brut reste moins performante, ce qui confirme l'intérêt d'une transformation préalable du signal pour améliorer la classification des sons environnementaux.

III.15 Conclusion

Au terme de ce chapitre, l'ensemble des expériences menées a permis de valider l'approche proposée pour la classification des bruits sonores environnementaux. Les résultats obtenus par les deux architectures ConvNeXtBase et EfficientNetV2S, testées sur deux types de représentations audio — Spectrogramme et Mel-spectrogramme — mettent en évidence l'impact déterminant du choix de la représentation acoustique sur les performances de classification. La configuration EfficientNetV2S — Mel-spectrogramme s'est imposée comme la plus efficace, avec une accuracy de 97.88%, ouvrant ainsi la voie à des perspectives prometteuses pour l'amélioration des systèmes de reconnaissance des sons environnementaux. Par ailleurs, l'étude complémentaire menée sur les données audio brutes a permis de comparer cette approche directe avec celle fondée sur les représentations temps-fréquence, et de confirmer que la transformation du signal reste plus performante pour cette tâche de classification.

CONCLUSION
GENERAL

Conclusion Générale

Le travail présenté dans ce mémoire a eu pour ambition de concevoir et d'évaluer une approche de classification des sons environnementaux fondée sur le transfert d'apprentissage, en s'appuyant sur des architectures de vision par ordinateur pré-entraînées sur des images à grande échelle. La démarche adoptée s'est structurée en trois temps. D'abord, les fondements théoriques de l'intelligence artificielle, de l'apprentissage automatique et des réseaux de neurones convolutifs ont été posés afin de comprendre comment ces modèles extraient des représentations hiérarchiques discriminantes, une capacité mise à profit ici dans le domaine sonore.

Ensuite, le corpus d'étude a été constitué à partir de la base de données ESC-50, en retenant 26 classes représentatives. Les signaux audio ont été convertis en spectrogrammes et en Mel-spectrogrammes pour permettre l'exploitation des modèles de vision sur des données originellement unidimensionnelles. Si le spectrogramme offre une description complète de l'énergie, le Mel-spectrogramme applique une échelle non linéaire calquée sur la perception humaine, produisant une représentation plus compacte et centrée sur les informations pertinentes. L'ensemble a subi un traitement rigoureux et une augmentation de données pour renforcer la généralisation des modèles.

La phase expérimentale, menée sur Google Colab avec GPU, a permis d'adapter les architectures ConvNeXtBase et EfficientNetV2S à cette tâche de classification. Les résultats confirment la pertinence du transfert d'apprentissage pour les bruits environnementaux, les performances des deux modèles étant satisfaisantes. Cependant, une hiérarchie nette se dégage puisque la combinaison d'EfficientNetV2S avec le Mel-spectrogramme surpasse systématiquement les autres configurations. Grâce à son entraînement progressif, EfficientNetV2S capte mieux les structures fines, tandis que ConvNeXtBase, malgré sa puissance inspirée des Transformers, est limité par la taille modeste du jeu de données.

En conclusion, cette étude apporte une double contribution en confirmant l'efficacité économique du transfert d'apprentissage sur de petits corpus et en démontrant l'influence déterminante du Mel-spectrogramme par rapport au spectrogramme linéaire. Néanmoins, la taille réduite du sous-ensemble de classes et le nombre limité d'architectures constituent des contraintes qui incitent à la prudence. Ces limites ouvrent la voie à des travaux futurs prometteurs, tels que l'extension à l'ensemble des 50 classes d'ESC-50, l'intégration de

modèles plus récents, l'exploration de l'apprentissage auto-supervisé ou encore l'optimisation fine des hyperparamètres.

BIBLIOGRAPHIE

Bibliographie :

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

[2]. Gelly, G. (2017). *Réseaux de neurones récurrents pour le traitement automatique de la parole* [Thèse de doctorat, Université Paris-Saclay].

[3]. Grumiaux, P.-A. (2021). *Deep learning pour le comptage et la localisation de sources de parole avec des signaux ambisoniques* [Thèse de doctorat, Université Grenoble Alpes].

[4]. Bagghi, R. (2024). *Applications des méthodes de réseaux de neurones profonds sur des données climatiques : cas d'étude sur la prévision des échouements des algues Sargasses en Guadeloupe* [Mémoire de master, Université des Antilles].

[5]. Petriu, E. M. (n.d.). *Neural networks: Basics*. University of Ottawa.

[6]. Kukreja, H., Bharath, N., Siddesh, C. S., & Kuldeep, S. (2016). An introduction to artificial neural network. *International Journal of Advance Research and Innovative Ideas in Education (IJARIIE), 1*(5).

[7]. Gupta, G., Khatri, G., & Sharma, C. (2025). Artificial neural networks. *Pratibodh: A Journal for Engineering*.

[8]. Hugging Face. (n.d.). *Introduction aux données audio*. Hugging Face Audio Course. Retrieved from https://huggingface.co/learn/audio-course/fr/chapter1/audio_data

[9]. Florent, M. (2024). *Neurones artificiels et neurones biologiques*. Tree Learning. https://www.tree-learning.fr/plateforme-lms-elearning/neurones-artificiels/

[10]. Ashtari, H. (2022, August 3). *What is a neural network? Definition, working, types, and applications in 2022*.

[11]. Boumaza, K., & Mamou, F. Z. (2022). *Apprentissage profond pour la reconnaissance des composants électroniques* [Mémoire de master, Centre Universitaire Abdallah Morsli].

[12]. Piczak, K. J. (2015). *ESC-50: Dataset for environmental sound classification*. ACM Multimedia. https://github.com/karolpiczak/ESC-50

[13]. Bennanni, Z., & Khour, K. (2020). *Classification des événements audio environnementaux à l'aide de réseaux de neurones convolutifs CNN* [Mémoire de master, Université Saad Dahlab de Blida].

[14]. Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

[15]. Alake, R. (2020, August 14). *Implementing AlexNet CNN architecture using TensorFlow 2.0+ and Keras*.

[17]. Askarizade, M. (2026). Efficient Alzheimer's disease classification using transfer learning with EfficientNetV2S. *Journal of Emerging Engineering Technologies*.

[18]. Abd El-Aziz, A. A., Mahmood, M. A., & Abd El-Ghany, S. (2024). A robust EfficientNetV2-S classifier for predicting acute lymphoblastic leukemia based on cross validation. *Symmetry*.

[19]. Talend. (n.d.). *Tout savoir sur le machine learning*. https://www.talend.com/fr/resources/is-machine-learning/

[20]. Yousef, A., & Zineddine, D. M. (2019). *Développement et évaluation d'une nouvelle approche d'exploration des sous graphes sous plateforme big data*.

[21]. Bouchara, M. (2025, January 17). *Réseau de neurones profond (Deep Neural Network)*. CallMeNewton. https://www.callmenewton.fr/guide-ia/reseau-neurones-profond/

[23]. MathWorks. (n.d.). *Introduction aux réseaux neuronaux convolutifs (CNN)*. Retrieved 2026, from https://fr.mathworks.com/discovery/convolutional-neural-network.html

[24]. upGrad. (n.d.). *Basic CNN architecture: How the 5 layers work together*. Retrieved February 5, 2026, from https://www.upgrad.com/blog/basic-cnn-architecture

[25]. Li, B., & Lima, D. (2021). Facial expression recognition via ResNet-50. *International Journal of Cognitive Computing in Engineering, 2*, 57-64.

[26]. Bonner, A. (2019, February 2). *The complete beginner's guide to deep learning: Convolutional neural networks and image classification*.

[27]. Asatani, N., et al. (2021). Classification of respiratory sounds using improved convolutional recurrent neural network. *Computers & Electrical Engineering, 94*, 107367.

[28]. Keras. (n.d.). *VGG16*. https://keras.io/api/applications/vgg/vgg16-function

[29]. Coursera Staff. (2025). *What is machine learning? Definition, types, and examples*. Coursera. https://www.coursera.org/articles/what-is-machine-learning

[30]. Benyettou, A. S. H., & Khadîdja. (2023). *Contrôle qualité et détection de défauts pour les systèmes de triage industriel par l'utilisation de deep learning* [Mémoire de master, Université Abou Bekr Belkaid].

[32]. McKinsey & Company. (n.d.). *An executive's guide to AI*. https://www.mckinsey.com/business-functions/mckinsey-analytics/ourinsights/an-executives-guide-to-ai

[33]. Bastien, L. (2018, October 19). *TensorFlow : tout savoir sur la bibliothèque machine learning open source*.

[34]. Paperspace. (n.d.). *Epoch*. https://machine-learning.paperspace.com/wiki/epoch

[36]. Machine Learning Mastery. (n.d.). *Understand the dynamics of learning rate on deep learning neural networks*. https://machinelearningmastery.com/understand-the-dynamics-oflearning-rate-on-deep-learning-neural-networks/

[37]. Mitchell, S., O'Sullivan, M., & Dunning, I. (2011). PuLP: A linear programming toolkit for Python. *The University of Auckland, Auckland, New Zealand*, 65.

[38]. Amin, H., Darwish, A., Hassanien, A. E., & Soliman, M. (2022). End-to-end deep learning model for corn leaf disease classification. *IEEE Access, 10*, 31103-31115.

[39]. Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies, 2*(1), 37–63.

[40]. Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment, 62*(1), 77–89. https://doi.org/10.1016/S0034-4257(97)00083-7

[41]. IBM. (n.d.). *Metrics: Accuracy*. https://www.ibm.com/docs/en/ws-and-kc?topic=metrics-accuracy

[42]. Powers, D. M. W. (2007). *Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation* (Technical Report SIE-07-001). School of Informatics and Engineering, Flinders University of South Australia.

[43]. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 6. https://doi.org/10.1186/s12864-019-6413-7

[44]. Kim, D. H. (2023). A study on an effective teaching of AI using Google Colab-based DCGAN deep learning model building for music data analysis and genre classification. *International Journal of Recent Technology and Engineering (IJRTE), 11*(6), 13–25. https://doi.org/10.35940/ijrte.E7351.0311623

[45]. Lutz, M. (2001). *Programming Python*. O'Reilly Media, Inc.

[46]. Paperspace. (n.d.). *Epoch*. https://machine-learning.paperspace.com/wiki/epoch

[47]. Talukder, M. A., Khalid, M., Kazi, M., Muna, N. J., Nur-e-Alam, M., Halder, S., & Sultana, N. (2024). A hybrid cardiovascular arrhythmia disease detection using ConvNeXt-X models on electrocardiogram signals. *Scientific Reports, 14*(1), 30366.

[48]. Rong, F. (2016, December). Audio classification method based on machine learning. In *2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)* (pp. 81-84). IEEE.

[49]. Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(4), e1312.

[50]. De Matteis, L., Janny, S., Nathan, S., & Shu-Quartier, W. (2022). *Introduction à l'apprentissage automatique*. Culture Sciences de l'Ingénieur.

[51]. Taye, M. M. (2023). Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers, 12*(5), 91.

[52]. Red Hat. (n.d.). *What is deep learning?* https://www.redhat.com/fr/topics/ai/what-is-deep-learning

[53]. Currie, G., & Rohren, E. (2021). Intelligent imaging in nuclear medicine: The principles of artificial intelligence, machine learning and deep learning. *Seminars in Nuclear Medicine*, *51*(2), 102-111.

[54]. Ghiassi, M., & Saidane, H. (2005). A dynamic architecture for artificial neural networks. *Neurocomputing*, *63*, 397-413.

[55]. André, N. (2025). *Représentations et fonctions d'activation réelles et hyper-complexes dans les réseaux de neurones pour le traitement du signal* [Thèse de doctorat, Université d'Avignon].

[56]. Liu, C., Gardner, S. J., Wen, N., Elshaikh, M. A., Siddiqui, F., Movsas, B., & Chetty, I. J. (2019). Automatic segmentation of the prostate on CT images using deep neural networks (DNN). *International Journal of Radiation Oncology*Biology*Physics, 104*(4), 924-932.

[57]. Li, G., Hari, S. K. S., Sullivan, M., Tsai, T., Pattabiraman, K., Emer, J., & Keckler, S. W. (2017). Understanding error propagation in deep learning neural network (DNN) accelerators and applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-12).

[58]. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1-6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

[59]. Giovannangeli, L., Lalanne, F., Auber, D., Giot, R., & Bourqui, R. (2021). Deep neural network for drawing networks, (DNN) 2. In *International Symposium on Graph Drawing and Network Visualization* (pp. 375-390). Springer.

[60]. Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J. M., Fnaiech, F., & Moreau, E. (2017). Détection de la fumée et du feu par réseau de neurones convolutifs. In *Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle*.

[61]. Ma, L., Guo, R., Gomez, A., Liu, T., & Yang, J. (2020). Automatic segmentation of the prostate on CT images using deep neural networks. *International Journal of Radiation Oncology*Biology*Physics*. https://doi.org/10.1016/j.ijrobp.2019.12.037

[63]. Boddapati, V., Petef, A., Rasmusson, J., et al. (2017). Classifying environmental sounds using image recognition networks. *Procedia Computer Science, 112*, 2048-2056.

[64]. Ashurov, A., Zhou, Y., Shi, L., Zhao, Y., & Liu, H. (2022). Environmental sound classification based on transfer-learning techniques with multiple optimizers. *Electronics, 11*(15), 2279.

[65]. Mushtaq, Z., & Su, S.-F. (2021). Environmental sound classification using a regularized deep convolutional neural network with data augmentation. *Applied Acoustics, 176*, 107389.

[66]. Fang, Z., Yin, B., Du, Z., & Huang, X. (2022). Fast environmental sound classification based on resource adaptive convolutional neural network. *Scientific Reports, 12*, 6599.

[67]. Abdoli, S., Cardinal, P., & Koerich, A. L. (2019). End-to-end environmental sound classification using a 1D convolutional neural network. *Applied Soft Computing, 80*, 34-44.

[68]. Oppenheim, A. V., & Schafer, R. W. (2010). *Discrete-time signal processing* (3rd ed.). Pearson.

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

[70]. Salamon, J., & Bello, J. P. (2015). Environmental sound classification with convolutional neural networks. In *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1–6).

[71]. Stevens, S. S., Volkmann, J., & Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America, 8*(3), 185–190.

[72]. Young, S., Evermann, G., Gales, M., et al. (2006). *The HTK book* (Version 3.4). Cambridge University Engineering Department.

[73]. Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. Prentice Hall.

[74]. Piczak, K. J. (2015). Environmental sound classification with convolutional neural networks. In *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1–6).

[75]. Talukder, M. A., Hossen, R., Uddin, M. A., Uddin, M. N., Acharjee, U. K., & Sultana, N. (2024). A hybrid cardiovascular arrhythmia disease detection using ConvNeXt-X models on electrocardiogram signals. *Scientific Reports, 14*, 30366. https://doi.org/10.1038/s41598-024-81992-w

[76]. Piczak, K. J. (2015). ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia* (pp. 1015–1018).

[77]. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097–1105.

[78]. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 248–255).

[79]. Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11966–11976).

[80]. Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. In *Proceedings of the 38th International Conference on Machine Learning (ICML)* (pp. 10096–10106).

[81]. Hu, P., & Tang, H. (2018, May). The principle and application of deep learning algorithm. In *2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018)* (pp. 835-838). Atlantis Press.

[82]. André, N. (2025). *Représentations et fonctions d'activation réelles et hyper-complexes dans les réseaux de neurones pour le traitement du signal* (Doctoral dissertation, Université d'Avignon).

[83]. Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, *6*(12), 310-316.

[84]. Brownlee, J. (2019, October 9). A gentle introduction to cross-entropy for machine learning. Machine Learning Mastery. https://machinelearningmastery.com/cross-entropy-for-machine-learning/