



HAL
open science

Analyse des problématiques liées à la reconnaissance de sons ambiants en environnement réel

Nicolas Turpault

► **To cite this version:**

Nicolas Turpault. Analyse des problématiques liées à la reconnaissance de sons ambiants en environnement réel. Informatique [cs]. Université de Lorraine, 2021. Français. NNT : 2021LORR0108 . tel-03304880

HAL Id: tel-03304880

<https://hal.inria.fr/tel-03304880>

Submitted on 28 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ
DE LORRAINE

THÈSE DE DOCTORAT

Nicolas Turpault

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Lorraine
Mention Informatique

École doctorale : IAEM

Unité de recherche : Laboratoire Lorrain de Recherche en Informatique et ses Applications
UMR 7503

Soutenue le 31 mai 2021

Thèse N° :

ANALYSE DES PROBLÉMATIQUES LIÉES À LA RECONNAISSANCE DE SONS AMBIANTS EN ENVIRONNEMENT RÉEL

JURY

Rapporteur : **Hervé GLOTIN**, Professeur, Université de Toulon, France
Rapporteur : **Geoffroy PEETERS**, Professeur, Télécom ParisTech, France
Examinatrice : **Annamaria MESAROS**, Maître de Conférences, Université de Tampere, Finlande
Examineur : **Christophe CERISARA**, Chargé de Recherche, CNRS-LORIA, France
Directeur de thèse : **Emmanuel VINCENT**, Directeur de Recherche, Inria Nancy - Grand Est, France
Co-directeur de thèse : **Romain SERIZEL**, Maître de Conférences, Université de Lorraine, France

Résumé

Notre vie est constamment bercée par les sons ambiants. Du bruit d'une voiture qui passe à un oiseau qui chante, de l'eau qui coule dans notre douche aux bruits de notre clavier, les sons ambiants sont partout. Les humains sans pertes auditives reconnaissent inconsciemment les sons qui les entourent et prennent de nombreuses décisions de la vie quotidienne en tenant compte des sons ambiants (réactions à des pleurs de bébé ou une alarme par exemple). Durant ces dernières années, la recherche autour de l'analyse automatique de ces sons ambiants s'est développée rapidement.

L'analyse des sons ambiants est un problème difficile à résoudre en raison de la complexité des scènes sonores et de leur manque de structure apparente. Les événements sonores qui constituent les scènes sonores sont très variés et de nombreux événements peuvent être actifs simultanément. Afin de reconnaître les événements sonores de façon automatique, on a généralement recours à des méthodes d'apprentissage automatique. Les méthodes par apprentissage profond sont devenues très populaires ces dernières années grâce à leurs performances élevées pour des tâches diverses dont l'analyse de sons ambiants. Les méthodes d'apprentissage s'appuient sur l'utilisation de jeux de données contenant les événements que l'on souhaite reconnaître. Dans l'idéal, ces jeux de données contiennent des annotations concernant l'activité liée à chacune des classes d'événements sonores et éventuellement à leur temporalité (on parle alors d'annotations fortes). Ces dernières années, des jeux de données fortement annotés ont été collectés et publiés pour permettre l'analyse de sons ambiants, mais ils sont souvent composés d'une faible quantité de données qui ne sont pas toujours enregistrées en conditions réelles. Obtenir des annotations fortes coûte cher, et il est donc difficile d'obtenir un gros jeu de données fortement annotées. En revanche, la collecte de données non annotées ou annotées partiellement et sans indication de temporalité (annotations faibles) est plus facile. C'est dans ce cadre que s'inscrit cette thèse.

Nous proposons de traiter le problème de la reconnaissance d'événements sonores en environnement domestique en utilisant des données non annotées et faiblement annotées. Le but est d'analyser les problèmes qui surviennent lors d'un scénario réel de reconnaissance d'événements sonores au sein d'une maison pour permettre l'assistance aux personnes en perte d'autonomie ou rendre la maison intelligente. Afin d'analyser ce problème, nous avons proposé une tâche de détection d'événements sonores dans un challenge international d'analyse de sons ambiants. Pour cette tâche nous avons défini un problème proche d'un scénario réel pour permettre l'analyse scientifique des différents problèmes qui apparaissent dans l'analyse de sons ambiants en environnement réel. Nous proposons un jeu de données pour permettre des analyses détaillées des problèmes scientifiques à résoudre pour permettre l'évolution continue de la tâche. Nous nous focalisons ensuite sur le problème de l'apprentissage semi-supervisé qui permet l'apprentissage de systèmes uti-

lisant des données annotées et des données non annotées. Cette analyse se concentre sur l'apprentissage d'une représentation qui serait utile pour des applications finales d'étiquetage ou de détection d'événements sonores. Nous analysons enfin l'impact de l'annotation faible des données dans l'apprentissage d'un système de reconnaissance d'événements sonores afin de proposer des conseils pour l'annotation faible des jeux de données ou des pistes de solutions.

Abstract

We're constantly surrounded by ambient sounds. From a car passing by to a bird's song or from the running water in the shower to the sound of a keyboard, ambient sounds are everywhere. Humans without hearing loss unconsciously recognize them and take multiple decisions using the information provided by ambient sounds in their everyday life (reaction to a baby crying or to an alarm for example). In the last years, the research interest in automatic ambient sound analysis has rapidly grown. Ambient sound analysis is a difficult problem because of the complexity of the sound scenes and their lack of apparent structure.

Sound events constituting these sound scenes are various and multiple events can appear simultaneously. To recognize sound events automatically, machine learning methods are usually used, in particular deep learning methods due to their good performance on a variety of tasks including ambient sound analysis. These methods require a training dataset containing the sound events to be recognized. Ideally, the dataset contains labels indicating the type of events and their time positions in the audio clips (strong labels). In recent years, some strongly annotated datasets have appeared that are designed for ambient sound analysis, but they usually contain only a small amount of data and are rarely recorded in real conditions. Strong annotations are expensive to collect, making it difficult to acquire a large scale strongly labeled dataset. However, collecting data without labels or with partial labels indicating the presence of some events without their time information (weak labels) is easier. This thesis fits in this context.

We propose to address the problem of sound event recognition in domestic environments using unlabeled and weakly labeled data. Our goal is to analyze the different problems that can appear in a real world scenario of sound event recognition in domestic environment with applications to assisted living and smart house. To analyse this problem we have organized a domestic sound event detection task in an international ambient sound analysis challenge. We have defined this task in such a way that it allows us to analyze the different problems appearing in a real world scenario. We have collected, annotated and shared a dataset designed for this analysis. From 2018 to 2020, we have organized three evaluation campaigns to allow for a detailed analysis of the systems submitted by participants and a continuous improvement the task definition. Then, we focus on the problem of learning systems using both labeled and unlabeled training data (semi-supervised learning). The analysis concentrates on learning a representation which could be useful for a variety of tasks in sound event detection or tagging. Finally, we analyze the impact of weak labels in the training dataset of a sound event recognition system to understand if this is the main problem of a sound event recognition system and provide advice for the labelling of real world data.

Remerciements

Il me faudrait plus de quelques mots endiablés
Pour réussir à vraiment tous vous remercier.
Dans ces remerciements, je ferai de mon mieux
Pour montrer ma gratitude mais sans faire d'envieux.

Ce texte, pour une thèse n'est pas conventionnel,
Sans doute à l'image de mon travail au cours d'elle.
Tout d'abord merci à Emmanuel et Romain,
Bienveillance et grand soutien depuis le jour un.

Sans vous rien de tout ça n'aurait été possible.
Vous m'avez rassuré, cadré, accompagné.
Dès le moindre tracas, vous étiez disponibles,
Malgré mes retards sans jamais vous agacer.

Merci à tous les collègues de ces 3 années.
Baldwin, Ken, Sunit et Mathieu tout au début,
Guillaume, Elodie et Antoine sont arrivés,
Puis Nicolas, Adrien, Imran, et Manu.

Ashwin, Lou, Tulika, Théo, Léna, Diego,
Michel, Mathieu F et tous ceux aux apéros.
Du travail à la rigolade, ou du goûter
aux diners, tous d'une grande aide pour me motiver.

Merci Denis, Hélène et tous les permanents
pour les bons moments et échanges intéressants.
De cette équipe, je garde vraiment de bons moments,
Malgré les va-et-vient de personnes incessants.

Jean-Yves, Marie et Annabelle, quel bel accueil,
et l'orga des meetups, avec vous un plaisir.
Mes excuses aux collègues manquants de cette feuille,
Un simple échange peut être important, sans mentir.

DCASE, DCASE, thanks for the opportunities
Toni, Annamaria, Sacha, Frederic

All Google and Audio analytics buddies,
All my co-authors, so inspiring, dynamic.

Plus de trois années ont passé pour terminer.
L'entourage, un moteur pour ne pas perdre pied.
Un partage, dont ils n'imaginent pas les bienfaits.
Rire, parler, pour un instant oublier, c'est parfait.

Les copains d'enfance, de Nantes, du Sud, de Nancy.
Un plein d'insouciance, pour un chemin sans souci.
Mathieu, parler sans rentrer, assis ou debout.
Elo, Candice, sans blaguer, qu'aurais-je fait sans vous ?

Victor, Thomas, un tour de France avant soutenance,
Paul, Simon, Laura, Lou, motivation vers Lyon.
La bande de Maulev, rappelle ce parfum d'enfance.
Les amis, libèrent l'esprit, rappellent la raison.

Pernelle, Isalis, Lubin, Niels, Romain merci.
Accueil, générosité, partage et valeurs
que je partage tant. Nancy, oh oh oh, Nancy.
À cette ville, vous lui donnez beaucoup de couleurs.

Finalement merci à ceux qui m'ont supporté
au quotidien et subirent ce rythme effréné.
Ma famille, je vous suis tellement reconnaissant
D'être toujours présents et si encourageants.
Cette thèse ne serait pas, sans un enseignant.
Frédéric Précioso, ce domaine inspirant,
Il me le fait découvrir, surtout l'apprécier.
Ce rôle d'enseignant est souvent sous-estimé.

À tous ceux qui sont venus ici pour me lire,
Je n'ai qu'un seul mot à dire, un honnête merci.

Table des matières

Liste des figures	vii
Liste des tableaux	ix
1 Introduction	1
1.1 Motivation	1
1.2 Contexte	2
1.3 Problématique générale	3
1.4 Problèmes traités et contributions	3
1.4.1 DCASE Tâche 4	3
1.4.2 Apprentissage semi-supervisé	4
1.4.3 Exploitation des annotations faibles	4
1.4.4 Liste des publications	4
1.5 Plan du document	5
2 État de l'art	7
2.1 Contexte	7
2.1.1 Un peu d'histoire	7
2.1.2 Les applications	8
2.1.3 Les tâches méthodologiques	11
2.1.4 Le challenge DCASE	14
2.2 Analyse de sons ambiants	15
2.2.1 Système d'analyse de sons ambiants	15
2.2.2 Représentation de bas niveau	15
2.2.2.1 Représentation de bas niveau issue du traitement du signal	15
2.2.2.2 Choix d'une représentation de bas niveau	18
2.2.2.3 Représentation de bas niveau apprise	18
2.2.3 Représentation de haut niveau	19
2.2.4 Classifieurs	21
2.2.5 Problèmes méthodologiques	22
2.3 Reconnaissance d'événements sonores en environnement réel	24
2.3.1 Contexte	24
2.3.2 Jeux de données	25
2.3.3 Exploitation de données partiellement annotées	27
2.3.4 Annotations faibles	31

3	Détection d'événements sonores en environnement domestique (Tâche 4 du Challenge DCASE)	35
3.1	Définition de la tâche et évaluation	35
3.1.1	Application	35
3.1.2	Définition de la tâche	36
3.1.3	Évaluation	37
3.1.3.1	Calcul de la métrique	39
3.1.3.2	Choix de la métrique	42
3.2	Corpus DESED	44
3.2.1	Motivation	44
3.2.2	Présentation générale	44
3.2.3	Collecte et annotation des données	46
3.2.3.1	Données issues de corpus de vidéos en ligne	47
3.2.3.2	Données synthétiques	51
3.2.4	Génération des données synthétiques	53
3.2.5	Résumé	55
3.3	Évaluation officielle de la Tâche 4	56
3.3.1	DCASE 2018	58
3.3.2	DCASE 2019	59
3.3.3	DCASE 2020	61
3.4	Conclusion	62
4	Systèmes de référence et analyse des résultats	63
4.1	Les systèmes de référence	63
4.1.1	DCASE 2018	63
4.1.2	DCASE 2019	66
4.1.3	DCASE 2020	68
4.1.4	Étude ablative du système de référence de DCASE 2020	71
4.2	Analyse des résultats	75
4.2.1	DCASE 2018	76
4.2.2	DCASE 2019	79
4.2.3	DCASE 2020	83
4.3	Conclusion	88
5	Apprentissage de représentation semi-supervisé	89
5.1	Description du problème semi-supervisé	89
5.1.1	Apprentissage par triplets	90
5.1.2	Stratégie de tirage supervisée	90
5.1.3	Stratégie de tirage non supervisée basée sur des transformations	91
5.2	Stratégies semi-supervisées proposées	92
5.3	Protocole expérimental	94
5.3.1	Jeu de données	94
5.3.2	Stratégies de tirage utilisées	94
5.3.3	Représentation de bas niveau et transformations	95

5.3.4	Modèle	96
5.3.4.1	Modèles de référence	96
5.3.4.2	Triplets	96
5.3.4.3	RNN appliqué au spectrogramme	96
5.3.4.4	VGGish	96
5.3.4.5	Professeur moyen	97
5.3.5	Métrique et validation	97
5.4	Analyse de performance de l'apprentissage par triplets	97
5.5	Comparaison de l'apprentissage par triplets avec l'apprentissage par clas- sifieur	100
5.5.1	Représentation de haut niveau apprise sur Audioset	101
5.5.2	Comparaison des méthodes d'apprentissage supervisé	102
5.5.3	Comparaison des méthodes d'apprentissage semi-supervisé	103
5.6	Conclusion	104
6	Impact des annotations faibles	107
6.1	Définition du problème	108
6.2	Isolation du problème	109
6.2.1	AAF	110
6.2.2	ETA AF	112
6.2.3	DAAF	112
6.3	Impact attendu des annotations faibles sur l'apprentissage de représentation	112
6.3.1	Apprentissage de bout-en-bout	113
6.3.2	Apprentissage par triplets	113
6.3.3	Réseau prototype	113
6.3.4	Métrique de validation	114
6.4	Impact attendu des annotations faibles sur l'agrégation temporelle	115
6.4.1	Agrégation moyenne	116
6.4.2	Agrégation max	117
6.4.3	Agrégation softmax	117
6.4.4	Agrégation L_p	118
6.4.5	Agrégation par attention	118
6.5	Description des expériences	119
6.5.1	Apprentissage de représentation	119
6.5.2	Impact de l'agrégation et de la durée des clips	120
6.5.3	Métrique d'évaluation	121
6.6	Résultats et discussion	121
6.6.1	Apprentissage de représentation	121
6.6.2	Impact de l'agrégation	124
6.6.2.1	Impact de l'agrégation à l'évaluation	124
6.6.2.2	Impact de la segmentation à l'apprentissage	125
6.6.2.3	Masquage des entrées et/ou des sorties à l'apprentissage et/ou à l'évaluation	128

6.6.2.4	Impact de l'agrégation à l'apprentissage	129
6.6.3	Impact de la durée du clip et de la densité d'événement	130
6.6.3.1	Impact de la durée du clip	130
6.6.3.2	Impact de la densité d'événement	133
6.7	Conclusion	135
7	Conclusion et perspectives	137
7.1	Conclusion	137
7.2	Poursuite des travaux	139
7.2.1	Perspectives théoriques	139
7.2.2	Perspectives pratiques	141
	Bibliographie	143

Liste des figures

2.1	Classification de scènes sonores.	11
2.2	Étiquetage d'événements sonores.	12
2.3	Détection d'événements sonores.	13
2.4	Étapes d'un système d'analyse de sons ambiants illustrées par un exemple.	16
2.5	Banc de filtres Mel permettant le calcul d'un spectrogramme en échelle Mel.	17
2.6	Annotations faibles et fortes pour la reconnaissance d'événements sonores.	32
3.1	Comparaison des différentes méthodes de comptage associées aux métriques utilisées pour la détection d'événements sonores.	38
3.2	Détermination des VP, FP, FN, VN et calcul de la précision et du rappel.	40
3.3	Illustration du problème d'une mesure par segment.	42
3.4	Tolérance utilisée pour le calcul de la métrique de la Tâche 4.	43
3.5	Jeux de données à disposition des participants de la tâche.	46
3.6	Utilisation des différents jeux de données d'apprentissage et de test.	57
4.1	Illustration d'un réseau convolutionnel et récurrent.	64
4.2	Illustration des cas où la métrique utilisée est non permissive.	66
4.3	Définition du modèle « professeur moyen ».	67
4.4	Définition du système utilisant le modèle « professeur moyen ».	68
4.5	Représentation du système utilisant la séparation de sources avec une agrégation des sources à posteriori.	70
4.6	Distribution des classes d'événements sonores dans le jeu de test.	76
4.7	Performance de segmentation selon la tolérance temporelle choisie.	78
4.8	Performance des 10 meilleurs systèmes soumis en 2019 en fonction du rapport signal-à-bruit.	81
4.9	Performance de segmentation des classes d'événements longs en fonction de la localisation temporelle de l'événement.	81
4.10	Distribution des instants de début et de fin des événements longs dans les jeux d'apprentissage et d'évaluation synthétiques.	82
4.11	Précision et rappel sur le jeu de données 60 s	85
4.12	Performance des classes d'événements longs en fonction de la localisation temporelle de l'instant de début des événements.	86
4.13	Performance en fonction de la durée des événements sonores dans le jeu de données one_event	86
4.14	Performance en fonction du rapport de puissance entre les événements cibles et les événements non-cibles.	87
4.15	Performance en fonction de la réverbération.	88

5.1	Création des triplets de façon supervisée.	91
5.2	Exemples positifs obtenus par les stratégies appliquées aux données non annotées.	92
5.3	Illustration des stratégies de tirage.	93
5.4	Visualisation des représentations de haut niveau de 3 classes obtenues par les différentes méthodes présentées.	99
5.5	F-mesure obtenue à partir des représentations extraites de chaque couche du modèle VGGish.	102
6.1	Illustration de la différence entre annotation forte et faible.	109
6.2	Distribution de la durée des événements sonores par classe dans l'ensemble d'apprentissage.	111
6.3	Représentation des différents jeux de données synthétiques créés pour ana- lyser les annotations faibles.	111
6.4	Méthodes d'agrégation utilisées.	116
6.5	Agrégation moyenne et max ($\tau = 0, 5$).	117
6.6	Comparaison de l'agrégation L_p avec les autres fonctions d'agrégation fixes.	119
6.7	F-mesure du système de référence pré-appris sur le corpus DESED et éva- lué sur le jeu de données AAF pour les fonctions d'agrégation moyenne, max et softmax.	124
6.8	F-mesure du système de référence pré-appris sur le corpus DESED et éva- lué sur le jeu de données AAF pour les fonctions d'agrégation max, L_p , et par attention.	126
6.9	Masquage des entrées et/ou des sorties du modèle.	127
6.10	F-mesure du système de référence ré-appris et évalué sur le jeu de données AAF en fonction de la durée des clips.	130
6.11	F-mesure du système de référence ré-appris et évalué sur le jeu de données AAF en fonction de la durée des clips d'apprentissage et de la durée des événements d'évaluation.	132
6.12	F-mesure du système de référence ré-appris sur le jeu de données ETAAF en fonction de la durée des événements tronqués à l'apprentissage	133

Liste des tableaux

2.1	Tâches et applications du challenge DCASE.	14
2.2	Jeux de données faiblement annotés.	26
2.3	Jeux de données fortement annotés.	28
3.1	Liste des classes d'événements d'intérêt.	47
3.2	Statistiques par classe du corpus DESED.	48
3.3	Jeux de données composant le corpus DESED.	56
3.4	Nombre de participants, d'équipes et de soumissions à la Tâche 4.	58
3.5	Performance des systèmes soumis à la Tâche 4 du Challenge DCASE 2018.	58
3.6	Performance des systèmes soumis à la Tâche 4 du Challenge DCASE 2019.	60
3.7	Performance des systèmes soumis à la Tâche 4 du Challenge DCASE 2020.	61
4.1	Performance du système de référence 2018.	65
4.2	Impact des caractéristiques d'entrées sur le jeu de données de test.	69
4.3	Performance des systèmes de référence de 2019 et de 2020.	69
4.4	Performances en fonction des jeux de données utilisés à l'apprentissage.	72
4.5	Performances en fonction de l'utilisation du changement de ton.	73
4.6	Performances en fonction de l'application ou non de réverbération.	73
4.7	Performances en fonction du rapport signal-à-bruit (SNR) du bruit appliqué à l'entrée du réseau professeur.	74
4.8	Performances en fonction de l'accroissement progressif ou non du taux d'apprentissage et du poids du coût de cohérence et de la valeur de ce poids.	74
4.9	Performances sur les jeux de test et d'évaluation publics en fonction de l'application ou non de réverbération et du changement de ton sur le jeu de validation synthétique.	75
4.10	Performance pour chacune des classes d'intérêt pour les 10 meilleurs systèmes soumis en 2018.	77
4.11	Performance des 5 meilleurs systèmes pour les classes d'événements courts ainsi que leur classement.	77
4.12	F-mesure par événement 5 meilleurs systèmes pour les classes d'événements longs ainsi que leur classement.	77
4.13	Performance des 10 meilleurs systèmes soumis en 2019 sur les données d'évaluation synthétiques dégradées.	83
4.14	Performance des 10 meilleurs systèmes soumis en 2020 en fonction de la durée du clip.	85
5.1	Stratégies de tirage des triplets.	95

5.2	F-mesure obtenue par les cinq stratégies d'apprentissage par triplets en fonction du nombre de triplets.	98
5.3	F-mesure obtenue par les stratégies d'apprentissage par triplets semi-supervisé en fonction du nombre de triplets issus de données annotées et non annotées.	100
5.4	Taille des représentations extraites de chaque couche du modèle VGGish. .	101
5.5	F-mesure obtenue à partir des différentes méthodes d'apprentissage supervisé.	103
5.6	F-mesure obtenue à partir des différentes méthodes d'apprentissage semi-supervisé.	104
6.1	Nombre d'événements Freesound isolés utilisés dans chaque ensemble de données.	110
6.2	F-mesure obtenue par les trois méthodes d'apprentissage de représentation sur le jeu de données ETAAF	122
6.3	F-mesure obtenue par les trois méthodes d'apprentissage de représentation sur le jeu de données AAF	123
6.4	F-mesure du système de référence ré-appris et évalué sur le jeu de données AAF, en fonction de la segmentation et la normalisation par lots utilisée. .	127
6.5	F-mesure du système de référence ré-appris et évalué sur le jeu de données AAF en fonction du scénario et du coût d'apprentissage	128
6.6	F-mesure du système de référence ré-appris et évalué sur le jeu de données AAF en fonction du masquage utilisé.	128
6.7	F-mesure du système de référence ré-appris et évalué sur le jeu de données AAF en fonction de l'agrégation utilisée à l'apprentissage.	129
6.8	F-mesure du système de référence ré-appris et évalué sur le jeu de données DAAF en fonction de la durée des clips.	134

Liste des acronymes

- CLEAR** *Classification des Événements, Activités et Relations (Classification of Events, Activities and Relationships)*
- CNN** *réseau de neurones convolutionnel (convolutional neural network)*
- CRNN** *réseau de neurones convolutionnel et récurrent (convolutional recurrent neural network)*
- DCASE** *détection et classification des scènes et événements sonores (detection and classification of acoustic sound scenes and events)*
- DESED** *détection d'événements sonores en environnement domestique (domestic environment sound event detection)*
- DNN** *réseau de neurones profond (deep neural network)*
- FFT** *transformée de Fourier rapide (fast Fourier transform)*
- FN** *faux négatif*
- FP** *faux positif*
- FUSS** *séparation de source universelle (free universal sound separation)*
- GMM** *modèle de mélange de gaussiennes (Gaussian mixture model)*
- GRU** *unités récurrentes par portes (gated recurrent units)*
- GTCC** *coefficients cepstraux gammatone (gammatone cepstral coefficients)*
- HMM** *modèle de Markov caché (hidden Markov model)*
- MFCC** *coefficients cepstraux en échelle Mel (Mel frequency cepstral coefficients)*
- MIL** *apprentissage multi-instance (multi-instance learning)*
- MLP** *perceptron multicouche (multilayer perceptron)*
- NMF** *factorisation matricielle non-négative (non-negative matrix factorization)*
- PCA** *analyse en composantes principales (principal component analysis)*
- PSDS** *score de détection sonore polyphonique (polyphonic sound detection score)*
- RBM** *machine de Boltzmann restreinte (restricted Boltzmann machine)*
- RNN** *réseau de neurones récurrent (recurrent neural network)*
- ROC** *caractéristique de performance (receiver operating characteristic)*
- SINS** *interfaçage du son en essaim (Sound INterfacing through the Swarm)*
- STFT** *transformée de Fourier à court terme (short time Fourier transform)*
- SVM** *machine à vecteurs de support (support vector machine)*

VAT *apprentissage antagoniste virtuel (virtual adversarial training)*

VN *faux négatif*

VP *vrai positif*

1 Introduction

1.1 Motivation

Les sons ambiants font partie de notre quotidien. Ils sont partout, que ce soit dans la rue (véhicules, travaux, *etc.*), dans la nature (animaux, vent, *etc.*), dans nos maisons (aspirateur, eau qui coule, *etc.*) ou bien sur notre lieu de travail (machines, claviers, *etc.*).

Les humains peuvent entendre des sons entre 20 Hz et 20 kHz, et reconnaissent des sons environnementaux tous les jours de façon naturelle sans s'en rendre compte. On peut décrire ces sons environnementaux par des verbes et des actions (aspirer fortement) [Lemaitre and Heller, 2013, Säger et al., 2016] ou bien par une classe présente dans une organisation sémantique (aspirateur) [Gemmeke et al., 2017] et même par des onomatopées (« whou-ou-ou ») [Sundaram and Narayanan, 2006]. L'information apportée par ces sons nous aide dans une multitude de tâches de la vie quotidienne pour permettre de comprendre un contexte ou bien d'obtenir des informations pour prendre une décision adaptée. Une application évidente est la prévention du danger qui est souvent caractérisée par une alarme ou une sirène ce qui permet aux humains de réagir avant même de voir le danger. On comprend aussi l'importance d'entendre une voiture approchant dans la rue ou bien d'entendre son bébé qui pleure chez soi sans pour autant les voir. C'est dans ce but qu'ont même été créés les outils tels que l'écoute-bébé.

Il est intéressant d'écouter les bruits approchant pour les identifier visuellement par la suite, tout comme il est intéressant d'entendre de l'eau qui coule chez soi pour détecter une éventuelle fuite ou un robinet resté ouvert. Les sons ambiants ont aussi un rôle à jouer pour l'urbanisation. Le bruit dans les villes est un problème de plus en plus fréquent. Il nous est important d'identifier les sources de sons dérangeants pour permettre de les réduire. Une autre application possible est l'étude de la biodiversité pour comprendre et analyser les possibles problèmes (dû au réchauffement climatique par exemple). Enfin, une application utile dans un grand nombre d'entreprises est la maintenance prédictive. Beaucoup de machines peuvent faire un bruit anormal avant de tomber en panne (une voiture par exemple) . Il est intéressant de détecter ce bruit rapidement pour prévenir la panne avant qu'elle ne survienne.

Toutes ces applications sont aujourd'hui réalisées de façon naturelle par l'homme puisqu'elles lui sont nécessaires. Certaines de ces applications sonores pourraient bénéficier de l'aide d'une machine pour aider l'homme à prendre des décisions adéquates. En effet, cela éviterait aux humains l'écoute active de ces sons qui est une tâche répétitive dont il est parfois difficile d'être précis et ces sons sont souvent désagréables. Sans parler des sourds et malentendants pour qui la reconnaissance automatique de sons ambiants pourrait par-

tiellement pallier leur handicap. La reconnaissance automatique de sons ambiants serait donc utile à une multitude de nouvelles applications comme l'urbanisation, l'étude des animaux, la maintenance prédictive, la surveillance, l'aide à la personne ou la domotique. Cependant, l'analyse des sons ambiants pose des défis supplémentaires par rapport à la détection et classification de la voix puisque les classes de sons ne sont pas définies de façon normative comme les phonèmes et elles ont des propriétés spectro-temporelles plus diverses que la voix.

1.2 Contexte

Récemment, la recherche s'est fortement développée autour de l'analyse des sons ambiants, favorisée par l'apparition d'une quantité importante de données disponibles. Une communauté s'est créée autour de la *détection et classification des scènes et événements sonores* (*detection and classification of acoustic sound scenes and events*) (DCASE) [Virtanen et al., 2017]. Le domaine étant récent, il évolue continûment et de nouvelles problématiques émergent chaque année.

Le but de la classification de scènes sonores est d'identifier le lieu d'enregistrement d'une scène sonore, par exemple « dans un bar ». Cette tâche est souvent assimilée à la classification d'activité comme « faire la cuisine » puisque les techniques de classification sont similaires.

La reconnaissance d'événements sonores consiste quant à elle à identifier les différents événements sonores qui constituent une scène. Pour une scène « dans un bar » par exemple, ces événements peuvent inclure des bruits de verres, de chaises, de parole, *etc.* La reconnaissance d'événements se décompose en deux sous-tâches : l'étiquetage d'événements sonores et la détection d'événements sonores. Savoir que dans un intervalle de temps donné il y a eu un ou plusieurs bruits de chaise et un ou plusieurs bruits de verres sans les compter ou les positionner précisément dans le temps correspond au SET alors que savoir leur fréquence et leur position temporelle correspond à la détection d'événements sonores. En plus de la position temporelle, la communauté s'intéresse aussi à la localisation spatiale de ces événements, on appelle cette tâche la détection et localisation d'événements sonores [Adavanne et al., 2019b].

Le problème de la reconnaissance d'événements sonores étant compliqué, le nombre de sons à reconnaître a d'abord été restreint et les sons ont été regroupés le plus souvent par lieux dans ce qui est appelé une ontologie (par exemple Audioset [Gemmeke et al., 2017]). Cette approche permet de regrouper les sons dans des catégories formalisées à l'avance, ce qui facilite l'apprentissage d'un système de classification. La qualité de l'ontologie et sa correspondance à l'application visée jouent alors un rôle crucial (ontologie par lieux, objets similaires, sons similaires, *etc.*). Plus récemment, la communauté a commencé à se rapprocher de ce qu'un humain pourrait faire, c'est-à-dire décrire les événements en langage naturel [Drossos et al., 2017]. Cette approche donne une information plus riche à propos du son, mais est aussi plus complexe. Deux sons proches ne vont pas seulement être annotés différemment, mais un même son peut avoir un nombre important de descriptions et chacune de ces descriptions peut contenir des ambiguïtés (plusieurs sens d'un même

mot) ou peut-être écrite différemment suivant l'application (« un chien aboyant sur une moto » ou « une moto poursuivie par un chien » selon que notre application est focalisée sur le chien ou la moto).

1.3 Problématique générale

La problématique générale de cette thèse concerne l'applicabilité de la reconnaissance d'événements sonores en conditions réelles. En effet, la majorité des recherches ces dernières années ont été effectuées sur des données synthétiques ou des données enregistrées en conditions contrôlées. Ces données sont utiles pour analyser la tâche visée et faire progresser la recherche, mais ne permettent pas de prédire la performance d'un système de reconnaissance d'événements sonores dans un scénario réel où de nombreux problèmes concomitants surgissent.

Pour la reconnaissance d'événements sonores et plus particulièrement pour la détection d'événements sonores, on souhaite idéalement que les données d'apprentissage soient annotées en indiquant non seulement quels événements apparaissent dans un clip audio mais aussi leur position temporelle. Ces annotations appelées *annotations fortes* sont cependant très coûteuses puisqu'il faut un temps important pour les réaliser et l'annotation des frontières temporelles nécessite une expertise particulière. Elles sont souvent remplacées par des *annotations faibles*, qui indiquent quels événements sonores comporte un clip audio mais pas leur position temporelle ni même leur fréquence. En conditions réelles, selon le budget disponible, on se retrouve ainsi le plus souvent avec beaucoup de données non annotées, une certaine quantité de données faiblement annotées et peu voire pas de données fortement annotées.

Par ailleurs, en conditions réelles, les conditions d'apprentissage ne correspondent pas toujours à celles de test. C'est le cas en particulier des systèmes déployés en environnement domestique où la variabilité des conditions de test est grande d'une maison à l'autre et il faut donc s'adapter à ces conditions particulières.

Dans cette thèse, nous faisons une analyse précise des problèmes posés par l'application de la reconnaissance d'événements sonores en conditions réelles et des propositions de formalisation et de résolution de certains de ces problèmes, que nous détaillons ci-dessous.

1.4 Problèmes traités et contributions

1.4.1 DCASE Tâche 4

Au sein de la communauté DCASE, j'ai joué un rôle central dans l'organisation de la Tâche 4 du DCASE Challenge en 2018, 2019 et 2020. Cette tâche vise à évaluer un scénario réaliste de détection d'événements sonores dans un environnement réel : l'environnement domestique. Ce travail m'a amené à définir la spécification initiale d'une tâche complexe et à la questionner et l'améliorer au fur et à mesure des années. Nous avons créé un corpus de détection d'événements sonores appelé *détection d'événements sonores en environnement domestique* (*domestic environment sound event detection*) ([DESED](#))

puis fourni des pistes de résolution de cette tâche en proposant un système de référence amélioré chaque année grâce aux idées des participants. La tâche attire chaque année des dizaines de participants et a permis des avancées scientifiques. Nous avons analysé les résultats en détail pour comprendre ces avancées, mettre en avant les éventuels problèmes qui ne sont toujours pas traités par les systèmes ou parfois redéfinir la tâche pour la rendre plus accessible tout en conservant son réalisme.

1.4.2 Apprentissage semi-supervisé

Dans un scénario réaliste tel que celui ci-dessus, les données d'apprentissage pour la reconnaissance d'événements sonores comportent souvent peu de données (faiblement) annotées et beaucoup de données non annotées. Cela nécessite une méthode d'apprentissage semi-supervisé capable de traiter conjointement les données annotées et non annotées. Dans ce cadre, nous nous sommes intéressés à l'apprentissage semi-supervisé d'une représentation (*embedding*) utilisable aussi bien pour l'étiquetage d'événements sonores que la détection d'événements sonores. Nous avons adapté différentes méthodes d'apprentissage supervisé (basées sur les triplets ou les réseaux prototypes) au cadre semi-supervisé et analysé expérimentalement leur comportement par un ensemble d'expériences en environnement domestique en fonction de la quantité de données non annotées et faiblement annotées. Cette analyse nous a conduit à identifier l'annotation faible des données comme un problème critique indépendamment du cadre semi-supervisé.

1.4.3 Exploitation des annotations faibles

L'annotation faible des données a l'avantage d'être plus rapide que l'annotation forte et réalisable par des non-experts, ce qui rend son coût plus abordable. Cependant, les données faiblement annotées présentent un inconvénient : elles n'indiquent pas explicitement sur quelles parties d'un clip audio l'apprentissage peut s'appuyer pour les différentes classes de sons présentes. Ce problème est un problème récurrent dans le domaine de l'analyse de sons ambiants. De nombreuses méthodes ont été proposées pour le résoudre. Cependant, il est presque toujours accompagné d'une multitude d'autres problèmes (superposition de sons, équilibre des classes, *etc.*) qui brouillent les tentatives d'analyse. Pour rendre cette analyse possible, nous avons conçu un nouveau corpus permettant d'isoler ce problème et analyser en détail l'impact des annotations faibles sur des méthodes d'apprentissage supervisé. Nous avons identifié par une analyse simple que ce problème complexe avait un impact sur les systèmes d'apprentissage mais qui dépend grandement de type d'événements sonores et du système d'agrégation utilisé.

1.4.4 Liste des publications

Les travaux de cette thèse ont donné lieu aux publications suivantes :

- Serizel, R., Turpault, N., Eghbal-Zadeh, H., and Shah, A. P. (2018). Large-scale weakly labeled semi-supervised sound event detection in domestic environments. In

- Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 19–23.
- Turpault, N., Serizel, R., and Vincent, E. (2019). Semi-supervised triplet loss based learning of ambient audio embeddings. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 760–764.
 - Serizel, R. and Turpault, N. (2019). Sound event detection from partially annotated data : Trends and challenges. In *IcETRAN conference*.
 - Turpault, N., Serizel, R., Shah, A. P., and Salamon, J. (2019). Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. In *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 253–257.
 - Serizel, R., Turpault, N., Shah, A. P., and Salamon, J. (2020). Sound event detection in synthetic domestic environments. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 86–90.
 - Turpault, N., Serizel, R., and Vincent, E. (2020). Limitations of weak labels for embedding and tagging. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 131–135.
 - Turpault, N. and Serizel, R. (2020). Training sound event detection on a heterogeneous dataset. In *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 200–204.
 - Turpault, N., Wisdom, S., Erdogan, H., Hershey, J. R., Serizel, R., Fonseca, E., Seetharaman, P., and Salamon, J. (2020). Improving sound event detection in domestic environments using sound separation. In *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 205–209.
 - Wisdom, S., Erdogan, H., Ellis, D. P. W., Serizel, R., Turpault, N., Fonseca, E., Salamon, J., Seetharaman, P., and Hershey, J. R. (2020). What’s all the FUSS about Free Universal Sound Separation data? *arXiv preprint arXiv :2011.00803*.
 - Ferroni, G., Turpault, N., Azcarreta, J., Tuveri, F., Serizel, R., Bilen, C., and Krszulović, S. (2020). Improving sound event detection metrics : Insights from DCASE 2020. *arXiv preprint arXiv :2010.13648*.
 - Turpault, N., Serizel, R., Wisdom, S., Erdogan, H., Hershey, J. R., Fonseca, E., Seetharaman, P., and Salamon, J. (2020). Sound event detection and separation : a benchmark on DESED synthetic soundscapes. *arXiv preprint arXiv :2011.00801*.

Les travaux réalisés dans les papiers de Wisdom et al. et Ferroni et al. ne seront pas discutés en détail dans ce manuscrit mais simplement cités dans la partie en lien avec ces travaux.

1.5 Plan du document

Le chapitre 2 expose l’état de l’art des différents domaines abordés dans la thèse : la reconnaissance d’événements sonores, l’apprentissage semi-supervisé, l’exploitation des

annotations faibles et l'adaptation aux conditions de test.

Le chapitre 3 décrit une partie contributions scientifiques en tant qu'organisateur de la Tâche 4 du DCASE Challenge, qui porte sur la détection d'événements sonores en environnement domestique. Nous décrivons comment nous avons défini cette tâche pour permettre l'avancée des recherches dans ce domaine, comment nous avons créé le jeu de données et nous rapportons et discutons les résultats officiels des systèmes soumis chaque année par les participants de la tâche.

Le chapitre 4 décrit l'autre partie des contributions scientifiques réalisées en tant qu'organisateur de la Tâche 4 du DCASE Challenge. Nous décrivons les systèmes de référence qui ont été développés chaque année ainsi qu'une étude ablative du système de référence de 2020. Nous faisons l'analyse détaillée des systèmes soumis chaque année par les participants grâce à de multiples jeux de données d'évaluation que nous avons créés pour isoler des problèmes scientifiques spécifiques.

Le chapitre 5 porte sur l'apprentissage de représentation semi-supervisé à partir de données non annotées et faiblement annotées. Nous exposons différentes méthodes d'apprentissage de représentation (basées sur les triplets, les réseaux prototypes) et les adaptons à l'apprentissage semi-supervisé. Nous comparons ces méthodes à l'apprentissage d'une représentation issue d'un classifieur. Nous évaluons leur performance pour l'étiquetage d'événements sonores en environnement domestique et analysons les résultats.

Le chapitre 6 étudie en détail l'impact des annotations faibles sur l'apprentissage d'un système d'étiquetage d'événements sonores. Nous définissons un scénario pour isoler le problème des annotations faibles et nous construisons un jeu de données associé. Nous analysons ensuite l'impact des annotations faibles du point de vue de la méthode d'agrégation temporelle des estimations. Nous concluons sur l'impact des annotations faibles non seulement pour l'apprentissage et le test de systèmes mais aussi sur les aspects à prendre en compte lors de la création d'un corpus.

Le chapitre 7 dresse le bilan des différents points étudiés dans cette thèse et propose des pistes de poursuite de ce travail.

2 État de l'art

Dans ce chapitre nous faisons un état de l'existant en analyse de sons ambiants et plus spécifiquement en reconnaissance d'événements sonores en environnement réel.

2.1 Contexte

Un environnement sonore peut être complexe par nature puisqu'il englobe une diversité d'événements sonores qui se superposent. Un événement sonore est un son particulier émis par une source sonore. Une scène sonore est généralement composée d'événements sonores d'intérêt accompagnés de bruit de fond composé :

- d'événements sonores qui ne font pas partie des événements sonores d'intérêt ;
- d'événements sonores dont le niveau sonore est trop faible, il est donc impossible de les distinguer séparément ;
- de bruit de mesure dû au système d'enregistrement.

2.1.1 Un peu d'histoire

Dans les années 90, [Bregman \[1990\]](#) a proposé une explication de la façon dont les humains organisent les sons et arrivent à percevoir et distinguer des événements sonores au sein d'un environnement sonore. Il a décrit les phénomènes psychoacoustiques et fait le lien avec les scènes visuelles qui étaient plus largement étudiées et mieux comprises à l'époque. De multiples travaux sont ensuite apparus proposant des algorithmes pour classifier les scènes sonores automatiquement [[Brown and Cooke, 1994](#), [Ellis, 1996](#)] ou pour reconnaître des événements spécifiques [[Ellis, 1996](#), [Goldhor, 1993](#), [Woodard, 1992](#)]. En 1996, il est déjà proposé d'utiliser des réseaux de neurones artificiels [[Choe et al., 1996](#), [Ellis, 1996](#)] mais les contraintes informatiques (puissance de calcul trop faible) ne permettent pas d'implémenter des modèles assez puissants pour être exploitables. Les approches utilisent généralement des représentations bas niveau du signal sonore et des algorithmes de décision simples permettant de corrélérer les représentations similaires [[Choe et al., 1996](#), [Sampan, 1998](#)].

Dans les années 2000, l'apparition d'un nombre plus important de petits jeux de données annotées focalisées sur la reconnaissance de scènes ou d'événements sonores ouvre la recherche autour de nouvelles applications. Parmi les applications proposées, on trouve principalement des applications de surveillance pour détecter les alarmes, sonneries et sons indicateurs de danger [[Clavel et al., 2005](#), [Ellis, 2001](#)], des applications de reconnaissance des activités domestiques [[Chen et al., 2005](#), [Härmä et al., 2005](#), [Peng et al., 2009](#)],

et des applications liées aux spectacles [Baillie and Jose, 2003, Cai et al., 2003] ou à l'urbanisme [Munich, 2004]. L'ensemble de ces initiatives montrent l'intérêt grandissant pour l'analyse des sons ambiants. Cependant, il est toujours difficile de se faire une idée de la résolution des problèmes de reconnaissance des sons ambiants puisqu'il y a peu de comparaisons entre les différentes approches et chacune est en général évaluée sur son propre jeu de données spécifique. L'apparition du workshop *Classification des Événements, Activités et Relations (Classification of Events, Activities and Relationships) (CLEAR)* en 2006 [Stiefelhagen et al., 2006] et 2007 [Stiefelhagen et al., 2007], qui incluait la détection et la reconnaissance d'événements sonores, a contribué à faire avancer le domaine. Cela a permis d'introduire un cadre pour la détection et la reconnaissance d'événements sonores et d'évaluer plusieurs solutions en utilisant le même protocole d'évaluation (métriques et jeux de données similaires) [Heittola and Klapuri, 2008, Zhuang et al., 2010].

Dans les années 2010, l'amélioration des systèmes informatiques et l'explosion du volume de données disponible sur internet ont permis le développement de nouvelles méthodes et la résolution de problèmes plus complexes. En parallèle, l'apparition des campagnes d'évaluation DCASE [Stowell et al., 2015] a permis de structurer une partie des efforts des chercheurs de la communauté. Les travaux réalisés depuis cette période sont traités plus en détail par la suite dans cet état de l'art.

2.1.2 Les applications

Les travaux sur l'analyse des sons ambiants s'articulent autour de différentes applications qui évoluent au cours du temps. On fait ici une analyse de l'état de l'art actuel de ces applications qui engendrent chacune différents problèmes scientifiques. Cette liste n'est pas exhaustive mais représente une grande partie des problèmes actuels du domaine en terme d'applications ou de méthodes.

Urbanisme Dans cette application, on désire identifier et quantifier les sources de pollution sonore en environnement urbain, notamment les travaux et les diverses formes de trafic routier. La lutte contre cette pollution passe ensuite par des décisions politiques qui dépassent le cadre de l'application proprement dite. Cette application soulève tout d'abord des problèmes liés à la collecte de données, qui peut se faire avec les smartphones des habitants [Ruge et al., 2013, Ventura et al., 2018] ou avec des capteurs déployés dans la ville, comme expérimenté à New York [Bello et al., 2019] ou en France à Lorient [Ardouin et al., 2018]. Des jeux de données de recherche ont aussi été créés en utilisant des données collectées sur internet [Mesaros et al., 2019, Salamon et al., 2014]. L'avantage de placer des capteurs dans la ville est d'avoir un contrôle sur le matériel et la position des microphones [Ardouin et al., 2018, Bello et al., 2019]. Se pose alors le problème de concevoir des capteurs audio connectés avec un faible coût et une faible consommation d'énergie pour répondre aux contraintes écologiques, financières et d'autonomie [Ardouin et al., 2018]. Des informations supplémentaires à l'audio peuvent être utilisées, comme la position spatio-temporelle des sons enregistrés [Bello et al., 2019]. L'estimation de l'intensité de chaque source lorsque plusieurs sources sont présentes de façon simultanée et leur traduction sous forme d'un score de pollution sonore peuvent s'avérer utiles à la

prise de décision finale [Kavalerov et al., 2019]. Quel qu'en soit le moyen, la collecte de données est donc toujours effectuée en ville et en extérieur, où la majorité des espaces sont publics. En plus des contraintes pratiques, les contraintes de respect de la vie privée sont à prendre en compte, par exemple il n'est pas possible d'enregistrer la voix d'une personne passant à côté d'un microphone sans son consentement. L'utilisation de l'audio dans les applications en lien avec la ville connectée ou les voitures autonomes est plus difficile car ces applications nécessitent généralement une meilleure précision temporelle. Mesaros et al. [2019] se sont intéressés non seulement à la classification des sons urbains mais aussi à la détection précise de ceux-ci au sein d'un enregistrement audio.

Maintenance prédictive La majorité des usines utilisent plusieurs machines qui émettent des bruits caractéristiques. Il est souvent possible de détecter un fonctionnement anormal ou l'arrivée d'une panne par les bruits anormaux que les machines produisent. On cherche ainsi à détecter les anomalies en séparant les clips audio normaux des clips audio contenant une anomalie [Ono et al., 2013]. On peut aller plus loin en cherchant à classifier le type d'anomalie [Henze et al., 2019]. La difficulté principale est liée à la rareté des anomalies et à l'ignorance de toutes les anomalies possibles, ce qui empêche la collecte ou la simulation d'un gros jeu de données contenant toutes les anomalies possibles. Pour résoudre ce problème, on peut utiliser un jeu de données contenant seulement des données sans anomalies afin d'apprendre le comportement « normal » de la machine et essayer de détecter les comportements anormaux [Koizumi et al., 2020]. Une deuxième solution est d'utiliser des données non annotées ayant une probabilité importante de contenir des anomalies et d'apprendre la détection d'anomalie à partir de ces données [Yang et al., 2018]. Ce second cas est plus complexe et revient généralement à apprendre le comportement « normal » à partir de données non annotées. La recherche autour de cette application est le plus souvent effectuée en collaboration avec des partenaires industriels qui collectent les données mais ne les distribuent pas publiquement. Récemment, deux corpus ont été développés afin d'analyser et de proposer des solutions à ce problème, bien que ceux-ci ne soient pas réels [Koizumi et al., 2019, Purohit et al., 2019].

Biodiversité et nature Les animaux utilisent très largement le son pour communiquer. L'analyse de sons ambiants a donc été appliquée aux problèmes de détection dans la nature en particulier en lien avec la biodiversité. Une tâche importante est celle de la compréhension des migrations animalières. Il s'agit par exemple de détecter la présence d'oiseaux dans un enregistrement audio [Stowell et al., 2019] ou bien d'identifier l'instant d'émission et l'espèce de l'oiseau émettant ces sons [Løstang et al., 2018]. L'analyse de sons ambiants peut aussi s'appliquer à l'identification de différentes espèces marines pour permettre leur suivi automatique de manière non intrusive [Ferrari et al., 2020]. Un problème récurrent concerne la distance importante entre les micros et l'événement que l'on souhaite reconnaître, qui rend la tâche difficile. Une autre tâche possible concernant les animaux bien qu'ils ne soient pas dans un environnement naturel consiste à reconnaître les problèmes respiratoires dans les exploitations agricoles [Carpentier et al., 2019, Chung et al., 2013].

Applications domestiques L'analyse des sons ambiants en milieu domestique peut être bénéfique pour aider les personnes sourdes ou malentendantes, surveiller la sécurité des personnes et de l'habitat, et rendre la maison plus intelligente. Les applications visées dans ce cas sont le plus souvent liées à la domotique [Debes et al., 2016], l'aide à la personne [Navarro et al., 2018], ou bien la sécurité [Radhakrishnan et al., 2005]. L'intérêt de reconnaître les événements sonores pour une personne sourde ou malentendante est de pouvoir bénéficier de microphones se déplaçant plutôt que d'installer des capteurs sur chaque appareil devant renvoyer une réponse différente d'un signal sonore. Lorsque l'on cherche à faire de l'aide à la personne afin de surveiller qu'il y a une activité par exemple, le problème peut être défini de façon à reconnaître les activités présentes dans une pièce sans se soucier précisément des événements sonores [Dekkers et al., 2017]. Lorsqu'on décide de faire de la sécurité (par exemple, la détection d'effraction), le problème est d'identifier des événements rares qui apparaissent en situation d'urgence [Kim et al., 2020] alors que, pour des applications d'aide à la personne, on essaye de reconnaître des événements plus familiers (par exemple, une alarme) [Mesaros et al., 2018a]. Les applications domestiques incluent aussi les applications qui permettent de reconnaître des événements sonores au sein d'un bureau, qui sont le plus souvent utiles dans le cadre de la domotique pour adapter l'environnement en fonction des événements présents ou de la surveillance. Elles soulèvent aussi des problématiques concernant les conditions d'enregistrement en intérieur et les caractéristiques spécifiques des sons domestiques qui sont étudiées [Vafeiadis et al., 2020].

Détection de contexte La détection de contexte est utile lorsque l'on cherche une granularité de classes d'événements plus grande. On peut chercher à répondre aux questions « Que se passe-t-il autour de moi ? Que suis-je en train de faire ? » [Dekkers et al., 2017] ou bien « Dans quel type de lieu suis-je ? » [Eronen et al., 2006]. L'intérêt de reconnaître l'environnement sonore est d'adapter d'autres applications (des algorithmes) à leur environnement [Mesaros et al., 2017b]. Une application consiste à avoir un microphone (dans un smartphone par exemple) que l'on déplace avec soi dans plusieurs environnements sonores. Un exemple concret est celui d'un téléphone ayant une sonnerie plus élevée lorsqu'on est dans un endroit bruyant (au sein d'une foule) et plus faible dans un endroit silencieux (dans un bureau).

Analyse polyvalente de sons ambiants Les applications répondant à la question « Que se passe-t-il autour de moi ? » peuvent avoir une granularité très différente. On peut chercher à reconnaître ce qui se passe de manière générale avec une granularité grossière (bruyant/calme, intérieur/extérieur). On peut aussi souhaiter une granularité plus fine et s'intéresser à certains événements plus particulièrement [Fonseca et al., 2019b]. Dans ce dernier cas, la majorité des applications deviennent spécifiques, comme celles présentées dans les paragraphes précédents (urbanisme, maintenance prédictive, biodiversité et nature, applications domestiques). Reconnaître tous les événements sonores de manière fine indépendamment de l'application est une tâche difficile et d'utilité pratique discutable. En effet, pour chaque application, on peut identifier les classes d'événements pertinentes

et, si l'on souhaite reconnaître des événements dans un nombre important de situations, il est possible d'utiliser le contexte (intérieur/extérieur par exemple) pour spécialiser le système. L'apprentissage d'un système générique couvrant un grand nombre de classes très diverses présente néanmoins un intérêt particulier, dans la mesure où la représentation ainsi apprise peut être utilisée comme point de départ pour l'apprentissage de modèles plus spécifiques [Gemmeke et al., 2017].

2.1.3 Les tâches méthodologiques

L'analyse des sons ambiants comporte plusieurs tâches. Elles ont des liens entre elles, mais leur formulation est souvent spécifique.

La classification de scènes sonores La tâche de classification de scènes sonores représentée dans la Figure 2.1 consiste à catégoriser un clip audio par une classe relative à l'ensemble de la scène sonore [Brown and Cooke, 1994]. Les classes considérées sont souvent associées à un lieu (gare, restaurant, parc, *etc.*) ou une activité (manger, faire du sport, *etc.*), plutôt qu'à des éléments sonores spécifiques. En effet, ces concepts sont faciles à annoter car ils sont connus de tous, décrivent la scène globale, ne changent pas fréquemment au cours de l'enregistrement, et ne nécessitent qu'un choix unique. La difficulté de cette tâche se trouve notamment dans la confusion possible entre des scènes sonores proches, par exemple une station de métro et une gare ferroviaire, et dans la diversité des sons appartenant à une même classe, par exemple le bruit du hall à bagages et le bruit à proximité des pistes dans un aéroport. Cette tâche est souvent considérée comme la tâche de référence dans l'analyse des sons ambiants en raison de la possibilité de formuler et de résoudre des problèmes simples à partir de celle-ci. Par exemple, on peut définir si la scène est en intérieur ou en extérieur.

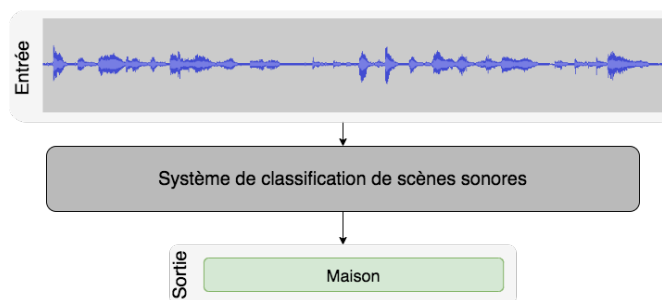


FIGURE 2.1 – Classification de scènes sonores.

La reconnaissance d'événements sonores La reconnaissance d'événements sonores consiste à déterminer quels événements composent un enregistrement sonore. Ce terme regroupe plusieurs tâches présentées dans les paragraphes suivants : l'étiquetage, la détection et la localisation des événements sonores. Pour l'ensemble de ces tâches, la répartition des événements en classes d'intérêt représente une partie cruciale de la conception

de solutions à des applications concrètes. Dans ce manuscrit, on distingue les tâches de reconnaissance d'événements sonores de la tâche de sous-titrage audio par le critère suivant : on utilise des ontologies lors de la reconnaissance d'événements sonores et on utilise le langage naturel lors du sous-titrage audio. Ces tâches peuvent en réalité se regrouper, mais la faible avancée des recherches en sous-titrage audio et l'utilisation exclusive de classes d'intérêt dans les travaux de cette thèse nous font faire cette distinction.

L'étiquetage d'événements sonores La tâche d'étiquetage représentée dans la Figure 2.2 consiste à déterminer quels événements sonores sont présents dans un clip audio. Elle est considérée comme plus fine que la classification de scènes sonores. La granularité des classes d'événements dépend de l'application visée. Par exemple on peut vouloir reconnaître un « chien » ou bien différencier un « grognement » d'un « aboiement ». Dans cette tâche, on ne s'intéresse pas à la dimension temporelle ou à la répétition des événements : on cherche quels événements sont apparus au moins une fois dans le clip. La longueur des clips est souvent supposée constante dans la littérature, mais ce n'est pas une nécessité. Cette tâche est considérée comme suffisante pour bon nombre d'applications reposant sur la détermination des événements ayant eu lieu dans un intervalle de temps donné. L'effort d'annotation de cette tâche dépend de la granularité des classes d'événements sonores. Dans les cas simples de la vie courante, un humain peut définir quelles classes d'événements sont apparues dans un fichier audio presque en temps réel.

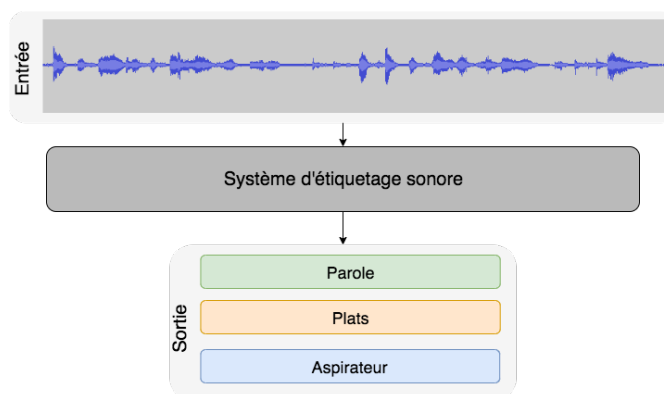


FIGURE 2.2 – Étiquetage d'événements sonores.

La détection d'événements sonores La tâche de détection représentée dans la Figure 2.3 consiste à déterminer quels événements composent un enregistrement sonore, ainsi que les instants auxquels ces événements se produisent. Cette tâche est considérée comme une application plus fine de l'étiquetage d'événements sonores. Si dans un clip audio de 10 s un chien aboie 10 fois, on cherche non seulement à savoir qu'un chien a aboyé, mais à déterminer combien de fois et à quels moments du clip audio cela s'est produit. La définition de cette tâche est aussi soumise à l'application et des différences peuvent se poser dans la définition du problème. Dans certains travaux, la détection d'événements

sonores est considérée comme de l'étiquetage d'événements sonores avec des durées de clips courtes. Le principe ici est d'avoir une précision plus fine qui se rapprocherait de la détection en temps réel. L'effort d'annotation est conditionné par le temps que prend un humain pour identifier les instants de début et de fin des événements, y compris lorsque plusieurs événements se superposent. Une difficulté supplémentaire vient de l'aspect parfois subjectif du concept d'instant de début et de fin, par exemple dans un environnement réverbérant.

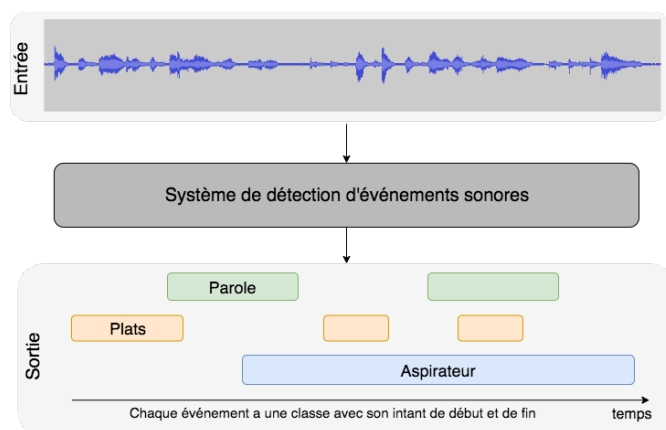


FIGURE 2.3 – Détection d'événements sonores.

La détection et localisation d'événements sonores La tâche de détection et de localisation d'événements sonores consiste à déterminer la localisation temporelle et spatiale des événements sonores qui composent un clip audio. Cette tâche peut être considérée comme un cas particulier de la tâche de détection d'événement sonore, visant à déterminer non seulement les instants de début et de fin de chaque événement mais aussi sa localisation spatiale. Cette tâche est utile dans les cas où la localisation spatiale est importante. Par exemple, une voiture autonome bénéficierait de savoir non seulement qu'un camion de pompier est en approche, mais aussi de quel côté (particulièrement à un croisement). Par rapport à la simple détection d'événements sonores, l'effort d'annotation est accru par la faible précision d'estimation de la direction d'arrivée du signal sonore par un humain à partir d'un enregistrement non-binaural. L'annotation précise de la localisation requiert des enregistrements dans un environnement contrôlé où les positions des sources sonores et des microphones sont connues.

Le sous-titrage audio La tâche de sous-titrage audio consiste à décrire les événements ou scènes sonores par un texte en langage naturel, plutôt que des classes d'événements fixées dans une ontologie. Utiliser le langage naturel pour décrire les scènes et événements sonores permet de rajouter des détails difficiles à ajouter lors de l'utilisation d'ontologies comme par exemple : « Bruit d'aspiration sourd avec un sifflement aigu semblable à un aspirateur obstrué ». Cela engendre une subjectivité plus importante de la part de

l'annotateur. Chaque annotateur peut avoir une description différente d'un même son, ce qui laisse apparaître leur perception (cela peut être considéré comme un avantage) mais aussi un certain nombre d'ambiguïtés. L'annotation en langage naturel peut aussi provoquer un biais engendré par les consignes, la culture ou la personnalité de l'annotateur. C'est pour réduire ce problème que les autres tâches utilisent des vocabulaires contrôlés (ontologies). Le traitement du langage naturel devient alors une part importante de ce problème à résoudre.

2.1.4 Le challenge DCASE

Le challenge DCASE est apparu en 2013 avec une tâche de classification de scènes sonores et une tâche de détection d'événements sonores. Depuis la deuxième édition en 2016, il est accompagné d'un workshop et se déroule annuellement. Le Tableau 2.1 résume les différentes tâches du challenge DCASE et les applications associées. Bien que ne présentant pas une liste exhaustive de problèmes et d'applications, ces tâches reflètent bien les centres d'intérêt de la communauté ainsi que l'évolution des différents problèmes à résoudre au cours des années. Les tâches de classification de scènes sonores, d'étiquetage audio et de détection d'événements sonores sont représentées chaque année depuis 2013, avec une attention particulière pour la détection d'événements sonores qui peut être considérée comme la plus complexe d'entre elles. Les tâches de localisation et de sous-titrage audio ne sont traitées que depuis 2019 et 2020 respectivement. Ces différentes tâches donnent généralement lieu à la création d'un jeu de données, permettant de nouvelles recherches autour des problèmes définis dans le challenge. Certaines tâches sont motivées par des problématiques d'ordre méthodologique ; c'est le cas par exemple de celles concernant les enregistrements dans des bureaux (la Tâche 2 en 2013 et 2016 et la Tâche 3 en 2019 et 2020). D'autres tâches sont motivées par les applications, par exemple la Tâche 3 en 2018 s'intéressant aux phénomènes migratoires des oiseaux.

Tâche DCASE	2013	2016	2017	2018	2019	2020
Tâche 1	Général					
Tâche 2	Bureau	Bureau	Alertes	Général	Industrie (anomalies)	
Tâche 3		Domestique & aire résidentielle	Urbain	Oiseaux	Bureau	
Tâche 4		Domestique	Véhicules	Domestique		
Tâche 5				Domestique	Urbain	
Tâche 6						Général

TABLEAU 2.1 – Tâches et applications du challenge DCASE. Les couleurs correspondent à la tâche méthodologique et le texte au domaine d'application. Le vert correspond à la classification de scènes sonores, le orange à l'étiquetage audio, le bleu ciel à la détection d'événements sonores, le bleu foncé à la détection et la localisation d'événements sonores et le rouge au sous-titrage audio.

2.2 Analyse de sons ambiants

2.2.1 Système d'analyse de sons ambiants

Les systèmes d'analyse de sons ambiants reposent sur une approche générale commune à l'ensemble des tâches ci-dessus. À partir d'un jeu de données d'apprentissage représentatif de la tâche visée, une méthode d'apprentissage automatique est employée afin d'obtenir un système capable de résoudre cette tâche pour d'autres données similaires appelées données de test ou d'évaluation. L'étape finale du système est un classifieur, qui attribue les sons à des classes prédéfinies. En raison du fossé sémantique entre le signal audio et les classes d'intérêt, ce classifieur n'est pas appliqué directement à la forme d'onde du signal mais à une représentation qui peut être fixée par des opérations de traitement du signal, pré-apprise indépendamment du classifieur, ou apprise conjointement au classifieur. Alors que la représentation des données audio peut être générale et partagée entre plusieurs tâches et applications, le classifieur est spécifique à la tâche et à l'application.

Dans la suite du manuscrit, on fait une différence entre les représentations de données dites de « bas niveau » et les représentations de « haut niveau ». Les représentations de bas niveau sont directement issues de la forme d'onde et sont calculées par traitement du signal ou bien apprises. Elles sont généralement en deux dimensions (représentation temps-fréquence) et facilitent l'analyse de l'information audio par rapport à la forme d'onde à une dimension [Serizel et al., 2017]. Les représentations de haut niveau, comme la représentation Audioset [Hershey et al., 2017], sont obtenues à partir des représentations de bas niveau. Elles sont généralement de taille plus petite et ont pour but de grouper et discriminer les signaux de façon utile à un ensemble d'applications.

La Figure 2.4 illustre l'architecture générale d'un système d'analyse de sons ambiants. Nous illustrons chaque étape par une visualisation avec une dimension donnée, purement à titre d'exemple. L'ensemble des étapes sont ici illustrées séparément mais celles-ci ne sont pas toujours explicites. L'ensemble du système est parfois appris par descente de gradient « de bout-en-bout » depuis les classes d'intérêt jusqu'à la forme d'onde sans définir explicitement quelles parties correspondent au calcul de la représentation de bas niveau, au calcul de la représentation de haut niveau ou au classifieur.

2.2.2 Représentation de bas niveau

2.2.2.1 Représentation de bas niveau issue du traitement du signal

La représentation de bas niveau du signal audio a fait l'objet de recherches poussées en traitement du signal. Serizel et al. [2017] décrivent les différentes représentations possibles. Nous n'en ferons pas la liste exhaustive dans cette thèse, et nous focalisons sur les représentations de bas niveau les plus importantes. La majorité des systèmes de l'état de l'art utilisent une représentation en deux dimensions représentant le temps et la fréquence appelée spectrogramme. La *transformée de Fourier à court terme* (*short time Fourier transform*) (STFT) est calculée à partir de la transformée de Fourier discrète appliquée sur des fenêtres temporelles courtes, en utilisant la *transformée de Fourier rapide* (*fast Fourier transform*) (FFT). En règle générale, on ne s'arrête pas à cette simple

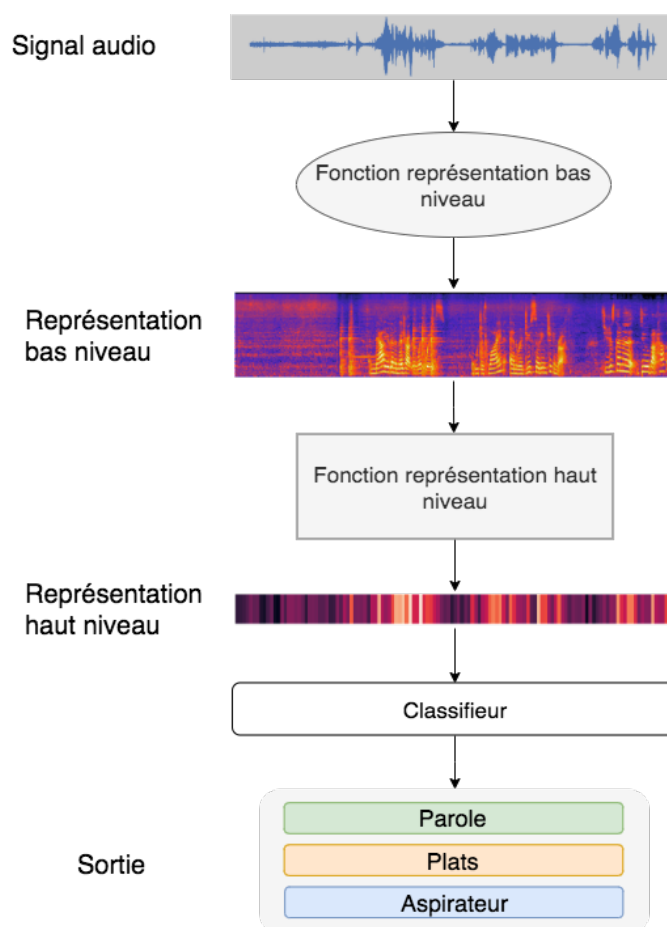


FIGURE 2.4 – Étapes d'un système d'analyse de sons ambiants illustrées par un exemple.

transformation et on cherche à avoir une représentation de bas niveau plus explicite et motivée par la perception sonore humaine. Pour ce faire, une représentation des fréquences en échelle non-linéaire (Mel, Bark, ou ERB) est utilisée. Ceci permet de représenter le son d'une manière plus proche de celle dont un humain l'entendrait puisqu'on n'a pas la même résolution fréquentielle ni sensibilité à l'amplitude partout sur le spectre auditif. Parmi celles-ci, l'une des représentations les plus utilisées dans l'analyse des sons ambiants est la représentation en échelle Mel. L'échelle Mel correspond à une approximation de la sensation psychologique de hauteur d'un son pur (une sinusoïde pure). En pratique, on extrait un spectrogramme en échelle Mel en appliquant un banc de filtres Mel au spectrogramme d'amplitude ou de puissance issu de la *STFT*. La Figure 2.5 représente 128 filtres Mel appliqués à un signal audio de fréquence d'échantillonnage de 44,1 kHz. Comme on peut le constater sur cette figure, les filtres sont beaucoup plus denses dans les basses fréquences que dans les hautes fréquences. Afin de réduire le contraste entre les valeurs d'amplitude ou de puissance, on considère souvent le logarithme de ce spectrogramme appelé spectrogramme log-Mel, ou bien les *coefficients cepstraux en échelle Mel* (Mel

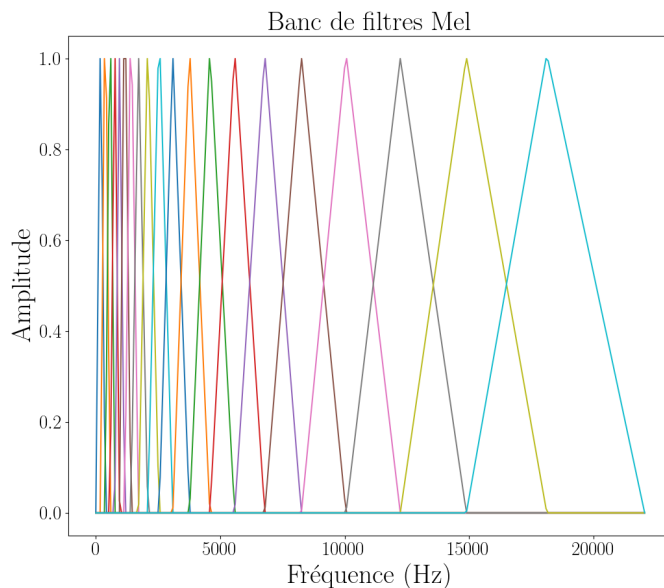


FIGURE 2.5 – Banc de filtres Mel permettant le calcul d’un spectrogramme en échelle Mel.

frequency cepstral coefficients) (MFCC). Ces derniers ont été très utilisés en parole [Davis and Mermelstein, 1980] et dans une moindre mesure en analyse de sons ambiants [Chu et al., 2009]; ils peuvent être avantageux pour certaines classes d’événements sonores mais malheureusement pas pour d’autres.

Une représentation alternative est basée sur les filtres *gammatone* composés d’une sinusoïde modulée en amplitude par une enveloppe ayant la forme d’une fonction de distribution gamma normalisée. Perez-Castanos et al. [2020] utilisent cette représentation pour la détection d’anomalies. Valero and Alias [2012] et Agrawal et al. [2017] utilisent les *coefficients cepstraux gammatone* (*gammatone cepstral coefficients*) (GTCC) pour l’analyse de scène sonore et la détection d’événements sonores. Plusieurs études montrent l’intérêt d’utiliser les GTCC plutôt que les MFCC pour l’analyse de sons ambiants [Agrawal et al., 2017, Valero and Alias, 2012]. Cependant McLoughlin et al. [2015], qui montrent les bonnes performances des GTCC pour l’étiquetage d’événements sonores, montrent aussi qu’ils ne sont pas robustes à un bruit de fond important. Les GTCC ne semblent donc pas appropriés aux scénarios réels, qui impliquent souvent un bruit de fond important. D’autres représentations sont utilisées mais se sont régulièrement montrées moins efficaces pour l’analyse de sons ambiants. Il y a les représentations simples telles que le taux de croisement avec zéro (*zero crossing rate*) [Piczak, 2015] ou bien plus complexes comme celles utilisant la transformée à Q-constant [Henaff et al., 2011], les ondelettes [Qian et al., 2017] ou la transformée de dispersion [Salamon and Bello, 2015a].

2.2.2.2 Choix d'une représentation de bas niveau

Mesaros et al. [2018b] listent les différentes représentations utilisées par les participants de la Tâche 1 du challenge DCASE 2017, en majorité basées sur l'échelle Mel. Geiger et al. [2013] comparent quant à eux les MFCC avec plus de 6000 descripteurs issus de la combinaison de 57 descripteurs de plus bas niveau comprenant le spectrogramme en échelle Mel. Ils montrent que la combinaison des descripteurs est meilleure que les MFCC et qu'un des descripteurs les plus importants est le spectrogramme en échelle Mel. Il faut cependant noter que la combinaison de descripteurs augmente la complexité calculatoire. Un dernier exemple concerne le travail de Wu and Lee [2019], qui analysent la représentation par spectrogramme en échelle Mel afin de savoir si elle est adaptée à la classification de scènes sonores et proposent d'utiliser des filtres gaussiens sur le spectrogramme afin de l'améliorer.

Lorsque l'on a accès à un nombre important de descripteurs, une idée est de faire du filtrage afin d'obtenir une représentation de bas niveau adaptée à notre application d'intérêt. Zhuang et al. [2010] ont comparé les différentes approches de banc de filtres, de MFCC et le filtrage de descripteurs basé sur une approche de boosting (*Adaboost*) afin d'obtenir une représentation adaptée à la tâche visée. Leur but initial était de vérifier que les représentations acoustiques utilisées en parole (log-Mel, MFCC) étaient adaptées à la détection d'événements sonores et de n'utiliser que les descripteurs importants. Cette approche peut se rapprocher des représentations obtenues par apprentissage.

2.2.2.3 Représentation de bas niveau apprise

Ces dernières années avec l'amélioration de la puissance de calcul, de la mémoire et du nombre de données, une alternative aux représentations de bas niveau issues de la recherche en traitement du signal a été proposée. C'est une représentation directement apprise à partir de la forme d'onde du signal sonore par un réseau de neurones cherchant à résoudre l'application d'intérêt ou bien une application dite « factice » permettant l'apprentissage de cette représentation. Lors de l'apprentissage d'une représentation, il est difficile de faire la distinction entre une représentation de bas niveau et une représentation de haut niveau. Dans cette partie, nous faisons le choix de présenter les travaux permettant de comparer les représentations de bas niveau issues de la recherche en traitement du signal et celles apprises. Kim et al. [2019] ont proposé un réseau de neurones convolutionnel opérant sur la forme d'onde et le comparent avec un réseau similaire opérant sur le spectrogramme en échelle Mel afin de comparer les représentations de bas niveau. Ils évaluent ce système pour trois applications audio dont la classification de scènes sonores. Pour des fenêtres et convolutions petites, le modèle utilisant la forme d'onde est meilleur que celui utilisant le spectrogramme. Cependant, même si ce modèle a des performances proches de l'état de l'art pour l'étiquetage de musique ou bien pour la détection de mots-clés, pour la classification de scènes sonores, les performances sont bien en dessous de l'état de l'art. Wang et al. [2019a] font une expérience similaire pour l'étiquetage d'événements sonores, et montrent que le système utilisant les spectrogrammes log-Mel est plus performant que le système appris sur les formes d'onde directement.

Dans un de nos travaux [Serizel and Turpault, 2019], nous décrivons les données d’entrée utilisées par les participants à la Tâche 4 du challenge DCASE 2019 sur la détection d’événements sonores en environnement domestique. Nous constatons que l’utilisation de représentations de bas niveau en entrée des réseaux de neurones est majoritaire et dominée par le spectrogramme log-Mel. D’autres méthodes continuent d’être étudiées mais ne parviennent toujours pas à surpasser les méthodes utilisant les spectrogrammes.

2.2.3 Représentation de haut niveau

Les représentations de haut niveau sont apprises et le plus souvent spécifiques à une famille d’applications. Comme indiqué précédemment, lorsque le système est appris de bout en bout, il est difficile de faire la distinction entre la représentation de bas niveau et celle de haut niveau. Sauf indication contraire, les travaux présentés ci-dessous adoptent l’approche la plus courante où une représentation de bas niveau issue du traitement du signal est utilisée en entrée, de sorte que la seule représentation apprise est celle de haut niveau. L’apprentissage de représentation peut être guidé par une application d’intérêt, par une application « factice » ou par des mesures dans l’espace de représentation. L’apprentissage guidé par une application d’intérêt est appelé apprentissage supervisé de représentation et, dans ce cas, il est fort probable que l’application guidant l’apprentissage de la représentation soit différente de celle visée lors de son utilisation. L’apprentissage guidé par une application factice est appelée apprentissage « auto-supervisé » de représentation. Contrairement à l’apprentissage supervisé, l’application « factice » est conçue pour permettre l’apprentissage d’une représentation pertinente mais elle ne correspond pas à une véritable application concrète. Enfin, l’apprentissage comparatif utilise un coût d’apprentissage directement basé sur l’espace de représentation utilisé. L’apprentissage de représentation n’est pas spécifique à l’audio et est très populaire pour l’analyse de textes ou d’images, c’est pourquoi des travaux issus de ces domaines seront cités dans les paragraphes suivants.

Apprentissage supervisé de représentation de haut niveau L’apprentissage supervisé de représentation vise à apprendre une représentation pour une application donnée et à la réutiliser pour une application similaire. L’exemple le plus connu concerne les représentations de haut niveau apprises sur les jeux de données Youtube-8M et Audioset pour l’étiquetage d’événements sonores avec un grand nombre de classes (512) par Gemmeke et al. [2017] et Hershey et al. [2017]. Ces deux représentations sont souvent utilisées pour des applications similaires mais avec un nombre plus faible de classes. Dans la même idée, Chen et al. [2020] proposent une représentation de haut niveau basé sur le jeu de données VGGSound. L’ensemble de ces approches utilisent des jeux de données très importants issus de vidéos Youtube. Bien que l’application d’étiquetage en 512 classes ait un intérêt pratique discutable à l’heure actuelle en raison de sa complexité et de l’absence de besoins réels correspondants, les jeux de données sont annotés afin de résoudre cette application. C’est pourquoi nous les catégorisons parmi les approches supervisées, contrairement aux approches auto-supervisées qui n’utilisent pas les annotations des données.

Apprentissage auto-supervisé de représentation de haut niveau Les représentations les plus connues en apprentissage auto-supervisé concernent les données textuelles avec des modèles tels que word2vec [Mikolov et al., 2013] ou BERT [Devlin et al., 2019] appris grâce à des applications factices de prédiction de mots cachés dans les phrases. En audio, un des modèles les plus populaires utilisant cette méthode appelé « wavenet » est utilisé pour générer de la parole à partir de texte [van den Oord et al., 2016]. Ce modèle est appris en prédisant la fenêtre audio suivante à partir de l'audio existante. De nouvelles méthodes sont récemment apparues basées sur une application factice de reconstruction du signal d'entrée par un auto-encodeur : l'encodeur apprend une représentation de haut niveau et le décodeur l'utilise pour reconstruire le signal d'origine. Kingma and Welling [2014] ont proposé d'apprendre une représentation de haut niveau grâce à un modèle génératif appelé auto-encodeur variationnel (*variational autoencoder*, VAE). Plus récemment Saeed et al. [2020] ont proposé une approche par cohérence, qui consiste à apprendre plusieurs réseaux de neurones similaires avec des entrées un peu différentes mais dont on cherche à retrouver une représentation de haut niveau similaire, et ils utilisent cette approche pour de nombreuses tâches. Cette approche ressemble aux approches siamoises connues de longue date [Bromley et al., 1993], à la différence que les modèles sont appris de façon différente, la comparaison n'est pas simplement faite à la sortie mais à différents niveaux du modèle et, dans le cas de Saeed et al. [2020], la représentation de haut niveau n'est pas la sortie du modèle.

Apprentissage comparatif de représentation de haut niveau L'apprentissage comparatif de représentation de haut niveau peut se faire de façon supervisée ou non-supervisée. Schroff et al. [2015] ont proposé d'apprendre une représentation de haut niveau de façon supervisée grâce à l'apprentissage par triplets. À partir d'un exemple de référence tiré aléatoirement et accompagné d'un exemple positif et d'un exemple négatif, un coût est formulé en fonction des distances entre ces trois exemples dans l'espace de représentation afin de forcer la représentation à rapprocher l'exemple positif et à éloigner l'exemple négatif de l'exemple de référence. Pour aller plus loin, Jati et al. [2019] apprennent des représentations de haut niveau en utilisant un coût par triplets ou un coût par quadruplets. La particularité de ce travail est de prendre en compte la hiérarchie des classes dans l'ontologie afin d'apprendre des représentations plus riches. Lorsque l'on essaye d'apprendre une représentation de façon supervisée comme dans ces travaux, une énorme quantité de données annotées est souvent utilisée. Des approches moins coûteuses en données sont tout de même possibles. Pons et al. [2019] proposent d'apprendre une représentation de haut niveau en utilisant une architecture de réseau appelée « prototype » qui transforme la distance dans l'espace de représentation (difficile à appréhender) en coût de classification. Des approches non-supervisées ont aussi été développées. Jansen et al. [2017] utilisent l'apprentissage par triplets mais ils définissent l'exemple positif par augmentation de données appliquée à l'exemple de référence et choisissent l'exemple négatif selon sa distance dans l'espace de représentation. Salamon and Bello [2015b] ont développé une méthode spécifique à l'étiquetage d'événements sonores et utilisent l'algorithme des k-moyennes sphérique pour apprendre un dictionnaire de prototypes (*codebook*) et effec-

tuer une classification simple des données par une forêt aléatoire (*random forest*). Bisot et al. [2017b] proposent quant à eux d'utiliser la *factorisation matricielle non-négative* (*non-negative matrix factorization*) (NMF) de façon supervisée ou non. Enfin, Cramer et al. [2019] ont utilisé la cohérence entre la vidéo et le son de vidéos Youtube pour apprendre leur représentation de haut niveau.

2.2.4 Classifieurs

Avant l'apparition des *réseau de neurones profond* (*deep neural network*) (DNN), les classifieurs utilisés dans le cadre de l'analyse de sons ambiants étaient principalement basés sur le *modèle de mélange de gaussiennes* (*Gaussian mixture model*) (GMM), le *modèle de Markov caché* (*hidden Markov model*) (HMM) ou la NMF. Zhuang et al. [2010] utilisent un classifieur complexe basé sur des méthodes de HMM, GMM et *machine à vecteurs de support* (*support vector machine*) (SVM) et sur un réseau de neurones à une couche pour discriminer les différentes fenêtres de contexte temporel. Gemmeke et al. [2013] utilisent quant à eux une NMF suivi d'un HMM. Enfin, Heittola et al. [2013] utilisent un modèle à base de GMM et HMM.

Ces méthodes précèdent les classifieurs par DNN qui se sont montrés plus performants ces dernières années. À titre d'exemple, en 2016, les participants du challenge DCASE utilisaient encore majoritairement des classifieurs basés sur la NMF, les GMM ou les SVM [Mesaros et al., 2018a] et ceux-ci étaient compétitifs par rapport aux DNN puisque les gagnants des différentes tâches n'ont pas tous utilisé des DNN. En 2017, la majorité des participants se sont mis à utiliser des DNN [Mesaros et al., 2017a]. Parmi les classifieurs par DNN, on ne cite ici que les plus importants. Les méthodes propres à nos applications seront détaillées dans les prochains chapitres.

Les classifieurs par DNN pour l'analyse de sons ambiants sont souvent inspirés des classifieurs proposés dans d'autres domaines comme l'image ou le texte. Le *perceptron multicouche* (*multilayer perceptron*) (MLP) est utilisé pour illustrer la théorie, mais il est peu employé en pratique en raison de sa complexité calculatoire élevée [Lu et al., 2017b]. Des couches totalement connectées semblables à celles des MLP sont cependant encore utilisées dans les dernières couches des architectures présentées ci-dessous. Le *réseau de neurones convolutionnel* (*convolutional neural network*) (CNN) [LeCun et al., 1998] est devenu un modèle très utilisé après qu'AlexNet a obtenu des performances impressionnantes [Krizhevsky et al., 2017] en classification d'images. Les CNN ont aussi été appliqués à des grands corpus audio et se sont montrés très performants. Hershey et al. [2017] ont ainsi comparé les architectures de CNN populaires sur le corpus Audioset. Dans les architectures de réseaux incluant un CNN, la sortie du CNN peut être ou non considérée comme la représentation de haut niveau. Étant donné que les données sonores sont des séries temporelles, le *réseau de neurones récurrent* (*recurrent neural network*) (RNN) est un modèle qui a aussi été largement utilisé. Un RNN garde en mémoire les instants temporels passés dans une séquence en mettant à jour cette mémoire à chaque instant. Le premier réseau récurrent qui s'est fait connaître est le *long-short term memory* (LSTM) [Hochreiter and Schmidhuber, 1997] qui permet de contrôler le passage de l'information au cours du temps grâce à un ensemble de « portes » (entrée, sortie, oubli).

2.2.5 Problèmes méthodologiques

Conditions d'enregistrement En pratique, le développement d'un système d'analyse de sons ambiants soulève plusieurs problèmes. L'un de ces problèmes concerne la diversité des microphones disponibles sur le marché qui ont chacun leurs spécificités. Par exemple, les smartphones ont des microphones différents et qui peuvent changer régulièrement. Mesaros et al. [2018c] ont analysé ce problème dans le cas où les données d'apprentissage ont été acquises avec un microphone donné mais on souhaite classifier des scènes sonores enregistrées avec un microphone différent. Ces différences de conditions d'enregistrement nécessitent d'adapter le modèle entre les données d'apprentissage et de test. Il s'agit d'un cas particulier du problème général d'adaptation de domaine. Gharib et al. [2019] ont défini un jeu de données spécifiquement pour ce problème dans le cas de la reconnaissance d'événements sonores.

Les différences de conditions d'enregistrement entre l'apprentissage et le test sont un problème récurrent et difficile en audio. En sus de la différence de matériel utilisé, d'autres différences comme le lieu d'enregistrement [Wysocki and Ladich, 2005], le niveau de bruit de fond lié notamment à la distance entre les sources sonores d'intérêt et le microphone [Clavel et al., 2005], la taille de la pièce, ou le nombre de microphones [Dekkers et al., 2017] jouent un rôle. Lorsque plusieurs microphones sont utilisés, l'information est plus riche, mais se pose la question de comment combiner les microphones, et comment les utiliser de manière efficace [Dekkers et al., 2017].

Annotations bruitées Lorsque les données audio sont collectées sur internet, en utilisant le canal audio de certaines vidéos (Youtube, Vimeo) ou bien des plateformes audio spécialisées comme Freesound [Font et al., 2013], les annotations correspondantes peuvent souffrir de multiples problèmes comme des annotations manquantes, des annotations très variées pour le même événement sonore et des annotations bruitées c'est-à-dire partiellement ou totalement erronées. Afin d'éviter le problème des annotations très variées, on utilise des ontologies qui permettent de regrouper les annotations définies. Ce problème peut néanmoins apparaître en raison de la difficulté à identifier certains événements sonores de façon non-ambiguë ou à les décrire au sein d'une taxonomie [Favory et al., 2018]. Pour éviter cela, Freesound a développé sa propre plateforme d'étiquetage collaboratif (*crowdsourcing*) d'événements sonores¹. Cette approche réduit la variabilité des annotations sans la supprimer totalement [Fonseca et al., 2017]. Le problème des annotations bruitées n'est pas spécifique à l'audio et concerne une multitude de domaines (image, texte, etc.) [Natarajan et al., 2013]. Choi et al. [2018] en ont proposé une analyse fine dans le cas de l'étiquetage de musique par exemple. Ce problème nécessite de concevoir des méthodes d'apprentissage robustes aux annotations bruitées [Fonseca et al., 2019a]. Li et al. [2017b] utilisent l'apprentissage par transfert pour réduire l'impact des annotations bruitées. Pour l'étiquetage d'événements sonores, Dorfer and Widmer [2018] proposent de ré-annoter les données potentiellement bruitées de façon itérative.

1. <https://annotator.freesound.org/>

Annotations manquantes Lorsque les annotations ne sont pas bruitées mais manquantes en partie ou en totalité, il faut trouver le moyen d'extraire de la connaissance des données sans annotations. L'absence d'annotations donne lieu à un problème d'apprentissage non-supervisé si elle est totale et à un problème d'apprentissage semi-supervisé si elle est partielle. L'apprentissage semi-supervisé est traité en détail dans la partie 2.3.3. L'apprentissage non-supervisé ne permet souvent pas de résoudre directement l'application visée puisqu'aucune annotation n'est disponible concernant cette application. Il peut toutefois permettre d'apprendre une représentation des données permettant de résoudre certaines applications avec un nombre faible d'annotations. Dans ce cas, le but est d'apprendre une représentation de haut niveau comme discuté dans la partie 2.2.3.

Annotations faibles Un autre problème lié à l'annotation des données concerne l'annotation partielle des clips audio. Le terme « annotation partielle » peut signifier que, parmi tous les événements sonores, seuls les événements d'intérêt sont annotés. Il s'agit alors d'un problème d'annotations manquantes lorsque l'application visée concerne certaines classes non-annotées. Ce terme peut aussi signifier que les annotations n'indiquent pas la temporalité des événements recherchés. Ce problème devient particulièrement important lorsque l'on essaye de faire de la détection d'événements sonores mais il a aussi été étudié pour l'étiquetage d'événements sonores [Xu et al., 2018]. Cette absence de temporalité dans les annotations porte le nom d'annotations faibles, par opposition aux annotations fortes qui comportent les instants de début et de fin précis de chaque événement, et elle mène à un problème d'apprentissage « faiblement supervisé ». Ce point est traité plus en détail dans la partie 2.3.4.

Apparition des événements L'analyse de sons ambiants soulève d'autres problèmes comme la polyphonie des événements sonores ou la distribution (équilibrée ou non) des classes d'intérêt dans le jeu d'apprentissage. Comme pour les annotations faibles, ces problèmes font souvent partie de problèmes plus complexes. De nombreuses méthodes ont été proposées pour traiter ces différents problèmes que ce soit pour la polyphonie [Cakir et al., 2017, Heittola et al., 2011, Pankajakshan et al., 2019], la distribution non-équilibrée des classes [Arora et al., 2019, Cakir et al., 2017] ou les annotations faibles [Kong et al., 2019, Wang et al., 2019b]. Dans la majorité des applications, ces problèmes apparaissent conjointement. À notre connaissance, aucune étude n'a cherché à analyser ces problèmes de manière isolée afin de mesurer leur impact respectif et d'identifier lesquels nécessitent une attention particulière.

Problèmes liés aux modèles Les réseaux de neurones sont devenus les modèles les plus courants dans le cadre de l'analyse de sons ambiants. Ces modèles permettent la résolution de problèmes complexes, grâce à l'utilisation d'un nombre important de paramètres. Mais, pour apprendre ces paramètres, une quantité importante de données est nécessaire. Dans le cadre de l'analyse de sons ambiants, il n'est pas facile d'acquérir et d'annoter un grand nombre d'exemples de chaque événement d'intérêt. Lorsque le nombre de paramètres à apprendre est trop important par rapport à la quantité de données disponible,

des problèmes de sur-apprentissage peuvent apparaître [Srivastava et al., 2014]. Des méthodes particulières peuvent être employées pour l'apprentissage automatique avec très peu de données par classe [Pons et al., 2019]. La complexité des réseaux de neurones artificiels constitue aussi un problème lorsque l'application est vouée à être déployée sur du matériel qui n'a pas la puissance de calcul suffisante (smartphone, IoT, *etc.*) [Cerutti et al., 2019]. Par ailleurs, les réseaux de neurones ne modélisent pas l'incertitude inhérente au modèle [Kendall and Gal, 2017], ce qui est un problème pour certaines applications (médicales par exemple). Enfin, ils peuvent souffrir d'un manque de robustesse car ils peuvent être dupés par des exemples dits antagonistes obtenus par une perturbation imperceptible des données d'entrée se traduisant par une sortie très différente [Subramanian et al., 2019].

2.3 Reconnaissance d'événements sonores en environnement réel

2.3.1 Contexte

La reconnaissance d'événements sonores en environnement réel est un problème complexe comportant une multitude de problèmes méthodologiques à traiter. Tout ou partie de ces problèmes peuvent apparaître en fonction de l'application visée.

Par exemple, si l'on cherche à reconnaître des bruits dans une rue passante à des fins d'analyse de la pollution sonore, le matériel d'enregistrement et sa position spatiale sont généralement fixes, de sorte que le problème d'adaptation à un nouveau microphone n'apparaît pas. Cependant les événements sont souvent polyphoniques avec un rapport signal-à-bruit qui peut être faible, et leur fréquence varie de façon importante au cours de la journée. Pour prédire ces effets, des données spatio-temporelles peuvent être acquises et utilisées [Cartwright et al., 2020]. Pour cette application, récupérer des données annotées est difficile, puisqu'elles ne peuvent être annotées par un utilisateur présent sur les lieux et doivent être annotées par des annotateurs plus tard. La granularité temporelle des annotations peut être assez grossière, puisque le but est de faire de l'analyse. Par exemple, on peut prendre des segments de 5 s et effectuer l'étiquetage des événements sonores. En revanche, un problème intéressant est de séparer les différentes sources sonores afin de mesurer leurs niveaux respectifs [Kavalerov et al., 2019]. On se retrouve alors avec des données non annotées pour ce problème puisque l'enregistrement de chaque source de façon isolée est impossible en général dans un scénario réel non joué.

Un autre exemple concerne les applications domotiques. Dans ce cas, l'acoustique de la pièce est différente dans chaque maison et il n'est pas possible de contrôler la position du ou des microphones. Parfois, le microphone utilisé n'est pas connu non plus. Pour cette application, la polyphonie est présente, les fréquences des événements peuvent varier, et des anomalies (bris de glace, *etc.*) peuvent apparaître. Ce sont autant de problèmes à résoudre pour des données dont les activités (scènes) pourraient être annotées par un utilisateur mais dont les événements sonores ne peuvent être annotés qu'après enregistrement. Une certaine précision temporelle est nécessaire pour assurer la réactivité des

applications domotiques, ce qui nécessite d'annoter fortement les données.

Dans cette thèse, nous étudions des problèmes d'étiquetage ou de détection d'événements sonores. Concernant la détection d'événements sonores, nous analysons entre autres les problèmes liés au rapport signal-à-bruit, à la durée des clips, aux différences d'annotations entre événements et à la séparation des événements sonores. Concernant l'étiquetage d'événements sonores, nous nous focalisons sur les problèmes surgissant lors de l'apprentissage semi-supervisé ou faiblement supervisé de représentation de haut niveau. L'ensemble de ces analyses sont faites sur un jeu de données contenant des données domestiques, mais certaines de ces analyses sont applicables à d'autres applications comme l'application d'urbanisme décrite ci-dessus.

2.3.2 Jeux de données

Le Tableau 2.2 présente un tour d'horizon des jeux de données faiblement annotées publiés pour l'analyse de sons ambiants ces dernières années. La création de ces différents jeux de données a été motivée par un problème méthodologique donné ou bien par une application. Parmi eux, la plupart sont constitués d'enregistrements sonores réels. Ceci s'explique par la quantité de données réelles disponibles et par l'effort raisonnable d'annotation. Les colonnes contenant le nombre de clips et le nombre d'heures montrent que, dans la plupart des cas, la quantité de données est néanmoins limitée. Audioset et YFCC100m échappent à cette règle. Audioset possède l'avantage de reposer sur une ontologie, qui est d'ailleurs réutilisée dans d'autres jeux de données. Cependant, la qualité d'annotation est médiocre pour un grand nombre de classes : certaines classes comme la parole et la musique sont sur-représentées, alors que d'autres ne contiennent qu'une poignée de clips ou bien un plus grand nombre de clips dont la majorité ne contiennent pas la classe annotée. YFCC100m utilise directement les étiquettes attribués par les utilisateurs, qui sont peu fiables et peuvent être ambigus. Bien que réels, ces jeux de données posent donc le problème de la conception d'algorithmes robustes aux annotations bruitées ou manquantes. Les jeux de données MIMII ou ToyADMOS comportent des données synthétiques associées à des annotations plus fiables, mais soulèvent un problème d'adaptation de domaine.

Une sélection de données Freesound a récemment été publiée (FSD50K). Elle contient un nombre important d'événements isolés qui peuvent être utiles à la création d'un plus grand nombre de données synthétiques et peut-être plus proches de la réalité que les jeux de données actuels. Les données restantes sont des scènes sonores enregistrées de façon plus réaliste. Cependant, la nature réelle ou simulée de chaque enregistrement n'est pas annotée. Il n'est donc pas possible de filtrer les données.

Le deuxième jeu de données récent est VGGsound qui contient le son et la vidéo de 200 000 vidéos Youtube. Ce jeu de données comporte des annotations automatiques ou des étiquettes fournies par le téléverseur de la vidéo Youtube. Une très faible partie de ces annotations (20 exemples par classe) sont vérifiées manuellement et les autres sont soit supprimées soit vérifiées grâce à un modèle prédictif. Aucune d'information n'est fournie sur la qualité des annotations et aucune étude n'a encore été conduite à ce sujet, ce qui permettrait de conclure quant au positionnement de ce jeu de données par rapport à

Nom	Br	Ma	Réel/Internet/ Synthétique	Durée clip	Nb clips	Nb classes	Poly- phonie	Domaine	Equi- libré	Durée (h)	Ontologie	Visuel
UrbanSound8k [Salamon et al., 2014]			FreeSound	max 4s	8k	10	Oui	Urbain	presque	8,8	Urban sound taxonomy	
ESC-50 [Piezak, 2015]			FreeSound	5s	2k	50	Non	Général	Oui	2,8		
CHIME-home [Foster et al., 2015]			CHIME	4s	6k	7	Oui	Domestique	Non	6,8		
YFCC100m [Thomae et al., 2016]			Flickr	ND	800k	ND	Oui	Général	Non			X
AudioSet [Gemmeke et al., 2017]			Youtube	max 10s	2,1M	527	Oui	Général	Non	5731	AudioSet	
Making sense of sound [Kroos et al., 2019]			FreeSound, ESC 50	5s	2k	60	Non	Général	Non	2,7		
YBSS-200 [Singh and Joshi, 2019]			Youtube	max 4s	2k	10	Seulement avec voix	Général	Oui	2,2		
FSDNonisy 18k [Fonseca et al., 2019a]	X		FreeSound	0,3-30s	18k	20	Oui	Général	Non	43	AudioSet	
FSDKaggle2019 [Fonseca et al., 2019b]	X		FreeSound YFCC	0,3-30s	29k	80	Oui	Général	Non	103	AudioSet	
MIMI [Purohit et al., 2019]			Synthétique	10s	32k	2	Non	Industrie	Non			
ToxADMOS [Koizumi et al., 2019]			Synthétique	10s/10min	15k	2	Non	Industrie	Fortement	570		
SONYC-ust-V2 [Cartwright et al., 2020]			SONYC	10s	18k	31	Oui	Urbain	Non	51	SONYC	
VGGSound [Chen et al., 2020]	X		Youtube	10s	200k	309	Oui	Général	Non	555		X
FSD50K [Fonseca et al., 2020]	X		FreeSound	0,3-30s	50k	200	Oui	Général	Non	108		

TABLEAU 2.2 – Jeux de données faiblement annotés. Br/Ma indique des annotations bruitées, manquantes ou les deux, ND signifie non déterminé.

Audioset.

L'ensemble de ces jeux de données prouve la difficulté de trouver des données adaptées à une application réelle quelconque. Lorsqu'une application réelle se présente et nécessite des données, on peut donc imaginer la difficulté d'acquisition de celles-ci lorsqu'elles sont très spécifiques et ne font pas partie des jeux de données disponibles.

Le Tableau 2.3 présente quelques jeux de données fortement annotées publiés ces dernières années pour l'analyse de sons ambiants. Dans l'ensemble, on constate au vu du nombre de clips et du nombre d'heures (fortement annoté) que très peu de données sont disponibles. Cela illustre le coût et la difficulté à produire ce type d'annotations. Le nombre important de jeux de données synthétiques (7 parmi 13) illustre aussi cette difficulté et soulève encore une fois le problème de l'adaptation de domaine. Le jeu de données DCASE 2017 Task 4 comporte des données faiblement annotées issues d'AudioSet pour l'apprentissage d'un système de détection d'événements sonores et n'utilise les données fortement annotées que pour l'évaluation de ce système. L'utilisation de données faiblement annotées augmente fortement la quantité de données d'apprentissage mais le problème d'apprentissage devient alors faiblement supervisé. Pour un effort d'annotation similaire, une quantité importante de données peuvent être faiblement annotées, mais cela nécessite de développer un système capable d'estimer les instants de début et de fin approximatifs des événements sonores. Les autres jeux de données sont généralement définis pour traiter des problèmes méthodologiques précis puis permettre la résolution d'applications réelles dans un second temps. Ces jeux de données nous montrent la difficulté à annoter fortement les données acquises en environnement réel. Cela constitue la motivation centrale des recherches sur l'apprentissage semi-supervisé ou faiblement supervisé.

2.3.3 Exploitation de données partiellement annotées

Parmi les problématiques observées dans les scénarios réels, l'absence d'annotation pour tout ou partie des données est probablement une des plus fréquentes. En effet, la disponibilité accrue de données sur internet et la facilité d'enregistrement de nouvelles données comparées au coût d'annotation incitent souvent à ne pas annoter une partie des données. Pour contourner ce problème, il est possible d'utiliser une représentation de haut niveau apprise sur les données non-annotées ou sur des données annotées d'un autre domaine. L'intérêt de l'utilisation de ces représentations de haut niveau est de réduire la partie du modèle qui est spécifique à la tâche donnée. Afin de traiter un problème semi-supervisé, il existe plusieurs méthodes détaillées ci-dessous basées sur la ré-annotation automatique, l'apprentissage actif, l'apprentissage de représentation de haut niveau, ou l'apprentissage par similarité.

Apprentissage semi-supervisé par ré-annotation Une possibilité pour utiliser la partie des données non-annotée est de ré-annoter automatiquement ces données et d'estimer la confiance dans ces nouvelles pseudo-annotations. [Moreno and Agarwal \[2003\]](#) étudient l'impact de l'utilisation d'un seuil de confiance. Ils comparent un algorithme itératif qui

Nom	N-A	F-A	Réel/Internet/ Synthétique	Durée clip	Nb clips	Nb classes	Poly- phonie	Domaine	Equi- libré	Durée (h) / fortement annoté (si différent)	Ontologie	V
CHIL Interactive Seminar database [Stiefelhagen et al., 2007]			CLEAR	5-20min	45	12	Non	Seminaire	Non	5		
MIVIA [Pogea et al., 2015]			Synthétique	~3min	580	3	Non	Alert	Oui	29		
DCASE 2016 Task 2 [Mesaros et al., 2018a]			Synthétique	2min	72	11	Oui	Bureau	Non	2,4		
TUT Sound events 2016 [Mesaros et al., 2016b]			Enregistrements (micro intra-oculaire)	3-5min	32	18	Oui	Résidentiel + domestique	Non	1,8		
DCASE 2017 Task 4 [Mesaros et al., 2017a]	App		sous partie d'Audioset	max 10s	+52k	17	Oui	Urbain	Non	144 / 4,3		
TUT Rare sound events [Mesaros et al., 2017a]			Synthétique	30s	3k	3	Non	Alertes	Oui	9 / 4,5		
TUT Sound events 2017 [Mesaros et al., 2017a]			TUT	3-5min	24	6	Oui	Rue	Non	2		
Urban SED [Salamon et al., 2017]			Synthétique	10s	10k	10	Oui	Urbain	Presque	30	Urban sound taxonomy	
AVE [Tian et al., 2018]		X	Sous partie Audioset	10s	+4k	28	Oui	Général	Non	11,5		X
Voice [Gharib et al., 2019]			Synthétique	0,8-3 min	1449	3×4	Oui	Alertes	Oui	38,6		
TAU Spatial sound events [Adavanne et al., 2019b]			Synthétique	1 min	500	11	Oui	Bureau	Oui	8,3		
DESED	X	X	Youtube + Synthétique	10 s	+20k	10	Oui	Domestique	Non	50 / 5h	Audioset	
DESED synthetic			Synthétique	varie	varie	10	possible	Domestique	Non		Audioset	
TAU-NGENS Spatial Sound Events 2020 [Poitits et al., 2020]			Synthétique	1 min	800	14	Oui	Résidentiel	Non	13,3		
TMAVD [Ziemanas et al., 2019]			Réel	5 min	47	21	Oui	Urbain	Non	4	MAYD	X

TABLEAU 2.3 – Jeux de données fortement annotés. App signifie apprentissage, N-A signifie non annoté, F-A signifie faiblement annoté et V signifie visuel.

ré-annote l'ensemble des données non-annotées à chaque itération pour un nombre d'itérations fixé avec un algorithme incrémental qui sélectionne à chaque itération une partie des exemples de chaque classe (ceux avec la plus grande confiance) à annoter et s'arrête lorsque toutes les données sont annotées. Cet algorithme incrémental est plus instable et dans de nombreux cas pas aussi efficace que l'algorithme itératif. Cette étude a été menée sur des données de parole pour l'identification de l'identité et du genre du locuteur. Concernant l'étiquetage d'événements sonores, [Zhang and Schuller \[2012\]](#) proposent de ré-annoter automatiquement les données non-annotées uniquement lorsque les pseudo-annotations dépassent un certain seuil de confiance et ils font varier ce seuil pour en montrer l'impact. La ré-annotation apporte un gain par rapport à l'approche supervisée utilisant peu de données, mais ce gain est plutôt minime. Le co-apprentissage consiste à apprendre deux classifieurs de sorte que chacun indique à l'autre quelles données sont à annoter et à rajouter dans leur jeu d'apprentissage. [Shi et al. \[2019\]](#) utilisent une méthode inspirée du co-apprentissage afin de ré-annoter automatiquement les données mais cette fois-ci en utilisant trois modèles. Les données associées à un modèle sont ré-annotées lorsque les deux autres modèles sont d'accord sur la pseudo-annotation d'un exemple avec un score de confiance élevé. Chacun des trois modèles est appris en utilisant une partie différente (avec recouvrement) du jeu de données annotées tirée par un algorithme de bootstrapping. Cette approche s'avère efficace mais le nombre de données ré-annotées ne doit pas être trop important sinon les performances se dégradent. [Kumar et al. \[2019\]](#) comparent le co-apprentissage et l'utilisation de bootstrapping à chaque itération de l'apprentissage pour réduire l'impact des mauvaises annotations et montrent que le co-apprentissage s'avère plus efficace.

Apprentissage semi-supervisé actif Dans la mesure où l'apprentissage semi-supervisé ré-annote les données en lesquelles le modèle a le plus confiance, cela a tendance à conforter le modèle dans son comportement. À l'opposé, l'apprentissage actif consiste à confier à un humain la tâche de ré-annotation des données en lesquelles le modèle a le moins confiance. L'apprentissage actif fonctionne en itérant les phases d'annotation et d'apprentissage afin d'annoter de façon efficace : l'apprentissage permet de définir quelles annotations apporteraient le plus de connaissance au système et les annotations permettent d'améliorer le système. L'apprentissage actif est très intéressant pour des applications réelles mais pose un certain nombre de problèmes notamment dus au choix des données à annoter après l'apprentissage [[Ducoffe and Precioso, 2018](#), [Gal et al., 2017](#), [Yu et al., 2010](#)]. L'estimation de l'incertitude sur les pseudo-annotations est un point-clé. On peut retrouver des méthodes basées sur l'entropie [[Yu et al., 2010](#)], des méthodes bayésiennes [[Gal et al., 2017](#)] ou bien basées sur des exemples antagonistes [[Ducoffe and Precioso, 2018](#)]. Concernant la reconnaissance d'événements sonores, [Zhao et al. \[2020\]](#) ont commencé par une approche simple qui utilise la représentation en clusters pour permettre l'apprentissage actif en étiquetage d'événements sonores, puis ils ont proposé une approche plus complexe basée sur les mêmes principes mais étendue pour améliorer la sélection des données et permettre leur annotation faible, pour finalement faire de la détection d'événements sonores faiblement supervisée.

Apprentissage semi-supervisé de représentation de haut niveau L'apprentissage semi-supervisé peut aussi viser à apprendre une représentation de haut niveau en combinant un coût basé sur les distances pour les données non-annotées et un coût supervisé pour les données annotées. [Darna-Sequeiros and Toledano \[2018\]](#) montrent l'intérêt d'apprendre une telle représentation et comparent différentes architectures sur Audioset. [Weston et al. \[2018\]](#) proposent d'apprendre une représentation de haut niveau basée sur les triplets, en s'appuyant sur la temporalité des images dans une vidéo ou le fait que des phrases soient adjacentes dans un texte. Dans un autre registre, [Thulasidasan and Bilmes \[2017\]](#) proposent l'apprentissage d'un système de reconnaissance de la parole basé sur des graphes où les arêtes représentent la distance euclidienne entre les vecteurs de représentation. L'algorithme des k plus proches voisins (*k-nearest neighbors*) est alors utilisé pour créer des mini-lots appropriés pour l'apprentissage (compromis entre bonne connectivité des arêtes pour régulariser le graphe et bonne diversité pour la descente de gradient). Pour la reconnaissance d'événements sonores, [Komatsu et al. \[2016\]](#) utilisent des NMF et l'algorithme des k -moyennes (*k-means*) de façon non supervisée pour créer des centroïdes ensuite utilisés pour guider une NMF utilisée de façon supervisée sur la partie annotée. Cette façon de résoudre le problème s'apparente à une forme d'apprentissage par transfert excepté que, dans le cas de l'apprentissage par transfert, la représentation de haut niveau utilisée est généralement apprise grâce à des données d'un autre domaine. Par exemple, [Pons et al. \[2019\]](#) proposent de faire de l'apprentissage avec très peu de données annotées (*few shot learning*), et utilisent la représentation intermédiaire apprise sur Audioset [[Hershey et al., 2017](#)] pour initialiser l'apprentissage de leur réseau prototype et faire du transfert de connaissance pour une application urbaine et une application de classification de scènes sonores.

Apprentissage semi-supervisé par similarité La dernière approche possible est d'utiliser la similarité ou la dissimilarité entre les exemples directement dans l'apprentissage. Dans cette approche on retrouve l'utilisation de plusieurs modèles appris ou non. [Dai et al. \[2017\]](#) utilisent la même application mais utilisent une méthode d'apprentissage antagoniste basé sur l'utilisation d'une donnée issue du corpus et d'un générateur puis d'un discriminateur devant indiquer laquelle est la donnée issue du corpus. Ils font une analyse des différents modèles de génération nécessaires pour obtenir de bonnes performances en apprentissage semi-supervisé et concluent qu'un mauvais générateur tend à donner une meilleure généralisation du modèle. [Laine and Aila \[2016\]](#) proposent l'utilisation du « modèle π » et d'une revisite de celui-ci pour une application de classification d'images. Le modèle π consiste à utiliser deux modèles similaires avec des entrées et des fonctions d'épuration de poids différentes entraînés à estimer des sorties similaires (coût de consistance). Ils montrent l'intérêt de ces méthodes et l'amélioration des résultats avec le nombre de données apportées indique que la méthode peut bénéficier de l'apport de données non-supervisées massives. En détection d'événements sonores, la méthode du professeur moyen (*mean teacher*) est devenue très utilisée. Cette méthode est proche du modèle π mais elle repose sur deux modèles distincts : l'un est appris sur les données et les poids de l'autre sont calculés comme la moyenne glissante exponentielle des poids du

premier. Cette méthode a d'abord été proposée par [Tarvainen and Valpola \[2017\]](#) pour la classification d'images. Nous en discuterons plus en détail dans le Chapitre 3.

2.3.4 Annotations faibles

Les annotations faibles sont des annotations qui n'indiquent que partiellement l'information souhaitée. Dans notre cas, ce sont des annotations indiquant les événements sonores présents dans un clip audio mais pas leurs instants de début et de fin. Dans le cadre de l'étiquetage d'événements sonores, ces annotations concordent avec la granularité souhaitée lors de la phase d'évaluation, mais elles peuvent tout de même avoir un impact lors de la phase d'apprentissage. Le problème des annotations faibles n'est pas spécifique à l'audio. [Zhou \[2018\]](#) présente les différentes catégories d'annotations faibles dans divers domaines. Dans notre cas, nous ne discutons pas l'ensemble de ces catégories mais seulement des annotations incomplètes.

Annotations faibles dans différents domaines La vision par ordinateur est l'un des domaines dans lesquels les annotations faibles sont présentes. [Guo et al. \[2018\]](#) proposent une solution à la classification d'image avec des annotations faibles et bruitées en regroupant les images pour déterminer si le bruit est « important » ou non. [Lu et al. \[2017a\]](#) utilisent un réseau convolutionnel pour la segmentation d'image à partir d'annotations faibles. Ils assignent les annotations des différentes classes d'objets à retrouver à des superpixels et essaient de réduire le bruit dans les annotations des superpixels. De manière itérative, ils réduisent le bruit des annotations, ré-apprennent un modèle avec ces superpixels, et ré-annotent les superpixels.

On retrouve aussi souvent des annotations faibles dans le contexte musical. [Huang et al. \[2020a\]](#) traitent l'étiquetage de musiques pour la recommandation musicale. Ils disposent d'annotations bruitées et faibles représentant le contenu d'une musique ainsi que des données d'écoute par les utilisateurs. Ils construisent un graphe indiquant quelles musiques sont écoutées régulièrement par les mêmes utilisateurs et l'utilisent pour apprendre une représentation basée sur les triplets. Ils classifient ensuite le contenu musical en utilisant les annotations faibles et bruitées ainsi que ce graphe de co-écoute. [Little and Pardo \[2008\]](#) essaient quant à eux de reconnaître les instruments présents dans une musique. Ils comparent des arbres extrêmement aléatoires, l'algorithme des k plus proches voisins et un SVM dans deux conditions : une où les notes des instruments sont segmentées (fortement supervisé) et l'autre avec des mix d'instruments (faiblement supervisé). Ils obtiennent de meilleures performances dans le second cas, et expliquent que cela est probablement dû à la différence de conditions entre l'apprentissage et le test pour l'apprentissage fortement supervisé puisqu'ils ont des données isolées. Cependant, ceci pointe un aspect intéressant des données faiblement annotées. Il est souvent facile de trouver des données faiblement annotées sur internet ou d'annoter des données de façon faible, mais il est plus difficile d'annoter des données (audio) de façon forte. Les données fortement annotées disponibles correspondent parfois à des cas idéaux car plus faciles à annoter mais ne reflétant pas la réalité.

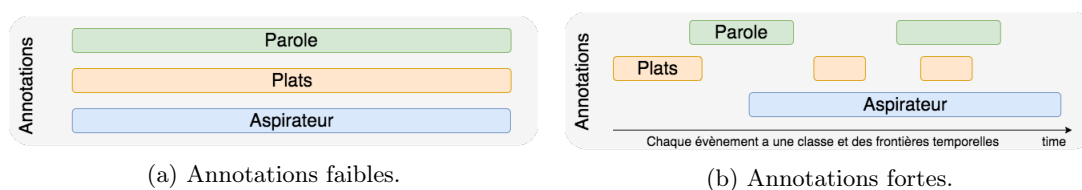


FIGURE 2.6 – Annotations faibles et fortes pour la reconnaissance d'événements sonores.

Dans la Figure 2.6 nous présentons les deux types d'annotations pour la reconnaissance d'événements sonores. La partie gauche représente les annotations faibles qui n'ont pas de frontières temporelles et la partie droite les annotations fortes, avec les différentes occurrences de chaque événement sonore et leurs positions temporelles.

Annotations faibles en reconnaissance d'événements sonores Pour l'étiquetage d'événements sonores, les annotations faibles pourraient sembler de faible importance puisque l'on cherche à identifier seulement quels événements apparaissent dans un temps donné, peu importe le moment. Cependant, on peut comparer ce problème à la reconnaissance d'instruments [Little and Pardo, 2008], où il est difficile d'identifier automatiquement les différents instruments dans un enregistrement audio sans avoir préalablement identifié la région temporelle dans laquelle chacun apparaît. Beaucoup de méthodes ont été proposées pour résoudre le problème des annotations faibles pour la reconnaissance d'événements sonores puisque c'est un problème récurrent étant donné le coût des annotations fortes. Ces méthodes se divisent en trois catégories : apprentissage multi-instance, agrégation, et segmentation.

Apprentissage multi-instance Kumar and Raj [2016] comparent des approches d'apprentissage multi-instance pour les annotations faibles en utilisant un GMM, un SVM ou un réseau de neurones simple. Ils évaluent aussi l'impact négatif des annotations faibles par rapport aux annotations fortes (données annotées à la seconde) pour leur problème. Su et al. [2017] proposent plusieurs solutions pour résoudre le problème des annotations faibles en utilisant des méthodes d'apprentissage multi-instance. Ils utilisent différentes échelles d'entrée pour permettre au réseau d'apprendre différentes structures temporelles, Ils apprennent un CNN puis sur-échantillonnent ses sorties pour obtenir des sorties de même temporalité qu'en entrée. Ils proposent d'utiliser 3 échelles différentes pour les données d'entrée, d'augmenter les données et d'apprendre des filtres gaussiens adaptés à chaque classe pour résoudre le problème des annotations faibles. Kong et al. [2019] analysent les annotations faibles du corpus Audioset. Ils présentent de façon détaillée différentes possibilités de l'apprentissage multi-instance (espace d'instances, espace de lots, espace de représentation). Leur travail est basé sur l'apprentissage par attention en proposant une nouvelle méthode d'attention non pas basée sur l'attention au niveau des décisions, mais au niveau des représentations qui montre de meilleurs résultats que les autres formes d'attention. Ils analysent aussi Audioset pour l'équilibre des données et la qualité des annotations mais, comme on peut le voir dans cette étude, il est difficile

de conclure pour chacun des problèmes présentés lorsque l'ensemble des problèmes sont présents dans le jeu de données et non isolés. Une des solutions serait de trouver un moyen d'isoler ce problème pour le jeu de données et, lorsque ceci n'est pas possible, de définir un scénario où l'analyse est possible.

Agrégation Pour résoudre le problème des annotations faibles dans l'étiquetage d'événements sonores, [Kim and Ghaffarzadegan \[2019\]](#) utilisent une méthode basée sur l'attention, qui consiste à multiplier les sorties fortes d'un réseau de neurones (à l'échelle de petites fenêtres temporelles) par une matrice de sorties fortes normalisées avant de les agréger temporellement pour obtenir les sorties faibles (à l'échelle du clip sonore). Ils utilisent à la fois des annotations faibles et fortes pour apprendre leur modèle et, lorsque seulement des annotations faibles sont disponibles, ils ont recours aux pseudo-annotations fortes pour l'utilisation de leur coût d'attention utilisant les annotations fortes. Dans l'ensemble des cas ils utilisent des couches pré-apprises du modèle 8M-VGGish qui est un modèle pré-entraîné sur des données de Youtube-8M. Ils ajustent leur modèle aux différents jeux de données qu'ils utilisent (DCASE 2017 Task 4 et Audioset équilibré). Ils proposent différentes formes d'attention (sans supervision, avec annotations fortes, et auto-supervisée qui utilise des pseudo-annotations). Ils montrent que l'attention, peu importe sa forme, aide pour des annotations fortes ou faibles. [McFee et al. \[2018\]](#) comparent les fonctions d'agrégation classiques par moyenne ou maximum avec une fonction d'agrégation qu'ils proposent, basée sur un softmax avec un paramètre appris et initialisé de sorte à être proche d'agrégation par moyenne au début de l'apprentissage. [Wang et al. \[2019b\]](#) comparent les agrégations par moyenne, maximum, softmax linéaire, softmax exponentiel, et par attention et montrent que la meilleure agrégation dépend du jeu de données utilisé (Audioset, DCASE 2017 Task 4). Ceci tend à montrer la difficulté du problème puisque les annotations faibles varient grandement avec la durée de l'événement sonore et donc la classe d'événements analysée. [Adavanne et al. \[2019a\]](#) se focalisent sur l'agrégation par attention après un nouveau type de CNN permettant d'obtenir des sorties fortes et faible. Ils montrent que la couche d'attention améliore les résultats et permet la réduction des paramètres du modèle. [Liu et al. \[2020\]](#) proposent une nouvelle méthode d'agrégation appelée agrégation par puissance puisqu'elle est basée sur le carré des sorties au numérateur. Ils obtiennent des résultats meilleurs que l'agrégation par attention et meilleurs bien que proches de l'agrégation linéaire. [Kong et al. \[2020\]](#) ne se focalisent pas sur l'agrégation mais comparent des méthodes basées sur l'attention apprise par des modèles de type Transformer. Ils comparent différents types de réseaux de neurones (CNN, réseau de neurones convolutionnel et récurrent (convolutional recurrent neural network) (CRNN)) utilisant ou non l'attention, et proposent d'utiliser des seuils automatiques pour la détection des événements sonores.

Segmentation Pour estimer les instants de début et de fin des événements sonores, [Kothinti et al. \[2018\]](#) détectent les pics sonores par une approche basée sur la dérivée ou sur le filtre de Kalman. Cette approche se montre efficace pour détecter les débuts d'événements sonores, puisque les événements d'intérêt ont très souvent un début très

prononcé. En revanche, la détection des fins d'événements, qui bien souvent n'est pas aussi nette que les débuts, n'est pas améliorée par cette méthode. [Chan et al. \[2019\]](#) proposent d'utiliser la NMF pour segmenter les événements, en utilisant la séquence d'activations temporelles correspondant à chaque événement pour déterminer à quel moment il est actif. Les jeux de données utilisés pour ces deux travaux sont les mêmes. Les résultats obtenus par [Kothinti et al. \[2018\]](#) sont meilleurs, cependant il est difficile de conclure que c'est grâce à une meilleure segmentation puisque les réseaux de neurones utilisés diffèrent.

Dans ce chapitre, nous avons décrit l'état de l'art général pour l'analyse des sons ambiants avec un focus particulier sur les problèmes que nous allons étudier dans cette thèse. En règle générale, nous avons vu dans ce chapitre que ces problèmes sont difficiles à isoler. De plus, il n'existe pas de jeu de données directement utilisable ni de problèmes bien définis pour la détection d'événements sonores en environnement réel domestique. Nous présentons plus en détail dans la suite de ce manuscrit la définition d'un problème et la création d'un corpus de détection d'événements sonores, l'analyse de différents problèmes liés à la détection d'événements sonores, l'apprentissage semi-supervisé de représentations de haut niveau et l'analyse de l'impact des annotations faibles.

3 Détection d'événements sonores en environnement domestique (Tâche 4 du Challenge DCASE)

Depuis 2018, j'ai joué un rôle moteur dans l'organisation de la Tâche 4 du Challenge DCASE. J'ai participé à la définition du problème, la collecte des données et la conception de la tâche et des jeux de données, et j'ai été responsable de la conception des systèmes de référence et de l'analyse des résultats.

Dans ce chapitre, nous définissons le problème de détection d'événements sonores environnement domestique proposé. Ensuite nous expliquons les étapes de création d'un corpus et discutons les différents choix faits à cet égard. Enfin, nous présentons les résultats officiels du challenge et discutons certaines des solutions proposées par les participants.

3.1 Définition de la tâche et évaluation

3.1.1 Application

En 2018, nous avons proposé une nouvelle tâche du Challenge DCASE, la Tâche 4, qui concerne la détection d'événements sonores en milieu domestique pour des applications liées à la maison intelligente et à l'assistance aux personnes en perte d'autonomie. Le but est à partir d'un unique microphone (par opposition aux scénarios utilisant de multiples capteurs pour chaque type d'objet) de reconnaître les différents événements dans une pièce de la maison. Dans l'optique de traiter ce problème général, nous nous concentrons tout d'abord sur une tâche simplifiée, se limitant à un sous ensemble d'événements sonores de la vie quotidienne. Les classes d'événements sonores choisies sont celles qui apparaissent régulièrement lors d'une journée et qui restent identifiables par un humain avec un degré de confiance raisonnable. Nous voulons une détection temporelle fine pour limiter le délai entre le bruit d'intérêt et l'information de l'utilisateur ou la prise de décision (semi-)automatique, sans aller jusqu'à la détection en temps réel qui requiert un degré d'ingénierie plus important. Nous nous plaçons dans un scénario jugé réaliste en ce qui concerne la disponibilité de données annotées correspondant à la tâche, ou la possibilité d'en annoter avec un budget limité. Cela implique qu'il est impossible d'annoter fortement une quantité importante de données.

3.1.2 Définition de la tâche

Caractéristiques de la tâche La tâche de détection d'événements sonores en milieu domestique proposée depuis 2018 est définie autour de différents aspects caractéristiques :

- Détection d'événements sonores avec une granularité temporelle fine (trouver les instants de début et de fin).
- Scénario domestique en intérieur.
- Niveau d'annotation hétérogène (données faiblement annotées, non-annotées et fortement annotées).
- Données hétérogènes : données enregistrées et données synthétiques.
- Pas de données enregistrées fortement annotées disponibles à l'apprentissage.
- Faible quantité de données faiblement annotées mais annotations fiables.
- Événements sonores variés et potentiellement simultanés (polyphonie).

Nous ne nous intéressons qu'aux événements appartenant aux classes d'intérêt pour l'application visée. Les autres événements sonores ne sont pas analysés. Du point de vue applicatif, cette hypothèse est justifiée par le fait que l'application finale visée n'a besoin que de ces classes d'intérêt pour fonctionner. Il n'est pas nécessaire de détecter les autres classes et les annoter serait trop coûteux par rapport à l'apport applicatif. Par exemple, si l'application d'intérêt est la détection de bruits quotidiens dans une maison, il n'est pas nécessaire de détecter le bruit d'un oiseau qui chante ou d'engins de travaux, même si ceux-ci apparaissent dans certains enregistrements audio.

En pratique, un nombre limité de classes d'événements sonores est donc choisi comme étant classes d'intérêt. Nous utilisons des données disponibles sur internet plutôt que d'organiser une phase d'enregistrement, ce qui nous permet de favoriser la diversité des données et d'éviter les conditions trop contrôlées. En 2018, nous avons proposé d'utiliser des données faiblement annotées en faible quantité et des données non-annotées en grande quantité (une partie avec probabilité importante de contenir les événements d'intérêt, l'autre non). En 2019, nous avons proposé d'utiliser un jeu de données synthétiques issues d'un mélange de bruits de fond et d'événements d'intérêt isolés. Ces données représentent souvent une scène sonore moins réaliste mais présentent l'avantage d'être annotées fortement et d'être plutôt faciles à générer. En 2020, nous avons proposé d'utiliser la séparation de sources afin de faciliter la détection d'événements sonores simultanés.

Notations Soit $\mathbf{a} \in \mathbb{R}^T$ le signal temporel correspondant à un clip audio, représenté sous forme d'un vecteur de T échantillons, et soit \mathbf{X} sa représentation de bas niveau (temps-fréquence). \mathbf{X} est une matrice de taille $F \times T$ où F est le nombre de bandes de fréquence et T est le nombre de fenêtres temporelles. Nous cherchons à estimer les instants de présence des événements sonores d'intérêt représentés par la matrice $\mathbf{Y} \in \mathbb{R}^{C \times T}$, où chaque élément y_{ct} de \mathbf{Y} est égal à 1 si l'événement c est présent à l'instant t et à 0 sinon. Nous notons par $\mathcal{C} = \{1, \dots, C\}$ l'ensemble des classes d'intérêt et par C le nombre de classes. Plusieurs événements sonores d'intérêt (de classes différentes ou non)

peuvent apparaître au même moment au sein d'un clip audio. Nous sommes donc face à un problème de détection de classes sonores multiples et simultanées (polyphonie). Pour résoudre ce problème, on utilise un système f_θ dont les paramètres θ sont appris sur un corpus d'apprentissage. On note par $\hat{\mathbf{Y}} = f_\theta(\mathbf{a})$ les sorties du système associées au signal d'entrée \mathbf{a} .

Dans la suite de ce chapitre, nous définissons les différentes méthodes d'évaluation possibles et justifions le choix de la métrique retenue pour la Tâche 4. Nous définissons ensuite le corpus utilisé pour l'apprentissage et l'évaluation des systèmes. Une description détaillée de certains exemples de systèmes f_θ est fournie dans le Chapitre 4.

3.1.3 Évaluation

Afin d'évaluer la tâche de détection des événements sonores d'intérêt dans un clip audio, Mesaros et al. [2016a] et Bilén et al. [2020] ont proposé plusieurs métriques. Toutes ces métriques sont basées sur le comptage de *vrai positif* (VP), *faux positif* (FP), et *faux négatif* (FN).

- Un vrai positif correspond à un événement correctement détecté.
- Un faux positif correspond à la détection d'un événement sonore alors que celui-ci n'est pas présent en réalité.
- Un faux négatif correspond à un événement non détecté.

Les critères utilisés pour déterminer les VP, FP, FN peuvent varier selon la façon de prendre en compte la temporalité des événements.

Calcul par segment Le nombre de VP, FP et FN peut être déterminé selon un critère par segments. Cette solution consiste à segmenter le clip audio en segments plus courts de durée fixe. Les annotations sont modifiées pour indiquer uniquement la présence ou l'absence de chaque événement dans chaque segment. Cela équivaut au calcul d'une métrique d'étiquetage d'événements sonores pour laquelle on peut faire varier la durée des segments (on peut utiliser un nombre plus ou moins important de segments dans un clip) [Mesaros et al., 2016a]. Cette approche ne tient pas compte de la durée exacte des événements, dans la mesure où les événements plus courts que la durée du segment sont représentés de la même façon que les événements qui couvrent la totalité du segment. La durée des événements plus longs se traduit quant à elle par le nombre plus ou moins grand de segments successifs dans lequel ces événements sont présents. Si un événement est présent dans 3 segments de suite, ne le détecter que dans une seule fenêtre pour chacun de ces segments est suffisant. Cela implique que les événements longs sont représentés dans un grand nombre de segments et prennent une part importante dans le calcul de la métrique par rapport aux événements courts. Un exemple est présenté dans la Figure 3.1.

Calcul par événement Alternativement, le nombre de VP, FP et FN peut être déterminé selon un critère par événement. Dans ce cas, on utilise toujours le clip audio en

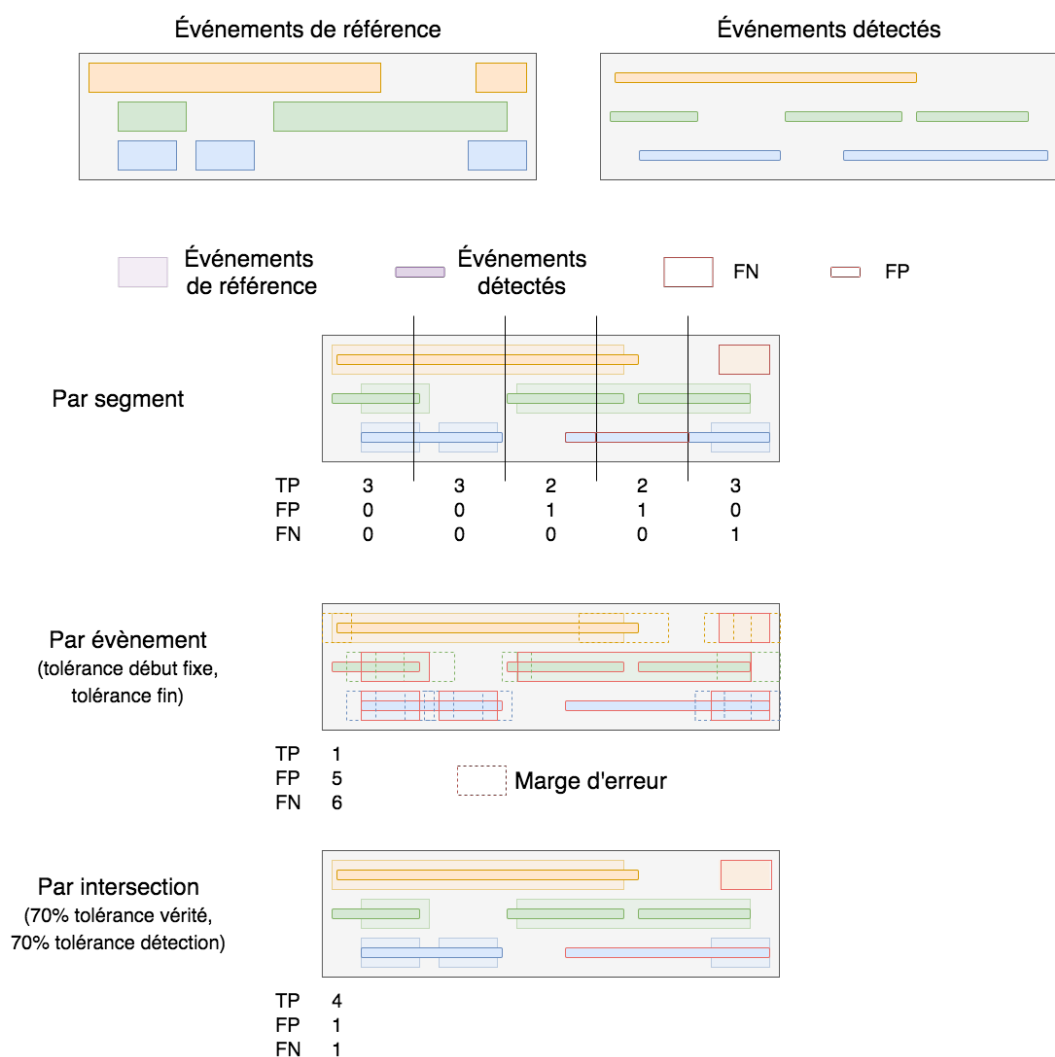


FIGURE 3.1 – Comparaison des différentes méthodes de comptage associées aux métriques utilisées pour la détection d'événements sonores.

entier, et on comptabilise un VP lorsqu'un événement détecté a les mêmes frontières temporelles qu'un événement annoté, à une marge d'erreur près, comme représenté dans la Figure 3.1 [Mesaros et al., 2016a]. La marge d'erreur est prise aux frontières de l'événement sonore, et la correspondance est faite entre événement et annotation. Il ne peut donc pas y avoir de micro-coupage dans la détection d'un événement long par exemple, sinon cela serait compté comme une erreur. Cela implique que le nombre d'événements détectés et annotés est important car chaque événement détecté se doit de correspondre à un événement de référence. Cette mesure présente l'avantage de prendre en compte la durée des événements. La marge d'erreur choisie peut être fixe, dépendre des classes ou bien des durées des événements à reconnaître. Un inconvénient de cette métrique est la

façon dont des événements temporellement très proches sont traités. Si deux événements temporellement très proches sont regroupés en un seul dans l'annotation, alors les événements détectés doivent être regroupés aussi sous peine d'être comptés comme une erreur. Ce problème de regroupement est difficile à gérer en pratique.

Calcul par intersection Enfin, le nombre de **VP**, **FP** et **FN** peut aussi être déterminé selon un critère d'intersection, comme représenté dans la Figure 3.1. Dans ce cas, le recouvrement temporel entre les événements détectés et les événements annotés est calculé [Bilen et al., 2020]. La comptabilisation d'un événement détecté comme TP est définie par le pourcentage de recouvrement minimum nécessaire entre la détection et la référence. Cette métrique autorise des coupures entre les événements de référence ou les événements détectés, comme on peut le voir dans la Figure 3.1. Le nombre d'événements détectés et annotés n'est donc pas important tant qu'ils s'intersectent suffisamment.

3.1.3.1 Calcul de la métrique

F-mesure Les nombres de **VP**, **FP** et **FN** obtenus sont utilisés pour calculer une ou plusieurs métriques qui reflètent les performances du système évalué. La précision P représente le taux de bonnes détections sur le nombre total de détections :

$$P = \frac{VP}{VP + FP}. \quad (3.1)$$

Le rappel R correspond au taux d'événements qui ont été correctement détectés sur le nombre d'événements de référence :

$$R = \frac{VP}{VP + FN}. \quad (3.2)$$

La Figure 3.2 illustre le calcul de la précision et du rappel. Alors que la précision se focalise sur les détections, le rappel se focalise sur les événements de référence. La F-mesure est une moyenne harmonique entre la précision et le rappel :

$$F = \frac{2 \cdot P \cdot R}{P + R}. \quad (3.3)$$

Cette quantité, majorée par 1 pour un système parfait, se concentre à la fois sur les détections et les événements de référence. Elle mesure l'efficacité du système à détecter correctement les événements de référence sans introduire trop de **FP**. L'objectif est d'avoir la F-mesure la plus élevée possible.

Taux d'erreur Une métrique alternative consiste à compter le nombre d'erreurs de substitution, d'insertion et de suppression. Pour un clip audio, nous définissons le nombre de substitutions (**Sub**) comme le nombre de détections correspondant temporellement à un événement de référence mais pour lesquelles la classe détectée est incorrecte :

$$\text{Sub} = \min(\text{FN}, \text{FP}). \quad (3.4)$$

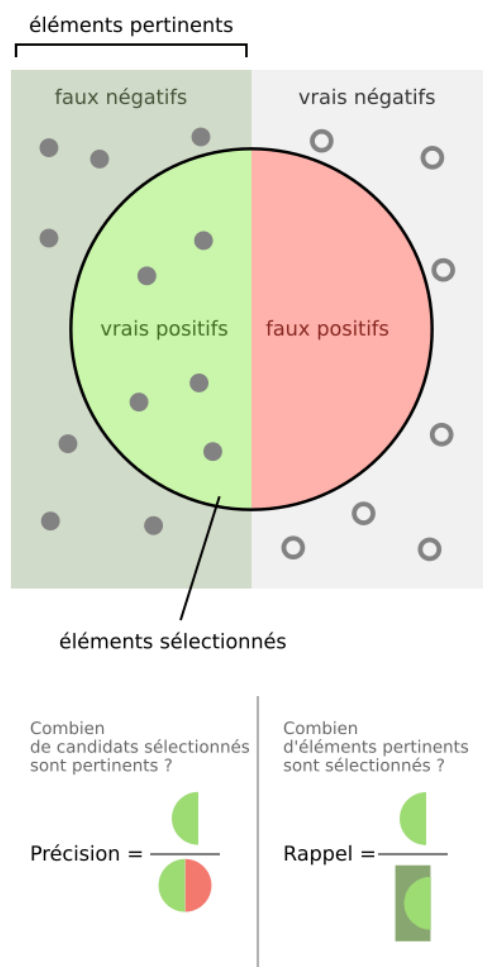


FIGURE 3.2 – Détermination des VP, FP, FN et VN et calcul de la précision et du rappel. Les éléments sélectionnés sont les détections et les éléments pertinents sont les annotations.

Illustration tirée de : https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel.

Le nombre d'insertions (Ins) est le nombre de détections qui ne correspondent à aucun événement de référence (soit le nombre de FP après le décompte des substitutions) :

$$\text{Ins} = \max(0, \text{FP} - \text{FN}). \quad (3.5)$$

Le nombre de suppressions (Sup) est le nombre d'événements de référence qui ne correspondent à aucun événement détecté (soit le nombre de FN après avoir décompté les substitutions) :

$$\text{Sup} = \max(0, \text{FN} - \text{FP}). \quad (3.6)$$

Le taux d'erreur TE est alors calculé comme le rapport du nombre d'erreurs faites par le système divisé par le nombre d'événements de référence (Ev) :

$$TE = \frac{\text{Sub} + \text{Ins} + \text{Sup}}{\text{Ev}}. \quad (3.7)$$

L'objectif est d'avoir le taux d'erreur le plus faible possible. Notons que TE est égal à 0 pour un système parfait et il peut dépasser 1 dans certains cas.

Micro-moyenne ou macro-moyenne La F-mesure ou le taux d'erreur peuvent être calculés de façon globale pour l'ensemble des classes grâce à une micro-moyenne. Prenons l'exemple de la F-mesure

$$F = \frac{2.P.R}{P + R} = \frac{VP}{VP + \frac{1}{2}(FP + FN)} \quad (3.8)$$

où VP, FP et FN représentent le nombre global de VP, FP et FN sur l'ensemble des clips. Dans ce cas, l'ensemble des exemples est traité avec la même importance, sans tenir compte des classes. Cependant, si une classe représente 90% des événements de référence, la micro-moyenne représentera majoritairement cette classe. Une approche alternative consiste à calculer la métrique pour chaque classe séparément et à prendre la moyenne ensuite (macro-moyenne) :

$$F = \frac{\sum_{c=1}^L F_c}{C} \quad (3.9)$$

où F_c est la F-mesure pour la classe c . Ceci permet d'attribuer une importance égale à toutes les classes, quel que soit le nombre d'événements de référence correspondant.

Score de détection sonore polyphonique Le *score de détection sonore polyphonique* (*polyphonic sound detection score*) (PSDS) est un score calculé à partir du nombre de VP, FP, FN proposé par Bilen et al. [2020] pour réduire les biais de la F-mesure et du taux d'erreur. Ces deux métriques prennent en entrée des valeurs binaires indiquant la présence ou l'absence d'un événement détecté (1 lorsqu'il est présent et 0 lorsqu'il est absent). Cependant, en pratique, les systèmes de détection d'événements sonores calculent des scores de sortie compris entre 0 et 1 et la présence ou l'absence d'un événement est décidée à partir d'un seuil de détection. Lorsque le score est plus élevé que le seuil, l'événement est considéré comme présent, sinon il est considéré comme absent. La variation de ce seuil peut parfois modifier de façon importante le résultat. Il est donc souvent difficile d'analyser et de comparer différents systèmes en se basant sur une valeur unique du seuil. Le PSDS contourne ce problème en se basant sur la courbe *caractéristique de performance* (*receiver operating characteristic*) (ROC) qui est la courbe du taux de VP en fonction du taux de FP. Cette courbe est obtenue en variant le seuil de détection. Le PSDS est basé sur le calcul de l'aire sous la courbe ROC. Plus l'aire est élevée plus le système est considéré comme performant. Outre l'utilisation de la courbe ROC, le PSDS offre une flexibilité importante en raison des différents paramètres impliqués dans son calcul. Les paramètres présentés ci-dessous pourraient être appliqués à la F-mesure mais ont été présentés pour la première fois pour le PSDS :

- Il est possible de pénaliser le croisement entre les FP et les FN permettant d'indiquer si l'erreur vient d'une substitution de classes qui est appelée « déclencheur de croisement ». L'ajout de cette pénalité se fait grâce à un paramètre $\alpha_{ct} \in [0, 1]$ qui est multiplié par le déclencheur de croisement avant d'être ajouté au calcul du taux de FP.
- Il est possible de pénaliser des scores trop différents entre les classes. Le calcul du taux de VP est toujours calculé par classe, puis agrégé ensuite. Un paramètre $\alpha_{st} \in [0, 1]$ multiplié par l'écart type du ratio de VP des classes est soustrait au calcul du ratio de VP et permet de gérer la pénalité appliquée lorsque l'écart de détection entre les classes est grand.

3.1.3.2 Choix de la métrique

Choix du critère Chacune des méthodes présentées ci-dessus (par segment, par événement, par intersection) permet de calculer une F-mesure, un taux d'erreur ou un PSDS. La Tâche 4 du Challenge DCASE 2017 qui avait pour but de détecter des événements en environnement urbain utilisait le taux d'erreur calculé par segments d'1 s, ce qui peut être assimilé à l'étiquetage d'événements sonores pour des clips d'1 s [Mesaros et al., 2019]. Nous ne faisons pas le choix d'une métrique calculée par segment car elle ne prendrait pas suffisamment en compte la différence temporelle entre les différents événements sonores à détecter. Par exemple, si les segments sont courts et qu'un événement apparaît dans plusieurs segments (par exemple un aspirateur), ne pas le détecter revient à le compter comme FN dans chacun de ces segments. A contrario, ne pas détecter un événement court (par exemple un bruit de plat) qui est donc présent dans un seul segment ne sera pénalisé qu'une seule fois. Si les segments sont longs, le cas est semblable à de l'étiquetage d'événements sonores qui est permissif pour les événements courts : un événement court de 500 ms dans un segment de 5 s peut être détecté durant les 5 s sans comptabiliser un seul FP. Cet exemple est illustré dans la Figure 3.3.

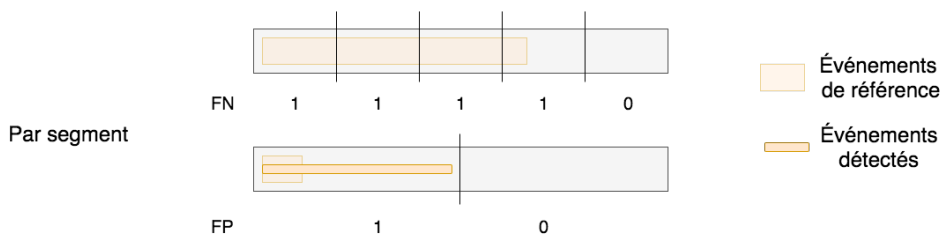


FIGURE 3.3 – Illustration du problème d'une mesure par segment.

Entre 2018 et 2020, nous avons utilisé le critère par événement qui nous semblait le plus adapté puisqu'il permet de compter les occurrences des événements et de donner plus d'importance à la segmentation. Depuis 2020, nous proposons d'utiliser également des métriques basées sur un critère par intersection. Ce critère permet un nombre d'événements détectés différent du nombre d'événements de référence et de compenser les problèmes d'annotation subjective dans la liaison ou la séparation d'événements très proches.

Métrique utilisée pour le challenge Le taux d'erreur n'est pas utilisé comme métrique pour la Tâche 4 parce que, comme son nom l'indique, il mesure les erreurs et donc pénalise les mauvaises détections mais ne récompense pas les bonnes. C'est un problème dans notre cas car, la détection d'événements sonores en environnement domestique réel étant un problème difficile et loin d'être résolu, le taux d'erreur est trop restrictif et ne permet pas de comparer les systèmes, dans la mesure où tous les systèmes ont un taux d'erreur élevé, parfois supérieur à 1. Ceci reste vrai même avec les systèmes actuels qui détectent de plus en plus de bons événements. En effet, la polyphonie étant possible, le nombre d'événements pouvant être détectés est très important par rapport au nombre d'événements présents. Prenons l'exemple concret de 100 clips audio et de 100 événements de référence. Nous faisons 100 détections et obtenons 50 VP, 50 FP et 50 donc FN, le taux d'erreur dans ce cas est de 1 (qu'il y ait des substitutions ou non ne change pas le compte) et donc équivalent à ne rien détecter alors que le système a bien détecté 50% des événements. Cet exemple ne dit pas le nombre de détections possibles mais, pour un exemple très simple de 5 segments audio par clips et 10 classes, il existe déjà 5000 détections possibles et le taux d'erreur peut rapidement atteindre des valeurs très élevées. Entre 2018 et 2020, nous avons donc choisi d'utiliser la F-mesure calculée par événement et agrégée par classe (macro-moyenne) comme métrique principale. La F-mesure permet de favoriser les bonnes détections plutôt que compter les erreurs et la macro-moyenne permet de donner une importance équivalente à chaque classe. En 2020, nous avons utilisé le PSDS comme métrique optionnelle. Depuis 2021, le PSDS est utilisé comme métrique principale avec des paramètres choisis grâce à l'analyse de son comportement sur les systèmes étudiés en 2020 [Ferroni et al., 2020].

Métrique utilisée dans la suite du manuscrit Dans ce manuscrit, la métrique principale utilisée est la F-mesure par événement, avec une tolérance de 200 ms sur l'instant de début de l'événement et le maximum entre 200 ms et 20% de la durée de l'événement comme tolérance sur l'instant de fin. La Figure 3.4 représente cette tolérance. La F-mesure est calculée par classe, puis la moyenne entre les classes est prise (macro-moyenne). Comme indiqué ci-dessus, cette métrique a été la métrique officielle pour la Tâche 4 du Challenge DCASE de 2018 à 2020.

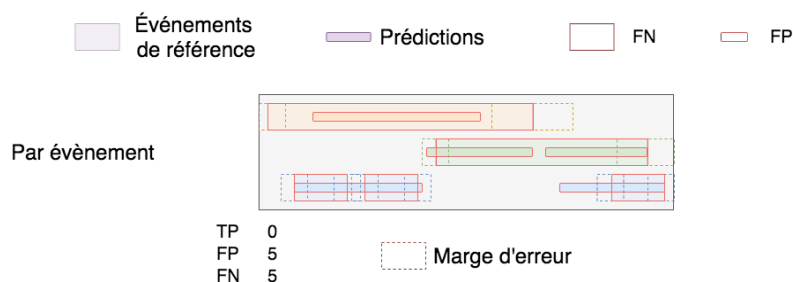


FIGURE 3.4 – Tolérance utilisée pour le calcul de la métrique de la Tâche 4.

3.2 Corpus DESED

3.2.1 Motivation

Pour promouvoir le développement de solutions de détection d'événements sonores en environnement domestique et les évaluer, on aimerait avoir un grand jeu de données fortement annotées qui contienne les classes d'intérêt correspondantes. Un tel jeu de données n'existe pas et il nous est impossible de le créer en raison du coût induit. Le seul jeu de données avec des annotations fortes en environnement domestique est TUT Sound Events 2016 [Mesaros et al., 2016b], qui a été enregistré avec des microphones intra-auriculaires majoritairement dans la cuisine. Ces spécificités ne correspondent à notre application, qui considère un microphone dont la position n'est pas nécessairement liée à une personne et qui peut se trouver dans plusieurs pièces de la maison. Nous devons donc constituer notre propre corpus pour l'évaluation et le test, constitué de données fortement annotées représentatives de la tâche et de l'application.

Des sources de données représentatives de cette application existent sur internet (Youtube, Vimeo). Elles sont parfois annotées avec des étiquettes définies par le téléverseur ou des métadonnées automatiques mais pas des annotations fortes. Une partie de ces données font partie du corpus annoté Audioset [Gemmeke et al., 2017], mais les annotations correspondantes sont faibles et d'une fiabilité très variable selon les classes.

Nous avons donc créé le corpus DESED en sélectionnant et annotant fortement une quantité suffisante de données de test et d'évaluation à partir de ces sources. Une fois ces données fortement annotées, il nous restait un budget limité pour l'annotation des données d'apprentissage. Cette situation est courante dans l'industrie pour des applications réelles. Deux choix s'offraient à nous : collecter et annoter fortement une faible quantité de données ou bien dépenser ce budget dans une quantité plus importante de données faiblement ou non-annotées. C'est ce dernier choix qui a été fait. Nous avons décidé de ne pas utiliser les annotations faibles non-fiables d'Audioset, mais plutôt d'en sélectionner un sous-ensemble, de les vérifier et les corriger. Nous n'utilisons pas les autres annotations. Ce choix permet aussi de représenter un cas d'usage pour lequel les classes d'intérêt ne sont pas disponibles dans Audioset mais de nombreuses heures d'enregistrements non-annotés sont disponibles.

3.2.2 Présentation générale

Origine des données Comme expliqué ci-dessus, les ensembles d'évaluation et de test de DESED sont constitués de données enregistrées fortement annotées, et ce sont les seules du corpus. Ces données sont issues d'Audioset pour l'ensemble de test et directement de Youtube ou de Vimeo pour l'ensemble d'évaluation. Les données de test sont parfois appelées données de développement test ou de validation et nous permettent de vérifier les performances de nos systèmes ainsi que de régler certains paramètres. Dans ce manuscrit, les données de test sont des données semblables aux données d'évaluation et permettent de tester les modèles. Ces données sont fixées pour une année donnée. Les données de validation sont différentes, elles peuvent être définies comme une partie des

données d'apprentissage qui ne seront pas utilisées dans l'apprentissage du modèle, mais aucune contrainte n'est imposée pour l'utilisation d'un tel jeu de données ou le nombre de données utilisées. Nous indiquerons notre façon de créer et d'utiliser les données de validation lorsque nécessaire dans ce manuscrit. Les données d'évaluation, sont elles utilisées uniquement pour rapporter les performances finales d'un système.

Les données d'apprentissage sont quant à elles hétérogènes et comportent quatre catégories de données.

- Des données faiblement annotées (annotations fiables) issues d'AudioSet sont disponibles en faible quantité. Celles-ci représentent un coût d'annotation faible et réaliste. La collecte et l'annotation de ces données sont discutées dans la partie 3.2.3.1.
- Des données non-annotées du domaine d'intérêt issues d'AudioSet sont disponibles en quantité importante. Ce sont des données non-annotées qui ont une probabilité élevée de contenir nos sons d'intérêt. L'intérêt des données non-annotées du domaine est de refléter un cas où des enregistrements domestiques seraient disponibles, sans l'information de ce qu'il s'est passé. Cela représente un coût très faible puisque l'acquisition de ces données est simple. Ces données peuvent tout de même apporter de l'information, notamment utile à la généralisation des systèmes. La collecte de ces données est discutée dans la partie 3.2.3.1.
- Des données non-annotées hors du domaine d'intérêt issues d'AudioSet sont disponibles en quantité très importante. Ce sont des données qui ont une probabilité faible de contenir nos classes d'intérêt. Les données qui ne sont pas du domaine sont en plus grande quantité. Depuis 2020, des données hors du domaine proviennent également du corpus *séparation de source universelle (free universal sound separation)* (FUSS) mais ce sont des événements uniques utilisés pour la génération de données synthétiques. Les données non-annotées hors du domaine sont des données enregistrées dans des conditions souvent éloignées de notre application d'intérêt. Il peut par exemple s'agir de données acquises grâce à un microphone positionné dans un parc. Celles-ci sont tout aussi simples à acquérir, mais leur utilisation au sein d'un système d'apprentissage est plus difficile. Elles sont parfois utilisées pour définir une représentation générale du son, sans savoir à l'avance si cette représentation est appropriée à notre domaine d'intérêt. La collecte de ces données est discutée dans la partie 3.2.3.1.
- Des données synthétiques fortement annotées sont disponibles depuis 2019 dans le jeu de données d'apprentissage. Elles sont composées d'événements isolés (enregistrés) extraits de Freesound combinés avec des bruits de fonds (enregistrés). Ce jeu de données est décrit plus en détail dans la partie 3.2.3.2.

Une partie des données d'apprentissage sera utilisée pour définir des jeux de données de validation utiles pour choisir les meilleurs paramètres d'un modèle d'apprentissage. L'ensemble des jeux de données sont représentés dans la Figure 3.5. Les classes d'intérêt de notre corpus sont définies dans la deuxième colonne du Tableau 3.1. Certaines de ces classes sont plus spécifiques à une pièce particulière alors que d'autres sont susceptibles

d'apparaître dans de nombreux endroits de la maison comme représenté dans la troisième colonne du Tableau 3.1.

Chronologie En 2018, la première version du corpus **DESED** ne comportait que des données Audioset : les données faiblement annotées, les données non-annotées du domaine et hors domaine, les données de test et les données d'évaluation. En 2019, les ensembles de test et d'évaluation de 2018 ont été groupés pour former l'ensemble de test de 2019 (qui est l'ensemble de test actuel), les données non-annotées hors domaine issues d'Audioset ont été mises de côté, un nouveau jeu de données d'évaluation issu de Youtube (qui sera rendu public) et de Vimeo (qui restera privé) est apparu et les données synthétiques du domaine ont été ajoutées. En 2020, le jeu de données **FUSS** a été ajouté.

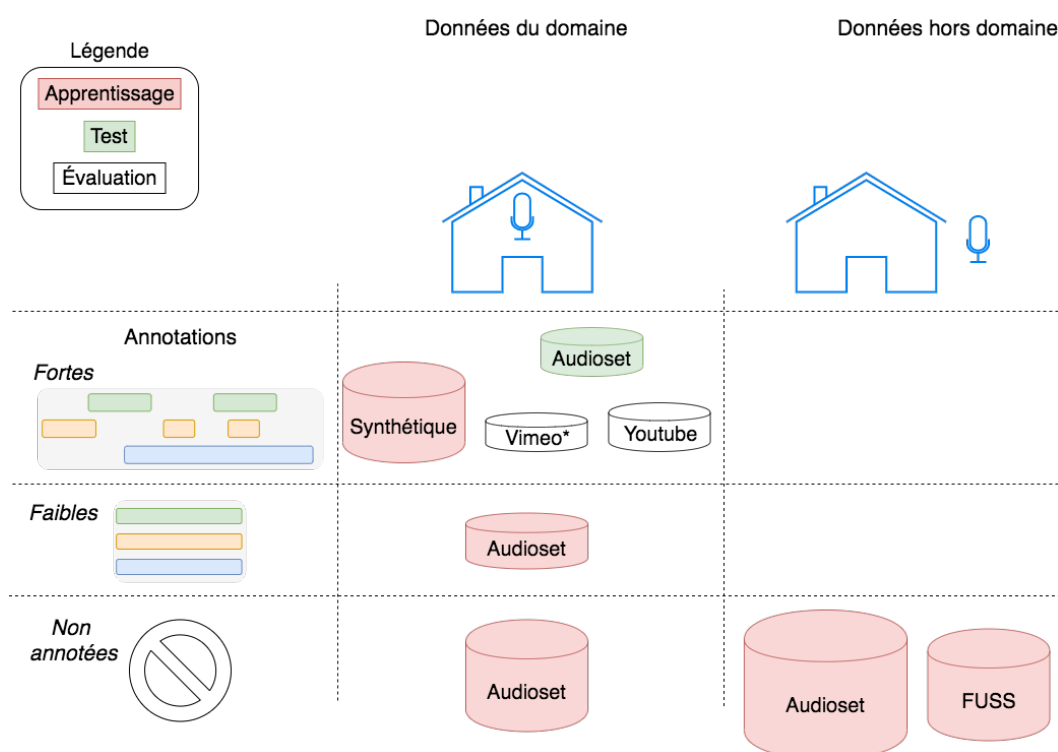


FIGURE 3.5 – Jeux de données à disposition des participants de la tâche. La taille des blocs indique la différence de taille entre les jeux de données. Les corpus d'origine des données sont indiqués dans les blocs.

3.2.3 Collecte et annotation des données

Dans cette partie, nous décrivons le processus de collecte et d'annotations des données qui a été utilisé pour les différents jeux de données composant **DESED**. Nous expliquons les choix faits et leur influence sur la tâche.

3.2.3.1 Données issues de corpus de vidéos en ligne

Collecte des données Audioset Les données issues du corpus Audioset [Gemmeke et al., 2017] sont des clips audio de 10 s extraits de vidéos Youtube. Pour sélectionner les données issues d’Audioset, il nous faut identifier une liste d’événements d’intérêt au sein de l’ontologie Audioset, qui est une ontologie hiérarchique contenant 635 classes, car les 10 classes que nous avons retenues ne sont pas directement présentes dans cette ontologie.

Audioset	DCASE	Pièce courante
Réveil	Alarme / sonnette / sonnerie	Chambre
Téléphone, sonnerie		Salle à manger / Salon
Sonnerie porte		
Blender, robot ménager	Blender	Cuisine
Chat	Chat	Divers
Plats, casseroles, poêles	Plats de cuisine	Cuisine
Couverts, argenterie		
Chien	Chien	Divers
Tondeuse électrique, rasoir électrique	Tondeuse / brosse à dent électrique	Salle de bain
Brosse à dent électrique		
Friture (nourriture)	Friture	Cuisine
Évier (remplir ou laver)	Eau qui coule	Cuisine
Eau du robinet, robinet		Cuisine / salle de bain
Chasse d’eau		Toilettes
Voix masculine		Voix
Voix féminine		
Voix d’enfant		
Conversation		
Aspirateur	Aspirateur	Divers

TABLEAU 3.1 – Liste des classes d’événements d’intérêt. La colonne de gauche représente les classes présentes dans l’ontologie Audioset, la colonne du milieu les noms donnés pour la Tâche 4 du Challenge DCASE et la colonne de droite la pièce la plus courante pour ce type d’événement.

Dans le Tableau 3.1 nous présentons nos classes d’intérêt ainsi que les classes d’Audioset dont elles sont issues et les pièces de la maison dans lesquelles on peut généralement les entendre. Les classes sélectionnées dans Audioset ne sont pas toutes au dernier niveau de l’ontologie, par exemple la classe « chat » d’Audioset contient les classes « ronronnement », « miaulement », *etc.*. Comme on peut le constater dans le tableau, nos 10 classes d’intérêt sont généralement issues du regroupement de plusieurs classes d’Audioset. Nous avons effectué ce regroupement pour deux raisons :

- certaines classes Audioset représentent des concepts trop spécifiques pour notre application, par exemple « voix d’enfant » ;

- certaines classes Audioset ont trop peu de données disponibles, par exemple « sonnette de porte ».

Nous visons une quantité de données équilibrée entre ces 10 classes. Tirer les clips de façon aléatoire parmi ceux contenant ces classes entraînerait un déséquilibre important : la moitié des données ne contiendraient que la classe « voix » qui est abondante dans Audioset, et certaines classes peu fréquentes dans Audioset comme « tondeuse/brosse à dent électrique » ne seraient peut-être même pas représentées. Pour éviter cela, nous tirons le même nombre de clips contenant chacune des 10 classes, sans tenir compte des autres classes présentes. Afin de réduire encore la sur-représentation de la classe « voix », nous ne tirons aucun clip audio spécifiquement pour cette classe. Cela signifie que les événements de la classe « voix » apparaissent dans le corpus, mais seulement dans des clips audio qui ont été tirés pour une autre classe d'intérêt. Autrement dit, les événements « voix » présents dans un clip ne sont jamais la seule classe d'intérêt dans ce clip.

Concernant le jeu de données d'apprentissage, étant donnée l'hétérogénéité de la qualité des annotations faibles entre les différentes classes d'Audioset, nous avons tiré aléatoirement jusqu'à 2000 clips audio par classe lorsque c'était possible pour permettre la vérification et l'annotation forte de 150 clips par classe d'intérêt. Les clips audio non écoutés ont été transférés dans la partie non-annotée du corpus (dans le domaine).

Concernant le jeu de données de test, les données issues d'Audioset proviennent de la partie « évaluation » d'Audioset pour éviter d'avoir des enregistrements de test provenant de vidéos similaires aux enregistrements d'apprentissage. Le but est d'annoter 100 clips audio par classe d'intérêt (hors « voix ») pour le jeu de données de test.

Le Tableau 3.2 indique le nombre de clips d'apprentissage et de test qui contiennent au moins l'une des classes d'intérêt, le nombre d'événements correspondants et leur durée.

Classe	Nombre			Durée des évts. (s)	
	Appren- tissage clips	Test		Test	
		clips	évts.	moyenne	médiane
Alarme/sonnette/sonnerie	205	187	420	1,96	1,03
Blender	134	80	94	5,23	4,70
Chat	173	121	341	1,38	0,85
Plats de cuisine	184	171	559	0,63	0,39
Chien	214	160	570	1,41	0,77
Rasoir / brosse à dent électrique	103	62	65	7,73	9,90
Friture	171	89	94	8,26	10,00
Eau qui coule	343	197	237	5,25	4,85
Voix	550	627	1752	1,50	1,15
Aspirateur	167	91	92	8,48	10,0
Total	2244	1785	4224	2,10	1,06

TABLEAU 3.2 – Statistiques par classe du corpus [DESED](#).

Dans la colonne Apprentissage, on constate que les classes « blender » et « rasoir/brosse à dent électrique » ont moins de 150 clips. Cela est dû à la faible quantité de données disponible pour ces classes dans Audioset et au manque de fiabilité des annotations (il existe probablement d'autres clips audio contenant ces classes mais qui ne sont pas annotés avec ces classes). Les classes représentées en plus grande quantité apparaissent régulièrement dans des clips audio contenant une autre classe d'intérêt. Par exemple l'« eau qui coule » apparaît fréquemment avec les « plats de cuisine » et peut apparaître dans les mêmes clips que « friture » ou « blender ». La « voix » apparaît dans 550 clips parmi les 1578 clips d'apprentissage et dans 627 clips parmi les 1168 clips de test, ce qui constitue plus d'un tiers des clips alors qu'aucun clip sélectionné ne contient cette classe uniquement. Le jeu de données est donc biaisé dans une certaine mesure. Toutefois, cela reflète la réalité des données Audioset et l'approche de sélection proposée a permis de limiter ce biais qui aurait conduit à un jeu de données largement dominé par la voix.

Annotations forte et faible des données Audioset Les données issues d'Audioset ont été annotées par les 4 organisateurs de la Tâche 4 en 2018, qui sont des chercheurs dans le domaine de l'analyse des sons ambiants (dont je fais partie). Pour la première phase d'annotation, chaque annotateur a annoté (vérifié et corrigé dans le cas des annotations faibles) un quart des clips. Il doit annoter la présence exhaustive de toutes les classes d'intérêt. Pour la deuxième phase d'annotation, les annotateurs deviennent des « vérificateurs », c'est à dire qu'ils vérifient les annotations faites par un autre annotateur. Lorsque le vérificateur confirme l'annotation du premier annotateur, celle-ci est validée et utilisée dans le corpus. Lorsque le vérificateur a un doute ou a dû faire un changement dans l'annotation un troisième annotateur est consulté. Au vu des 3 annotations, une décision collégiale est prise pour décider de l'annotation du clip audio. S'il est impossible de parvenir à un consensus, le clip audio n'est pas gardé dans la partie du corpus annoté en raison de sa trop grande ambiguïté. Ces ambiguïtés interviennent le plus souvent quand la vidéo n'est pas disponible pour vérifier la potentielle présence d'un événement d'intérêt.

Cette tâche d'annotation nous a permis d'identifier la difficulté de la tâche de détection d'événements sonores. Des exemples d'ambiguïtés fréquentes lors de l'annotation sont les ambiguïtés entre un chat qui miaule et un bébé qui pleure, entre de la friture et un robinet qui coule, ou encore la perception d'un rire comme voix ou bien la détermination d'un seuil permettant d'identifier un son comme événement sonore ou comme faisant partie du bruit de fond. L'annotation forte des données a utilisé le même principe que la phase d'annotation faible, excepté que dans ce cas l'annotation implique de déterminer les instants de début et de fin de chacun des événements. Le principe de vérification reste le même mais dans ce cas les ambiguïtés sur les instants de début et de fin sont plus grandes donc le nombre d'annotations qui nécessitent la vue de 3 annotateurs et une décision collégiale est beaucoup plus important. L'Algorithme 3.1 décrit le processus d'annotation utilisé.

Le temps d'annotation varie en fonction des annotations souhaitées. Nous avons estimé le temps nécessaire pour produire une annotation faible entre une à deux fois la durée

Algorithme 3.1: Pseudo-algorithme d'annotation des données.

```

Result: Annotations des données
 $N \leftarrow$  nombre de clips à annoter ;
 $N_A \leftarrow$  nombre d'annoteurs ( $N_A > 2$ );
 $A \leftarrow$  ensemble des annoteurs ;
for  $i = 1, 2, \dots, N_A$  do
    for  $u = 1, 2, \dots, \frac{N}{N_A}$  do
         $a_u \leftarrow$  clip audio d'indice  $u$  ;
         $Ann_{i,u} \leftarrow$  Annotation de  $a_u$  par Annotateur  $i$  ;
         $j \in \llbracket 1, N_A \rrbracket, j \neq i$  ;
         $Ann_{j,u} \leftarrow$  Vérification de l'annotation  $Ann_{i,u}$  par Annotateur  $j$  ;
        if  $Ann_{i,u} \neq Ann_{j,u}$  then
             $k \in \llbracket 1, N_A \rrbracket, k \notin \{i, j\}$  ;
             $Ann_{k,u} \leftarrow$  Vérification de l'annotation  $Ann_{i,u}$  et  $Ann_{j,u}$  par
                Annotateur  $k$  ;
             $Ann_u \leftarrow$  Décision collégiale des annoteurs de  $a_u$  ;
            if  $Ann_u$  n'est pas définie then
                 $a_u$  est exclu du jeu de données annotées ;
            end
        end
        else
             $Ann_u \leftarrow Ann_{i,u}$ 
        end
    end
end

```

du clip audio par annoteur. La vérification prend le temps du clip audio. Le temps nécessaire pour les annotations fortes était quant à lui de l'ordre de 1,5 à 10 fois la durée du clip audio par annoteur. La vérification varie entre 1 à 3 fois le temps du clip audio en fonction de la complexité du clip audio (polyphonie, ambiguïtés des instants de début et fin). Ces chiffres ne sont que des estimations approximatives effectuées une fois les annotations complétées. Les durées ont été calculées à partir du nombre d'heures journalières nécessaires à annoter les clips lors de la première phase d'annotation. Il est à noter que ces annotations ne comportent que 10 classes d'événements à annoter et non pas l'ensemble des classes d'événements sonores qui peuvent être présentes dans un clip audio, ce qui rendrait la tâche d'annotation encore plus difficile.

Collecte et annotation des données Youtube et Vimeo Les données issues de Youtube et Vimeo font partie de l'ensemble d'évaluation utilisé depuis 2019. Ces données ont été collectées et annotées à la main. Les données sont recueillies grâce à des requêtes simples (classes d'intérêt ou synonymes) dans Youtube ou Vimeo parmi les vidéos sous licence

Creative Commons CC-BY¹. La personne en charge de la collecte² doit vérifier que la vidéo contienne la classe d'intérêt et donner un instant du clip audio où l'événement apparaît. Les 10 secondes sont ensuite déterminées de façon aléatoire autour de cet instant afin d'obtenir des événements présents en totalité ou de manière partielle dans le clip de manière similaire à ce qui se produit dans Audioset. La phase d'annotation est basée sur le protocole utilisé ci-dessus pour les annotations fortes d'Audioset mais réalisée par seulement 3 annotateurs, dont 2 en commun avec la première phase (dont je fais partie).

3.2.3.2 Données synthétiques

Collecte des données Freesound Afin de collecter les données issues de Freesound, nous utilisons une partie des données déjà annotées sur la plateforme Freesound Annotator^{3, 4}. L'autre partie est extraite directement depuis Freesound. L'écoute de quelques clips audio Freesound tirés aléatoirement parmi les clips contenant nos classes d'intérêt nous a appris plusieurs points importants :

- les données Freesound sont très différentes des données Youtube/Vimeo ;
- la majorité des clips audio contenant nos classes d'intérêt contiennent uniquement notre classe d'intérêt (pas d'autres événements sonores) ;
- les événements sonores sont souvent entourés de silence ou les clips sont composés de plusieurs occurrences du même événement séparées par du silence (exemple : « aboiement » « silence » « aboiement ») ;
- le rapport signal-à-bruit est souvent très élevé (peu de bruit de fond) et la qualité d'enregistrement est bonne (bonnes conditions d'enregistrement).

Après la collecte des données, une phase d'annotation manuelle est nécessaire pour identifier les enregistrements qui contiennent effectivement nos classes d'intérêt et qui n'ont pas ou peu de bruit de fond pour pouvoir être considérés comme événements isolés.

Annotation des données Freesound Afin d'isoler les événements sonores, les données Freesound nécessitent un travail de vérification et de segmentation. Le travail de vérification permet de vérifier si la classe d'événement sonore est bien la seule présente dans le clip audio et si le bruit de fond est suffisamment faible. Ce travail a été fait manuellement. La segmentation permet de supprimer le silence autour des événements de façon semi-automatique par une approche basée sur la puissance du signal. Les clips audio collectés depuis Freesound ayant un bruit de fond faible, cette approche a été jugée comme suffisante. Nous définissons la puissance moyenne d'un clip de taille T par

$$e_{\text{tot}} = \frac{\|\mathbf{a}\|^2}{T} \quad (3.10)$$

1. <https://creativecommons.org/licenses/by/4.0/>

2. Ce travail a été réalisé par Pierre Goncalves que nous remercions pour sa contribution.

3. <https://annotator.freesound.org>

4. Un remerciement particulier à Xavier Favory pour son aide lors de l'accès aux données annotées.

où \mathbf{a} est un vecteur représentant le clip. Ensuite, nous calculons la puissance locale :

$$e_{\text{loc}}(t) = \frac{\|\mathbf{a}_{t-\tau:t+\tau}\|^2}{2\tau}, \forall t \in \llbracket 1, T \rrbracket \quad (3.11)$$

où 2τ représente la durée de la fenêtre temporelle choisie. Nous définissons le rapport de puissance local par

$$r(t, \Theta) = \frac{e_{\text{loc}}(t)}{\Theta \times e_{\text{tot}}} \quad (3.12)$$

où Θ est le seuil d'activation qui définit la tolérance au bruit de fond. Le clip audio segmenté \mathbf{a}_{seg} est finalement défini par

$$\mathbf{a}_{\text{seg}} = \{a_t | r(t, \Theta) > 0, \forall t \in \llbracket 1, T \rrbracket\}. \quad (3.13)$$

Le seuil Θ est fixe pour l'ensemble d'un clip audio. Sa valeur est égale à 0,001 par défaut, mais l'annotateur peut l'ajuster en fonction des enregistrements si nécessaire. Ce principe simple nous a permis de segmenter plus de 1000 clips audio Freesound afin d'obtenir des événements « isolés », qui contiennent tout de même parfois un léger bruit de fond. Ces événements isolés servent de base à la génération des clips sonores avec annotations fortes. Le jeu de données [FUSS](#) contient aussi des événements cibles qui peuvent permettre la génération de nos données. Bien que ceux-ci soient vérifiés pour ne contenir qu'un événement et peu de bruit de fond, ils ne sont pas segmentés et ne sont donc pas utilisés en tant qu'événements cibles pour générer des clips fortement annotés. Ils sont notamment utilisés pour générer des données synthétiques utiles à la séparation de sources ou en tant qu'événements non-cibles.

Données de bruit de fond pour les clips synthétiques Pour les jeux de données synthétiques d'apprentissage de [DESED](#), nous avons utilisé les données de bruit de fond des corpus *interfaçage du son en essaim (Sound INterfacing through the Swarm)* ([SINS](#)) [[Dekkers et al., 2017](#)] et TUT Acoustic Scenes 2017 [[Mesaros et al., 2016b](#)]. Nous ne sélectionnons que les clips audio annotés « autre » de [SINS](#) et ceux annotés « maison », « bureau » ou « librairie » de TUT Acoustic Scenes 2017, qui sont les plus proches de notre application. Pour le jeu de données synthétiques d'évaluation, nous avons collecté quelques enregistrements de bruit de fond issus de vidéos Youtube. Ces enregistrements sont des enregistrements en milieu domestique de longue durée pour lesquels il y a très peu ou pas d'activité et qui ne contiennent donc pas nos classes d'intérêt.

Enfin, pour permettre de générer des bruits de fond plus réalistes, des événements non-cibles provenant du corpus [FUSS](#) ont été utilisés. [FUSS](#) est un corpus où les événements sonores ont été vérifiés et ne contiennent que des événements isolés, mais ceux-ci ne sont pas segmentés. La segmentation des données n'est pas un problème dans notre cas puisque nous n'essayons pas de détecter ces événements, ils sont simplement ajoutés pour enrichir le bruit de fond. [FUSS](#) ne contient pas directement les annotations, mais les clips audio sont issus de FSD50K, il est donc possible de récupérer celles-ci, ce qui nous permet de filtrer les classes que nous souhaitons utiliser. Les jeux de données d'apprentissage et d'évaluation de [FUSS](#) sont utilisés dans les jeux correspondants de [DESED](#).

3.2.4 Génération des données synthétiques

Depuis 2019, le corpus [DESED](#) contient des données synthétiques. Ces données synthétiques permettent d'obtenir à coût modéré des données fortement annotées au sein du jeu d'apprentissage. L'un des enjeux de la génération des données synthétiques est de reproduire des clips audio réalistes. Si les clips audio ne sont pas réalistes, il existera une importante différence de domaine entre les données synthétiques utilisées à l'apprentissage et les données d'évaluation qui sont enregistrées en conditions réelles. C'est un problème qu'il sera donc nécessaire de gérer lors de l'apprentissage des systèmes. En outre, les données synthétiques peuvent être utiles à l'analyse des différents problèmes présents dans Audioset (chevauchement d'événements, volume du bruit de fond, position de l'événement, taille des clips, *etc.*) qu'il est très difficile d'isoler sans l'utilisation de données synthétiques.

Algorithme 3.2: Pseudo-algorithme de génération de clips audio

Result: Clip audio généré
 $D \leftarrow$ durée du clip audio désirée (1) ;
 $B \leftarrow$ Choisir aléatoirement un bruit de fond (2) ;
if $B < D$ **then**
 | $B \leftarrow$ Répéter B jusqu'à obtenir la durée du clip audio ;
end
 $B \leftarrow$ Appliquer une réponse impulsionnelle de salle (*optionnel*) (7) ;
 $S \leftarrow B$;
 $\mathcal{F} \leftarrow$ Choisir les événements sonores à apparaître en fonction d'une distribution déterminée au préalable (3) ;
for F dans \mathcal{F} **do**
 | $F \leftarrow$ placer l'événement et tronquer F à la durée désirée (4) ;
 | $F \leftarrow$ augmenter F (transposition du ton) (*optionnel*) (5) ;
 | $F \leftarrow$ changer l'amplitude de F pour la mettre à l'échelle du rapport signal-à-bruit désiré (6) ;
 | $F \leftarrow$ ajouter les fondus d'apparition et de fermeture (10 ms) ;
 | $F \leftarrow$ Appliquer une réponse impulsionnelle de salle (*optionnel*) (7) ;
 | $S \leftarrow S + F$;
end

L'Algorithme 3.2 présente le principe général de génération des clips audio synthétiques. Le bruit de fond est tout d'abord défini, puis les événements sonores à apparaître sont tirés aléatoirement selon une distribution de co-occurrence ou non. Notons que les événements sonores à apparaître peuvent être des événements de nos classes d'intérêt ou non. Nous avons généré plusieurs jeux de données synthétiques en faisant varier les paramètres de l'Algorithme 3.2.

Paramètres de génération utilisés Les différents paramètres utilisés sont (*cf.* numéros de (1) à (7) dans l'Algorithme 3.2) :

- (1) Durée des clips : 10 s ou 60 s.
- (2) Bruit de fond choisi parmi : banque de bruit de fond, les événements longs cibles ou des événements non-cibles.
- (3) Choix des événements sonores de premier plan : basé sur des distributions de co-occurrence ou un seul événement ou choix parmi les événements courts seulement (lorsque bruit de fond parmi les événements longs).
- (4) Positionnement de l'événement : spécifique ou aléatoire.
- (5) Augmentation : avec ou sans changement de ton.
- (6) Changement du rapport signal-à-bruit (multiples niveaux)
- (7) Réverbération : avec ou sans.

La majorité de ces paramètres sont explicites et seront donnés lors de la discussion des différents jeux de données synthétiques utilisés dans le chapitre 4. Ici, nous nous attardons sur le choix des événements sonores de premier plan basé sur une distribution de co-occurrence des événements. Lorsque nous utilisons la distribution de co-occurrence des événements, notre but est de générer des données synthétiques avec une distribution d'événements proche de celle des données d'évaluation. Par exemple, pour définir la co-occurrence dans notre jeu synthétique d'apprentissage, nous utilisons le jeu de données de test (fortement annoté) pour estimer une probabilité de co-occurrence des différents événements. Afin de définir la co-occurrence des événements, nous utilisons un principe simple : pour chaque classe, nous définissons la distribution des classes qui apparaissent en même temps que cette classe dans le jeu de test avec leur probabilité d'apparition. La « co-occurrence » est définie comme le nombre d'événements de la classe d'intérêt présents dans les clips audio divisé par le nombre total d'événements de cette classe. Pour la génération des données, nous reprenons les principes utilisés lors de l'acquisition des données issues d'AudioSet. Nous créons donc un clip audio à partir d'un événement « principal » puis nous ajoutons des événements « co-occurents ». La distribution des classes est uniforme pour les événements principaux (hors « voix ») alors que la sélection des événements « co-occurents » se base sur la probabilité de co-occurrence. Cette approche permet d'obtenir une distribution de classes proche de celle du jeu réel. Ce principe de co-occurrence est utilisé dans la génération des données d'apprentissage, de validation ainsi que certains jeux d'évaluation.

Données d'apprentissage et validation Les données de validation sont issues des données d'apprentissage. Les données de [FUSS](#) ont déjà un ensemble de validation défini. Dans [DESED](#), les données de validation en 2018 représentent 20% des données faiblement annotées. Après 2018, les données de validation représentent 10% des données enregistrées faiblement annotées et 10% des données synthétiques. Cependant, notre façon de choisir les données de validation synthétiques change entre 2019 et 2020. En 2019, nous avons généré des clips audio à partir des sons isolés de l'ensemble d'apprentissage [DESED](#) puis

nous avons séparé cet ensemble en deux sous-ensembles d'apprentissage et de validation. En 2020, nous avons séparé les événements sonores isolés et les bruits de fond en un jeu d'apprentissage (90%) et un jeu de validation (10%). Les événements sonores isolés ont été choisis en s'assurant que les utilisateurs Freesound qui ont téléversés les données soient représentés uniquement dans l'un des deux jeux (apprentissage ou validation). Il a été vérifié que les bruits de fond d'apprentissage et de validation issus du corpus SINS proviennent de scènes sonores différentes (TUT Acoustic Scenes 2017 ne nécessite pas cette vérification). Cette modification permet d'éviter le sur-apprentissage des systèmes envers les données synthétiques. En effet, en 2019, les événements isolés et les bruits de fond pouvant se retrouver à la fois dans les ensembles d'apprentissage et de validation, un système pouvait se spécialiser à reconnaître ces événements isolés et obtenir de bonnes performances sur l'ensemble de validation (mais pas celui d'évaluation). En 2020, les systèmes nécessitent une généralisation à de nouveaux événements isolés pour obtenir de bonnes performances sur le jeu de validation, ce qui est le comportement souhaité. Sauf indication contraire, les données d'apprentissage et de validation ont utilisé un algorithme de génération similaire chaque année.

3.2.5 Résumé

Le Tableau 3.3 résume les jeux données disponibles qui composent le corpus DESED. Les jeux de données regroupés dans la banque de données permettent la génération des données synthétiques. L'ensemble de ces données ainsi que les scripts de génération des données synthétiques sont disponibles ^{5, 6}. La Figure 3.6 illustre la façon dont les différents jeux de données initiaux sont utilisés et combinés pour créer les jeux d'apprentissage et de test de la tâche. L'ensemble des propriétés du corpus DESED sont résumées ici :

- une ontologie est disponible ;
- les données sont enregistrées ou bien synthétiques ;
- les données sont non-annotées, faiblement annotées ou fortement annotées ;
- les données enregistrées et fortement annotées sont exclusives au test et à l'évaluation (problème non-supervisé ou faiblement supervisé) ;
- les données d'apprentissage fortement annotées sont synthétiques et sont une association de bruit de fond et d'événements isolés ;
- les annotations comprennent les classes d'intérêt uniquement ;
- les données sont sélectionnées de façon équilibrée pour chacune des classes d'événements (hors « voix ») et les déséquilibres proviennent de la co-occurrence des événements ;
- les données faiblement annotées sont sélectionnées depuis Audioset (données vérifiées) ;
- les annotations originales d'Audioset des données non vérifiées sont supprimées pour la création du jeu de données non-annotées ;

5. <https://github.com/turpaultn/DESED>

6. <https://project.inria.fr/desed/>

		Annotation	Origine	Nombre de clips	
Corpus données		Apprentissage	Non-annoté	Audioset	14412
		Apprentissage	Faible	Audioset	1578*
		Apprentissage	Forte	Synthétique	2019 : 2045* 2020 : 2584*
		Test (test 2018, eval 2018)	Forte	Audioset	1168 (288, 880)
		Évaluation publique	Forte	Youtube	692
		Évaluation privée	Forte	Vimeo	352
		Évaluation synth	Forte	Synthétique	2019 : 12139 2020 : 11612
Banque de données	Bruit de fond	Apprentissage	Non-annoté	SINS	2060*
		Apprentissage	Non-annoté	TUT	846*
		Évaluation	Non-annoté	Youtube Freesound	5 12
	Événements	Apprentissage cible	Événements isolés	Freesound	1009*
		Apprentissage cible et non-cible	Événements isolés	FUSS	7237
		Validation cible et non-cible	Événements isolés	FUSS	2883
		Évaluation cible	Événements isolés	Freesound	314
		Évaluation cible et non-cible	Événements isolés	FUSS	2257

TABLEAU 3.3 – Jeux de données composant le corpus **DESED**. * indiquent qu'une partie des données est utilisée pour définir le jeu de validation.

- les données non-annotées dans le domaine sont des données dont l'annotation Audioset contenait une de nos classes d'intérêt ;
- les données non-annotées hors domaine sont des données dont l'annotation Audioset ne contenait pas une de nos classes d'intérêt.

3.3 Évaluation officielle de la Tâche 4

Chaque année la Tâche 4 du challenge a reçu des soumissions de multiples participants du monde entier et un classement général des systèmes est proposé. Le Tableau 3.4 indique le nombre de participants et d'équipes ayant soumis, ainsi que le nombre total de soumissions reçues (jusqu'à 4 soumissions par équipe sont possibles). Les systèmes des participants sont comparés entre eux et avec nos systèmes de référence. Ces derniers sont décrits plus en détail et analysés dans la partie 4.1. Dans cette partie, nous discutons

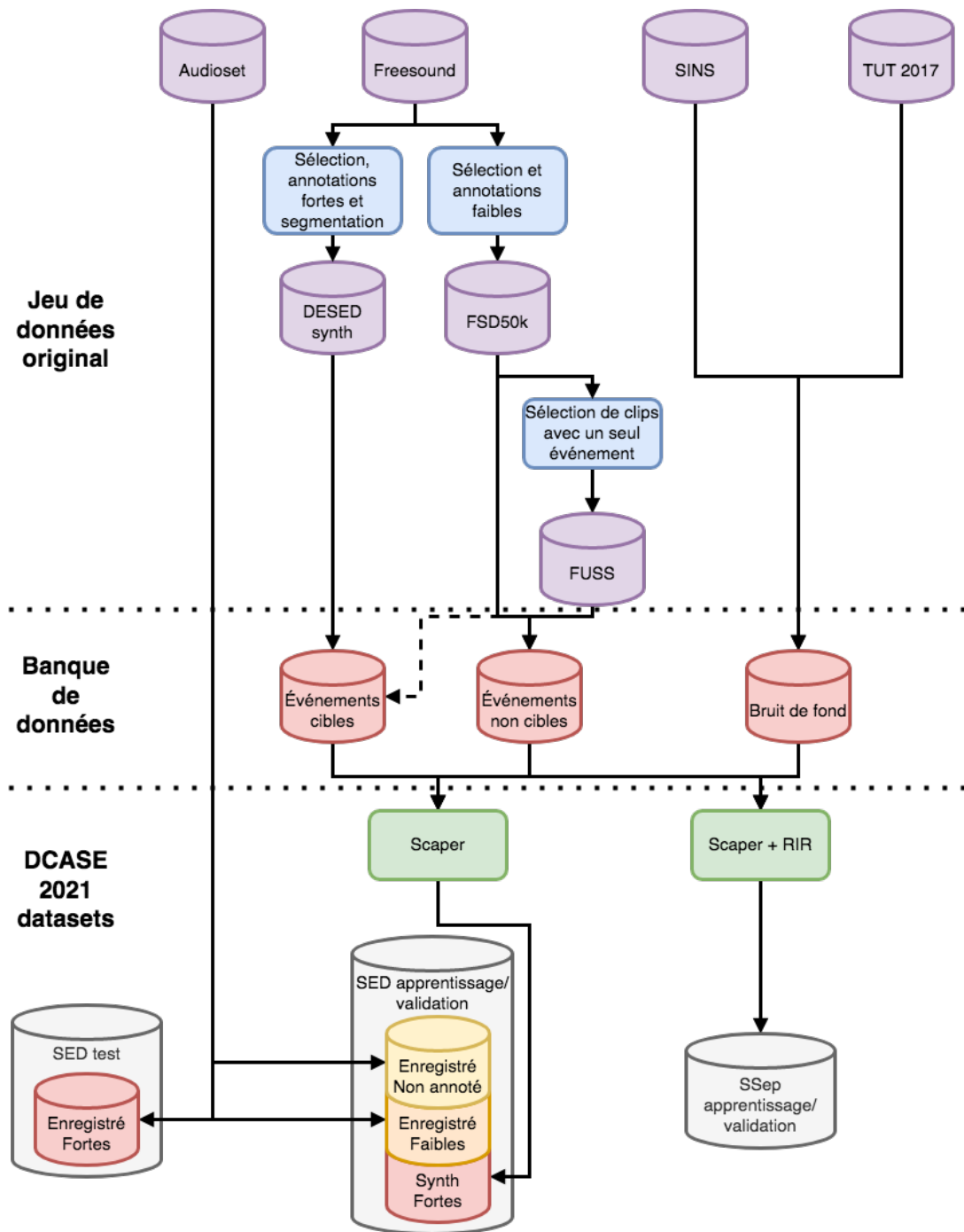


FIGURE 3.6 – Utilisation des différents jeux de données d’apprentissage et de test. La flèche en tirets représente une utilisation possible des données **FUSS** qui n’est pas encore exploitée dans le corpus officiel.

des résultats généraux et présentons les systèmes soumis que nous considérons comme particulièrement intéressants.

Année	Nb participants	Nb équipes	Nb soumissions
2018	57	16	50
2019	60	18	57
2020	90	21	73

TABLEAU 3.4 – Nombre de participants, d'équipes et de soumissions à la Tâche 4.

3.3.1 DCASE 2018

Rang	Système	Classifieur	Nb Params	F (%)
1	jiakai_psh	CRNN	1M	32,4
2	liu_ustc	CRNN, Capsule-RNN	4M	29,9
3	kong_surrey	VGGish 8 layer CNN	4M	24,0
4	kothinti_jhu	CRNN, RBM, cRBM, PCA	1M	22,4
5	harb_tug	CRNN, VAT	497k	21,6
6	koutini_jku	CRNN	126k	21,5
7	guo_thu	CRNN multi-échelle	970k	21,3
8	hou_bupt	CRNN	1M	21,1
9	lim_etri	CRNN	239k	20,4
10	avdeeva_itmo	CRNN, CNN	200k	20,1
11	wangjun_bupt	RNN	1M	17,9
12	pellegrini_irit	CNN, CRNN avec MIL	200k	16,6
13	moon_yonsei	RseNet, SENet	10M	15,9
14	dinkel_sjtu	CRNN, HMM-GMM	126k	13,4
15	wang_nudt	CRNN	24M	12,6
	baseline 2018	CRNN	126k	10,8
16	raj_iit	CRNN	215k	9,4

TABLEAU 3.5 – F-mesure par événement des systèmes soumis à la Tâche 4 du Challenge DCASE 2018 (sur l'ensemble d'évaluation).

La Tâche 4 du Challenge DCASE 2018 a enregistré 50 soumissions par 16 équipes différentes impliquant un total de 57 chercheurs. Le Tableau 3.5 présente les caractéristiques des systèmes soumis ainsi que leur performance pour le jeu de données d'évaluation 2018. Le meilleur système de JiaKai [2018] (**jiakai_psh**) utilise une approche de « professeur moyen » qui exploite les données non-annotées pour régulariser le classifieur appris sur les données faiblement annotées. Le deuxième système de Liu et al. [2018] (**liu_ustc**) s'appuie sur une détection d'activité sonore basée sur l'énergie comme pré-traitement d'un « réseau capsule » (détecteurs locaux agrégés par un RNN). D'autres soumissions sont notables comme le système de Kothinti et al. [2018] (**kothinti_jhu**) basé sur un

pré-traitement par machines de Boltzmann restreintes, ce qui permet une bonne détection des instants de début des événements. [Dinkel et al. \[2018\]](#) (`dinkel_stju`) proposent un système utilisant des modèles de mélange de gaussiennes et des modèles de Markov cachés pour détecter les instants d'activation des événements sonores. [Cances et al. \[2018\]](#) (`pellegrini_irit`) effectuent un post-traitement utilisant l'apprentissage multi-instance pour exploiter les annotations faibles de façon efficace. Les systèmes de [Kothinti et al. \[2018\]](#), [Dinkel et al. \[2018\]](#) et [Cances et al. \[2018\]](#) obtiennent de bonnes performances de segmentation des événements mais souffrent de mauvaises performances de classification. [Kong et al. \[2018\]](#) proposent un système commun à différentes tâches qui repose sur une architecture de CNN. Parmi les autres systèmes, plusieurs équipes ont proposé des améliorations de l'algorithme en deux passes qui servait de référence [[Harb and Pernkopf, 2018](#), [Hyeonggi et al., 2018](#), [Koutini et al., 2018](#)]. Plusieurs équipes ont aussi proposé d'utiliser différentes représentations de bas niveau des données [[Kothinti et al., 2018](#), [Raj et al., 2018](#), [Wang et al., 2018](#)], des augmentations de données [[Avdeeva and Agafonov, 2018](#), [Hyeonggi et al., 2018](#), [Lim et al., 2018](#), [Wang et al., 2018](#), [Wang and Li, 2018](#)], différentes échelles de temps en entrée de leur modèle [[Guo et al., 2018](#), [Lim et al., 2018](#)] ou bien une agrégation particulière des détections fortes pour obtenir les détections au niveau du clip [[Hou and Li, 2018](#)]. Une analyse plus détaillée liée à la segmentation est discutée dans la partie 4.2.1.

Le résultat important de cette analyse concerne les mauvaises performances générales en terme de segmentation (le meilleur système a une F-mesure de segmentation de 37,02%). Cet aspect semble être un élément critique à l'amélioration des performances. C'est cette analyse qui nous a notamment poussés à proposer l'utilisation d'un jeu de données synthétiques, dont le but est d'utiliser des données fortement annotées dans l'apprentissage (même si le domaine peut être éloigné) afin d'aider à la segmentation.

3.3.2 DCASE 2019

La Tâche 4 du Challenge DCASE 2019 a recueilli 57 soumissions de 18 équipes différentes impliquant un total de 60 chercheurs. Pour rappel, en 2019, nous avons introduit la possibilité d'utiliser des données synthétiques à l'apprentissage, donc la définition de la tâche a changé. Le Tableau 3.6 présente le classement des équipes et les résultats obtenus pour les différents jeux d'évaluation réels. Le classement est effectué en utilisant la F-mesure par événement (macro-moyenné) sur le jeu de données d'évaluation (Éval) qui est le regroupement des jeux « public » et « privé ». Le jeu d'évaluation de 2018 compose 75% du jeu de test 2019, car le jeu de test depuis 2019 est l'ensemble des jeux de test et d'évaluation de 2018. Les résultats ne sont donc pas comparable aux résultats de 2018. Douze équipes ont surpassé le système de référence et les meilleurs systèmes [[Delphin-Poulat and Plapous, 2019](#), [Lin and Wang, 2019](#), [Shi, 2019](#)] ont surpassé celui-ci de 16%. Les meilleurs systèmes ont aussi surpassé le meilleur système de 2018 de 10%. Nous notons que les trois premiers systèmes du classement ont utilisé un modèle « professeur moyen » semi-supervisé. [Lin and Wang \[2019\]](#) se sont focalisés sur l'apprentissage semi-supervisé en utilisant un apprentissage guidé et en montrant que les données synthétiques peuvent aider dans cette configuration lorsqu'elles sont utilisées avec suffisamment de données

Rang	Système	Classifieur	Clips réels		
			Éval	Public	Privé
1	Lin, ICT	CNN	42,7%	47,7%	29,4%
2	Delphin-Poulat, OL	CRNN	42,1%	45,8%	33,3%
3	Shi, FRDC	CRNN	42,0%	46,1%	31,5%
4	Cances, IRIT	CRNN	39,7%	43,0%	30,9%
5	Yan, USTC	CRNN	36,2%	38,8%	28,7%
6	Lim, ETRI	CRNN, Ensemble	34,4%	38,6%	23,7%
7	Kiyokawa, NEC	ResNet, SENet	32,4%	36,2%	23,8%
8	Chan, NU	NMF, CNN	31,0%	34,7%	21,6%
9	Zhang, UESTC	CNN, ResNet, RNN	30,8%	34,5%	21,1%
10	Kothinti, JHU	CRNN, CRBM, PCA	30,7%	33,2%	23,8%
11	Wang B., NWPU	CNN, RNN, ensemble	27,8%	30,1%	21,7%
12	Lee, KNU	CNN	26,7%	28,1%	22,9%
	Baseline 2019	CRNN	25,8%	29,0%	18,1%
13	Agnone, PDL	CRNN	25,0%	27,1%	20,0%
14	Rakowski, SRPOL	CNN	24,2%	26,2%	19,2%
15	Kong, SURREY	CNN	22,3%	24,1%	17,0%
16	Mishima, NEC	ResNet	19,8%	21,8%	15,0%
17	Wang D., NUDT	CRNN	17,5%	19,2%	13,3%
18	Yang, YSU	CMRANN-MT	6,7%	7,6%	4,6%

TABLEAU 3.6 – F-mesure par événement des systèmes soumis à la Tâche 4 du Challenge DCASE 2019.

enregistrées. [Delphin-Poulat and Plapous \[2019\]](#) se sont focalisés sur l'augmentation de données et l'optimisation des paramètres du système de référence [[Delphin-Poulat et al., 2020](#)]. [Shi \[2019\]](#) s'est intéressé à un type spécifique d'augmentation qui est le mélange de clips audio et de leurs labels. [Cances et al. \[2019\]](#) ont proposé un système avec apprentissage multi-tâche où les détections faibles et fortes sont considérés comme des tâches distinctes. Ce dernier système est aussi le moins complexe des systèmes proposés.

On peut constater une différence importante entre les résultats obtenus sur les données publiques et les données privées. Les données privées sont des données de Vimeo alors que les données publiques sont des données de Youtube. Les données d'apprentissage sont issues d'AudioSet, et donc initialement de vidéos Youtube. Bien qu'il n'y a aucun lien entre les vidéos d'apprentissage AudioSet et les vidéos Youtube du jeu public, la plateforme de partage est la même et certaines conditions notamment de compression ou de traitement sont similaires. Ces conditions peuvent être différentes sur la plateforme Vimeo. La différence de performance entre le jeu de données privé et public peut donc indiquer des spécificités des plateformes Vimeo et Youtube que les systèmes n'arrivent pas à prendre en compte. Ils sur-apprennent les caractéristiques de la plateforme Youtube et ne généralisent pas aux spécificités de la plateforme Vimeo.

Rang	Nom d'équipe	Séparation	Éval	Public	Privé
1	Miyazaki	Non	51,1	55,7	39,6
2	Hao	Non	47,8	52,3	39,0
3	Ebbbers	Non	47,2	50,9	38,7
4	Koh	Non	46,6	51,5	34,5
5	Yao	Non	46,4	50,5	36,0
6	CTK	Non	46,3	50,5	35,3
7	Liu	Non	45,2	51,2	30,3
8	Zhenwei	Non	45,1	49,0	35,2
9	Huang	Oui	44,7	49,5	32,7
10	Cornell	Oui	44,4	48,6	33,8
11	Kim_AiTeR	Non	44,4	48,0	35,5
12	Tang_SCU	Non	44,1	47,5	35,3
13	YenKu_NTU	Non	43,6	48,5	30,8
	Top 2019	Non	42,7%	47,7%	29,4%
14	LJK_PSH	Non	41,2	45,8	29,7
15	deBenito_AUDIAS	Non	38,2	42,0	29,1
16	PARK_JHU	Non	36,9	40,2	28,7
	Baseline	Oui	36,5	39,8	28,8
17	Xiaomi_task4	Non	36,0	40,7	25,3
18	Hou_IPS	Non	34,9	38,1	27,8
19	Chen_NTHU	Oui	34,5	37,8	26,9
20	Rykaczewski_Samsung	Non	21,9	24,0	15,7
21	Copiaco_UOW	Oui	7,8	8,5	5,8

TABLEAU 3.7 – F-mesure par événement des systèmes soumis à la Tâche 4 du Challenge DCASE 2020. Top 2019 représente le meilleur système soumis en 2019.

3.3.3 DCASE 2020

La Tâche 4 du Challenge DCASE 2020 a accueilli 73 soumissions (dont 10 avec séparation de sources) de 21 équipes différentes impliquant 90 chercheurs. En 2020 la séparation de sources apparaît dans la définition du problème ainsi que l'utilisation de la réverbération dans les clips audio synthétiques. Le Tableau 3.7 présente les résultats des équipes pour les données d'évaluation enregistrées qui sont les mêmes que celles de 2019. Nous constatons que 13 équipes sur les 21 présentes dans ce tableau ont dépassé le résultat du meilleur système de l'année 2019, allant même jusqu'à 8% d'amélioration pour le meilleur système. La différence n'est pas liée à la séparation de sources puisque le premier système utilisant la séparation de sources, celui de [Huang et al. \[2020b\]](#), est classé neuvième avec des résultats proches de ceux du meilleur système de 2019. Parmi les systèmes, [Miyazaki et al. \[2020\]](#) utilisent un modèle d'attention, plus spécifiquement un Transformer qui est une approche populaire pour des applications utilisant le texte par exemple. [Hao et al. \[2020\]](#) effectuent une adaptation de domaine pour renforcer l'apport des données synthétiques

et réussissent à obtenir un système peu complexe avec de très bons résultats. [Koh et al. \[2020\]](#) proposent un système basé sur l'augmentation de données (mélange de fichiers audio et transposition temporelle des signaux) ainsi que la ré-annotation des données non-annotées. [Cornell et al. \[2020\]](#) proposent plusieurs idées originales dont l'utilisation du pré-traitement de normalisation par chaîne de fréquence, l'augmentation de données, l'apprentissage de réseaux antagonistes pour traiter la différence de domaine et le post-traitement des détections en utilisant un modèle de Markov caché. Ils utilisent également la séparation de sources basée sur le système de référence. Ils montrent l'intérêt de chacune des approches proposées de façon séparée et de façon combinée mais ils ne se placent qu'à la dixième place.

3.4 Conclusion

Dans ce chapitre, nous avons présenté une partie des contributions scientifiques liées à l'organisation de la Tâche 4 du Challenge DCASE. Nous avons mis en lumière les problèmes pouvant apparaître lors de scénarios de détection d'événements sonores en environnements réels. Nous avons développé un jeu de données appelé [DESED](#) pour aider à l'analyse et la résolution de ces problèmes. Nous avons adapté ce jeu de données au cours des années afin de prendre en compte les nouveaux problèmes identifiés ou les nouvelles solutions possibles. L'identification de nouveaux problèmes passe par le développement d'un système de référence, qui sera présenté dans le chapitre 4, et la réflexion autour de pistes d'amélioration permises notamment grâce l'analyse des systèmes soumis chaque année. Nous avons finalement présenté les résultats principaux ainsi que les systèmes marquants de chaque édition.

4 Systèmes de référence et analyse des résultats

Dans ce chapitre, nous nous intéressons à la définition des systèmes de référence de la Tâche 4 du Challenge DCASE entre 2018 et 2020 et à l'analyse du système de référence de 2020. Ensuite, nous analysons en détail les différents systèmes soumis pour chaque édition en rapport avec certains problèmes précis. À partir de 2019, l'utilisation des données synthétiques nous a permis de proposer aux participants des jeux de données qui isolent certains problèmes particuliers. Ces données sont incluses dans le jeu de données d'évaluation au même titre que les données enregistrées afin de permettre une analyse plus poussée de chacun des systèmes et comprendre quels sont les problèmes les plus rencontrés et les plus critiques. Ceci nous permet la redéfinition continue de la tâche d'une année à l'autre. Dans ce chapitre, nous définissons les systèmes de référence, nous proposons une étude ablative du système de référence de 2020, et finalement nous analysons en détail les comportements des systèmes soumis à la Tâche 4.

4.1 Les systèmes de référence

Les systèmes de référence sont un point important de la Tâche 4 du Challenge DCASE. Ces systèmes servent de point de comparaison aux participants durant toute la durée du challenge et de nombreux systèmes s'en inspirent. Le partage du code permet la réutilisation des parties souhaitées par les participants.

4.1.1 DCASE 2018

Le système de référence de 2018¹ est un **CRNN**, c'est-à-dire un **CNN** suivi d'un **RNN**. La Figure 4.1 illustre le modèle **CRNN** utilisé. Les données d'entrée de ce système sont des spectrogrammes log-Mel de 64 bandes extraites de fenêtres de 40 ms avec 50% de chevauchement. Le **CNN** utilise 3 couches convolutionnelles de 64 filtres de taille 3×3 et une réduction de facteur 4 sur l'échelle des fréquences par l'utilisation d'une agrégation maximum entre chaque couche ainsi qu'un *dropout* de 30%. L'entrée de chaque couche du **CNN** est normalisée par lots (*batch normalization*). Le **RNN** est bidirectionnel et utilise une couche de 64 *unités récurrentes par portes* (*gated recurrent units*) (**GRU**) avec un *dropout* de 30% à l'entrée. Le réseau se termine par une couche dense linéaire de 10 unités en sortie (pour les 10 classes d'intérêt) utilisant une fonction d'activation sigmoïde.

1. https://github.com/DCASE-REPO/dcase2018_baseline/tree/master/task4

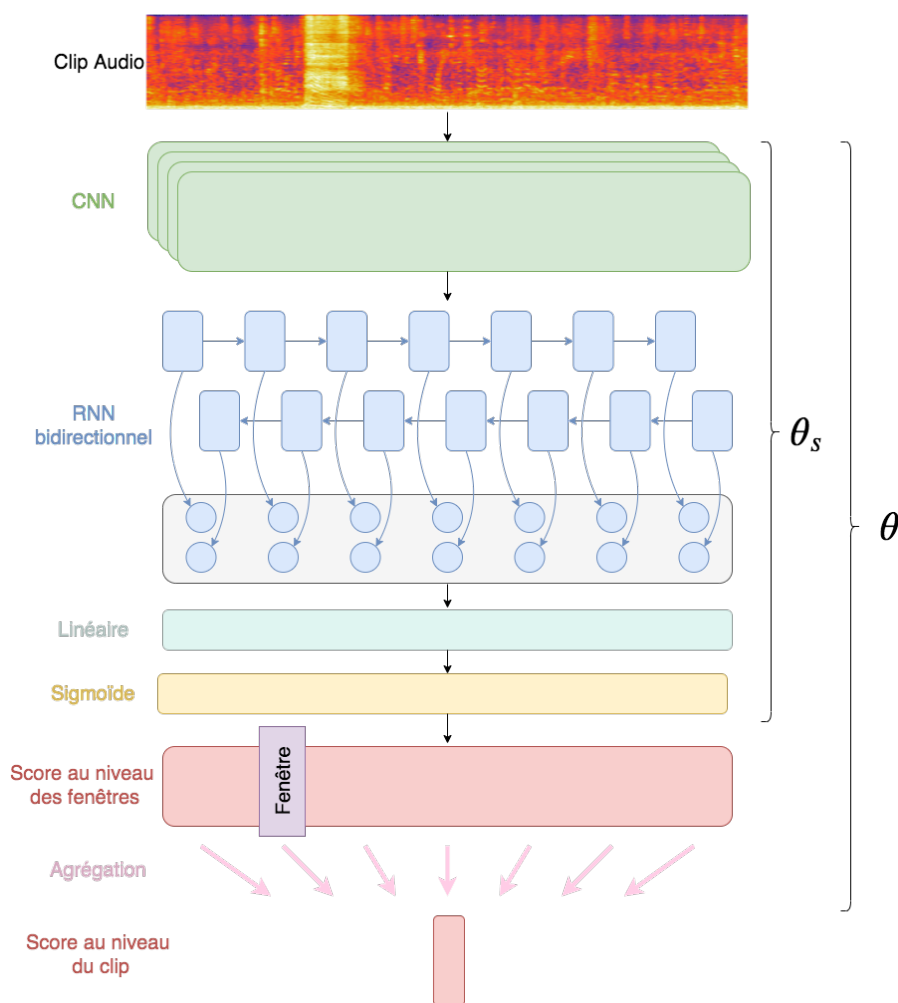


FIGURE 4.1 – Illustration d'un réseau convolusionnel et récurrent.

Notre système utilise deux modèles appris successivement : le premier modèle est utilisé pour pseudo-annoter les données non-annotées et le deuxième modèle pour faire la détection d'événements sonores sur le jeu d'évaluation. Le premier modèle est entraîné pendant 100 époques (une époque représente une passe complète sur l'ensemble d'apprentissage) avec un arrêt prématuré après 15 époques sans amélioration. Nos données d'apprentissage représentent 80% des données faiblement annotées, les 20% restants sont utilisés pour la validation. Ce modèle est appris à l'échelle du clip, c'est-à-dire que les détections au niveau des fenêtres sont agrégées en moyenne pour obtenir des détections au niveau du clip. Ces détections sont comparées aux annotations faibles correspondantes. Une fois appris, ce modèle nous permet de pseudo-annoter les 14412 clips non-annotés. Les pseudo-annotations au niveau des clips sont transformées en pseudo-annotations au niveau des fenêtres en indiquant la présence de l'événement sur la totalité du clip (bien que ce ne soit pas le cas en réalité) pour permettre un apprentissage du second modèle

au niveau des fenêtres. Nous apprenons le second modèle qui utilise la même architecture que le premier à partir des 14412 données précédemment non-annotées et maintenant pseudo-annotées grâce au premier modèle. Pour permettre l'apprentissage supervisé au niveau des fenêtres, le deuxième modèle n'utilise pas la couche d'agrégation moyenne ce qui permet la détection temporelle des événements. L'ensemble du jeu de données initialement faiblement annoté (pas seulement les 80% utilisés pour apprendre le premier modèle) dont les annotations sont transformées en annotations temporelles est utilisé pour la validation de ce second modèle. Afin d'assurer une certaine stabilité des événements temporels, un filtre médian sur 51 fenêtres ($\simeq 1$ s) est appliquée à la sortie du réseau.

Les résultats obtenus avec le modèle de la première passe et de la deuxième passe sont présentés dans le Tableau 4.1. Les résultats sont faibles, mais ceci s'explique par une métrique peu permissive et le fait que l'approche proposée et les données utilisées qui n'ont pas d'annotations fortes au départ ne permettent pas de résoudre le problème de façon aussi précise que ce que la métrique nécessiterait. La Figure 4.2 montre un exemple où la métrique donnerait 0% alors que détections et événements de référence sont proches. Le calcul de la métrique du premier modèle utilise les détections temporelles (sans l'agrégation moyenne) auxquelles on applique un filtre médian de 51 fenêtres. Le modèle n'a donc pas vu de données fortement annotées à l'apprentissage et a été optimisé sur un critère au niveau des clips. L'approche en deux étapes permet d'améliorer sensiblement la performance sur certaines classes. Ce tableau nous indique l'omniprésence de 0% pour certaines classes : « chat », « chien » et « voix ». Cela s'explique en partie par le fait que le premier modèle ne détecte pas ou très peu de ces classes. L'autre gros problème est la mauvaise performance de segmentation du modèle qui n'arrive pas à détecter les événements courts. La classe « plats de cuisine » est assez représentative dans ce sens. Les données fortement annotées de mauvaise qualité obtenu à la première passe ne permettent pas d'améliorer la performance de la deuxième passe. En revanche, les événements longs

Classe	1 ^{ère} passe	2 ^{ème} passe
Alarme/sonnette/sonnerie	3,2%	3,9%
Blender	10,1%	15,4%
Chat	0,0%	0,0%
Plats de cuisine	1,9%	0,0%
Chien	0,0%	0,0%
Tondeuse/brosse à dent électrique	18,2%	32,4%
Friture	9,4%	31,0%
Eau qui coule	7,6%	11,4%
Voix	0,0%	0,0%
Aspirateur	24,8%	46,5%
Macro-moyenne	7,51%	14,06%

TABLEAU 4.1 – F-mesure par événement du système de référence 2018 calculé sur l'ensemble d'évaluation 2018.

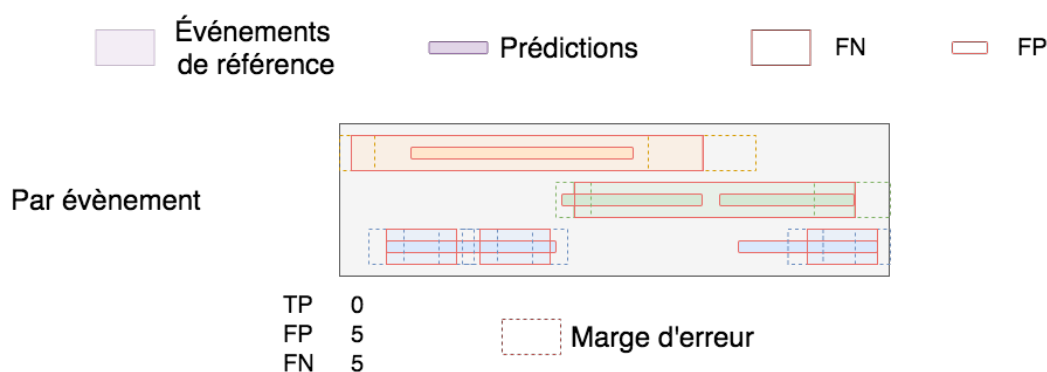


FIGURE 4.2 – Illustration des cas où la métrique utilisée est non permissive.

sont généralement bien reconnus dès la première passe et la performance s’améliore à la deuxième passe. Dans ce cas l’apport des données pseudo-annotées (pour des classes moins sensibles à la segmentation) est bénéfique. Ceci pousse à penser que nous sommes ici face à un problème de segmentation.

4.1.2 DCASE 2019

Le système développé en 2019² est inspiré du système qui avait obtenu les meilleures performances en 2018 [JiaKai, 2018]. Les différences principales de ce système avec celui de 2018 sont l’utilisation de données fortement annotées synthétiques qui n’étaient pas disponibles en 2018 et l’utilisation de l’apprentissage appelé « professeur moyen ». Le modèle utilisé est un **CRNN** proche de celui utilisé en 2018. Le **CNN** est identique à celui de 2018 à l’exception qu’entre les couches, en plus de la réduction de facteur 4 sur l’échelle des fréquences, il y a aussi une réduction de facteur 2 sur l’échelle temporelle et le *dropout* est fixé à 50% au lieu de 30%. Le **RNN** a 2 couches bidirectionnelles de 64 unités récurrentes par portes. Lorsqu’une sortie temporelle (non agrégée) est nécessaire, une couche linéaire de 10 neurones avec une activation sigmoïde est utilisée. Lorsque la sortie est agrégée, l’agrégation moyenne utilisée en 2018 est remplacée par une agrégation d’attention (multiplication de la couche linéaire activée par sigmoïde par une couche linéaire activée par un softmax).

Le modèle est appris avec l’optimiseur Adam durant 100 époques et un accroissement progressif du taux d’apprentissage est utilisé lors de la moitié des époques. Les données faiblement annotées et synthétiques sont séparées en jeux d’apprentissage et de validation, avec respectivement 80% et 20% des données. La validation du modèle se fait grâce à la somme d’une F-mesure calculée au niveau du clip sur les données de validation faiblement annotées et une F-mesure temporelle calculée par évènement sur les données de validation synthétiques. Cette métrique utilise les sorties du modèle « étudiant ».

Le changement principal est que le système utilise un apprentissage dit de professeur moyen qui s’appuie sur deux modèles d’architecture identique. Le premier **CRNN** qui a

2. https://github.com/turpaultn/DCASE2019_task4/

pour ensemble de paramètres θ est appelé « étudiant ». Il est appris directement à partir des données. L'autre modèle est appelé « professeur » et ses paramètres notés θ' sont mis à jour par une moyenne exponentielle glissante des paramètres du modèle « étudiant ». Seul le modèle étudiant est utilisé lors de l'inférence, le modèle professeur est utilisé afin d'aider le modèle étudiant à apprendre.

La Figure 4.3 représente le principe utilisé lors de l'apprentissage du professeur moyen. Le modèle étudiant est appris de façon supervisée grâce aux annotations du jeu de données d'apprentissage et de façon non-supervisée en comparant ses sorties avec les sorties du modèle professeur. Lors de la comparaison entre les sorties des deux modèles, le modèle professeur utilise la même donnée d'entrée que le modèle étudiant sur laquelle on ajoute un bruit gaussien. Un *dropout* différent par modèle peut aussi être utilisé et est représenté par η et η' dans la Figure 4.3. Notons par θ_s l'ensemble des paramètres du modèle étudiant sans la couche d'agrégation comme défini dans la Figure 4.1. En fonction des annotations disponibles pour les données d'entrées, différents coûts peuvent être optimisés pour mettre à jour les paramètres du modèle étudiant. Le modèle dont les paramètres sont θ_s permet de faire des détections au niveau des fenêtres alors que le modèle dont les paramètres sont θ permet de faire des détections au niveau des clips (voir Figure 4.1). Notons que $\theta_s \subset \theta$. Le modèle étudiant utilise :

- un coût de classification faiblement supervisé $L_{\text{class}_w}(\theta)$ sur les données faiblement annotées (entropie croisée binaire),
- un coût de classification fortement supervisé $L_{\text{class}_s}(\theta_s)$ sur les données synthétiques (entropie croisée binaire sur chaque fenêtre),
- un coût de cohérence faible $L_{\text{cons}_w}(\theta)$ entre les sorties au niveau du clip du modèle

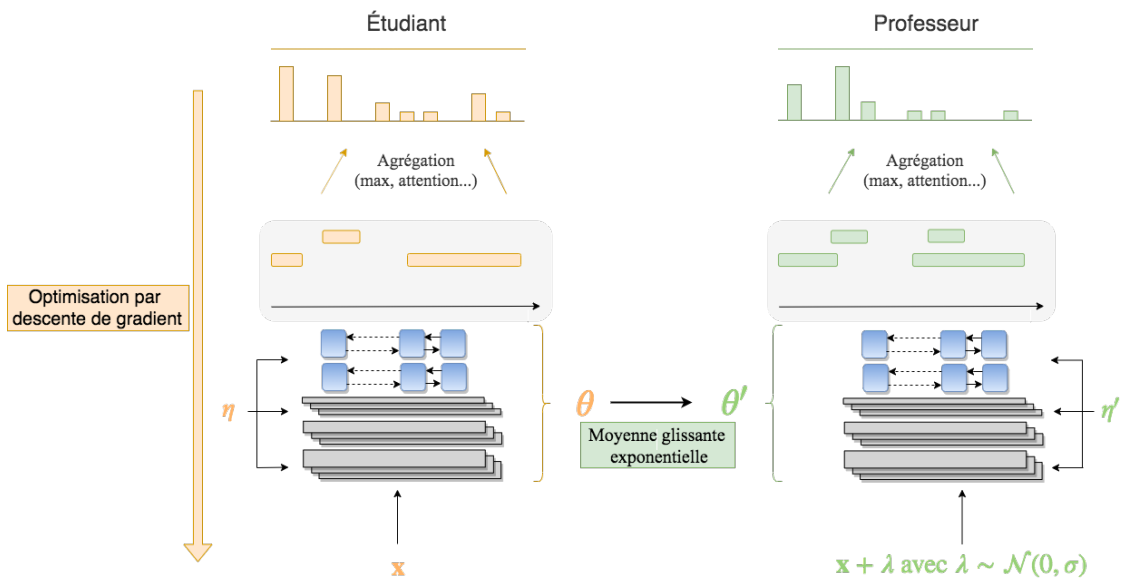


FIGURE 4.3 – Définition du modèle « professeur moyen ». η et η' représentent le bruit appliqué aux différents modèles par *dropout*.

professeur et du modèle étudiant (erreur quadratique moyenne),

- un coût de cohérence forte $L_{\text{cons}_s}(\theta_s)$ entre les sorties au niveau des fenêtres du modèle professeur et du modèle étudiant (erreur quadratique moyenne sur chaque fenêtre).

Chaque lot de données contient un ensemble de données non-annotées, faiblement annotées et fortement annotées pour permettre l'optimisation simultanée de tous ces coûts. La proportion que représente chaque sous-ensemble dans les lots est un paramètre qui sera étudié par la suite. Le coût total est calculé en combinant ces 4 coûts :

$$L(\theta) = L_{\text{class}_w}(\theta) + s(e)L_{\text{cons}_w}(\theta) + L_{\text{class}_s}(\theta_s) + s(e)L_{\text{cons}_s}(\theta_s) \quad (4.1)$$

où $s(e)$ représente le poids des coûts de cohérence (apprentissage par le modèle professeur) dans le coût total. La valeur de $s(e)$ est augmentée au cours de l'apprentissage (par exemple selon une courbe sigmoïde), où e représente le nombre d'époques écoulées. Ce paramètre est important parce qu'au début de l'apprentissage le modèle professeur est mauvais, donc il semble important de démarrer avec un poids faible pour favoriser l'apprentissage à partir des données annotées et d'augmenter le poids au cours du temps. La Figure 4.4 représente le système utilisant le modèle de professeur moyen utilisé comme système de référence de la Tâche 4 en 2019.

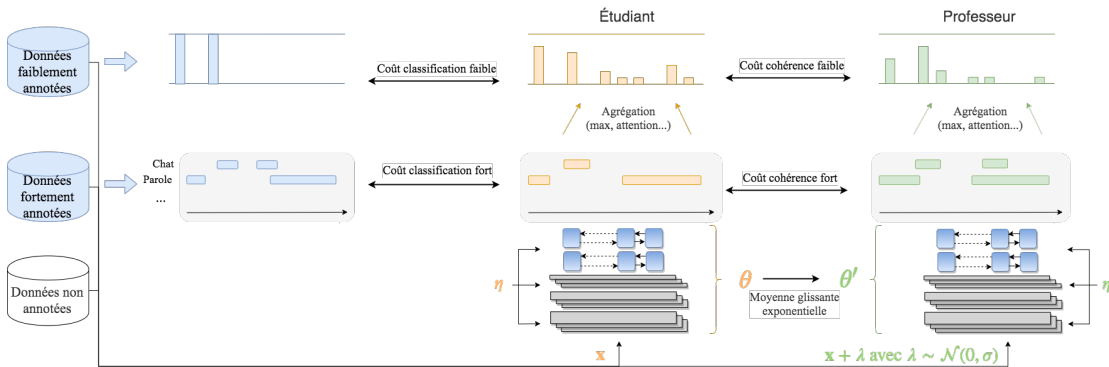


FIGURE 4.4 – Définition du système utilisant le modèle « professeur moyen ». η et η' représentent le bruit appliqué aux différents modèles par *dropout*.

4.1.3 DCASE 2020

Système de référence sans séparation de sources Le système de référence développé en 2020³ utilise un modèle de professeur moyen comme celui de 2019. Celui-ci est optimisé en s'inspirant du système qui a fini second en 2019 [Delphin-Poulat and Plapous, 2019]. Ce modèle est lui-même basé sur le système de référence qui a été optimisé en utilisant une architecture de modèle différente, des techniques d'augmentation de données ou de post-traitement différentes. Contrairement aux systèmes de 2018 et 2019 qui utilisaient

3. https://github.com/turpaultn/dcase20_task4

des données d’entrées échantillonnées à 44,1 kHz et un spectrogramme à 64 bandes Mel, le système de 2020 utilise des données d’entrées échantillonnées à 16 kHz⁴. Le nombre de bandes Mel utilisé est de 128 comme indiqué par [Delphin-Poulat and Plapous \[2019\]](#). Le Tableau 4.2 compare les caractéristiques d’entrée utilisées en 2019 et 2020 et indique que ce changement impact positivement la F-mesure mais réduit légèrement le PSDS. De manière générale, ce changement a peu d’impact, mais semble favoriser la segmentation des événements.

	44,1 kHz & 64 Mels	44,1 kHz & 128 Mels	16 kHz & 128 Mels
F-mesure	31,4	32,3 %	34,1 %
PSDS	0,518	0,511	0,502

TABLEAU 4.2 – Impact des caractéristiques d’entrées sur le jeu de données de test.

Le [CRNN](#) est composé d’un [CNN](#) de 7 couches avec 16, 32, 64, 128, 128, 128, et 128 filtres 3×3 par couche respectivement. Les facteurs de réduction respectifs sont de [2, 2], [2, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2] par couche, où la première dimension est le facteur temporel et la deuxième le facteur fréquentiel. Le *dropout* est de 50% entre chaque couche convolutionnelle. La fonction d’activation des couches convolutionnelles est désormais l’unité linéaire par porte (remplaçant l’unité de rectification linéaire) qui par sa composante linéaire permet de propager l’erreur plus loin dans le réseau et de mettre à jour de façon plus importante les poids des premières couches du réseau. Le [RNN](#) est composé de 2 couches bidirectionnelles de 128 unités récurrentes par porte. La sortie se fait par une couche linéaire activée par sigmoïde ou par agrégation basée sur l’attention.

Les augmentations discutées par [Delphin-Poulat et al. \[2020\]](#) et leur post-traitement n’ont pas été retenus dans notre système de référence car ils sont jugés trop spécifiques et ont tendance à sur-apprendre [[Serizel et al., 2020](#)]. Les données d’entrées synthétiques sont différentes pour inclure la nouvelle façon de générer directement des jeux synthétiques d’apprentissage et de validation à partir d’événements isolés différents comme discuté dans la partie 3.2.4. Les données faiblement annotées sont séparées en jeu d’apprentissage et de validation, avec respectivement 90% et 10% des données. La validation du modèle se fait grâce à la somme d’une F-mesure calculée au niveau du clip sur les données de validation faiblement annotées et une F-mesure temporelle calculée par événement sur les données de validation synthétiques, en utilisant les sorties du modèle étudiant. Comme

4. Nous avons effectués ce changement principalement pour des raisons de compatibilité avec le modèle de séparation.

	Public
Baseline 2019	29,0 %
Baseline 2020	38,1 %

TABLEAU 4.3 – Macro-moyenne des F-mesures par événement sur le jeu d’évaluation public.

on peut le voir dans le Tableau 4.3, la performance entre 2019 et 2020 s'est améliorée de 9% grâce à ces changements. Cependant, il est difficile de comprendre quels changements influent véritablement sur la performance, c'est pourquoi nous avons proposé une étude ablative de certains paramètres que nous discutons dans la partie 4.1.4.

Système de référence avec séparation de sources En 2020, nous avons proposé d'utiliser la séparation de sources comme pré-traitement de la détection d'événements sonores. Le modèle de séparation de sources utilisé est basé sur celui de Kavalerov et al. [2019] et Tzinis et al. [2020]⁵ et suppose un nombre fixe de sources. Il est appris sur le corpus FUSS [Wisdom et al., 2020]. La séparation de sources permet d'utiliser en entrée du modèle de détection d'événements sonores plusieurs sources sonores dont la détection est plus simple. L'agrégation des résultats de chaque source peut se faire à différents niveaux du modèle. Nous avons comparé l'agrégation des sources dans les premières couches du CNN, entre le CNN et le RNN, et à la sortie du modèle [Turpault et al., 2020b]. Nous avons choisi une agrégation à la sortie du modèle en utilisant le modèle pré-entraîné sur les mélanges (système de référence de détection d'événements sonores) comme illustré dans la Figure 4.5 car l'apprentissage du modèle à l'aide des sources et une agrégation à l'intérieur du modèle n'ont pas amélioré la performance de notre système. Ces résultats indiquent que l'utilisation de plusieurs sources complexifie le modèle qui a des difficultés à apprendre les paramètres et que l'apprentissage de la séparation de sources réalisé uni-

5. Merci à Scott Wisdom, John Hershey et Hakan Erdogan d'avoir défini et appris les modèles.

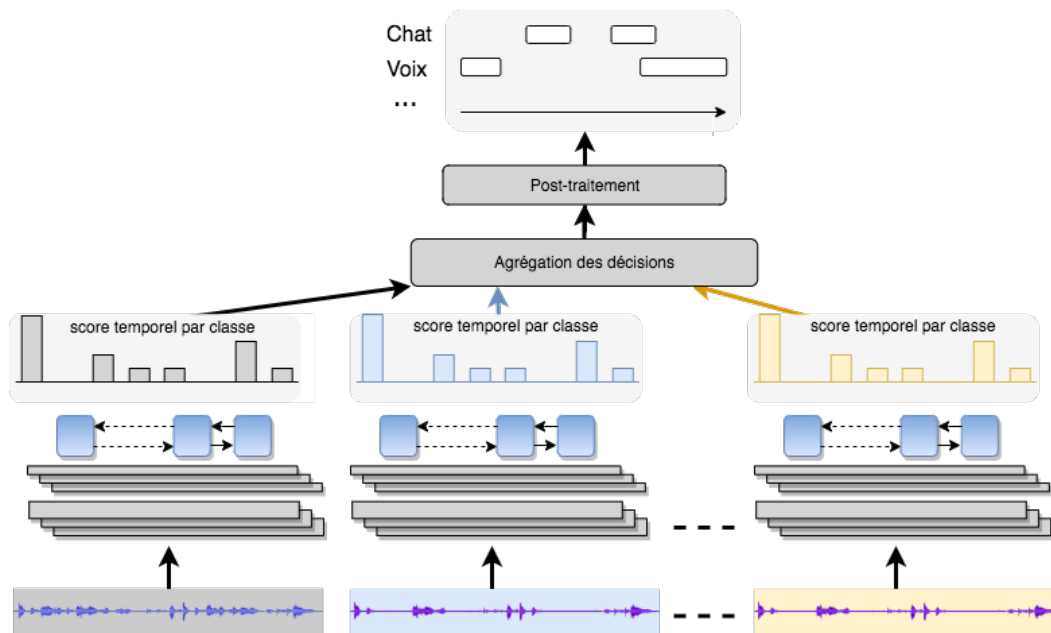


FIGURE 4.5 – Représentation du système utilisant la séparation de sources avec une agrégation des sources à posteriori.

quement sur des données synthétiques peut avoir des difficultés à s'adapter aux données enregistrées. Nous avons aussi proposé une analyse de différents modèles de séparation utilisant un nombre différent de sources et des apprentissages différents [Turpault et al., 2020b], qui ne sera pas présenté dans ce manuscrit.

4.1.4 Étude ablative du système de référence de DCASE 2020

Dans cette partie, nous effectuons une étude ablative du système de référence de DCASE 2020. Nous évaluons les résultats à l'aide de deux métriques. La première est la F-mesure par événement macro-moyennée avec une tolérance sur les instants de début de 200 ms et sur les instants de fin égale au maximum entre 200 ms et 20% de la durée de l'événement de référence (comme précédemment) qui évalue la segmentation stricte des événements. La deuxième est le PSDS calculé avec une tolérance d'intersection de 50% pour les événements de référence et les événements détectés, un déclencheur de croisement de 30%, un paramètre α_{ct} de 1 prenant en compte les déclencheurs de croisement et α_{st} de 0 ne prenant pas en compte l'écart type dans le calcul de la macro-moyenne. Le PSDS évalue la segmentation de façon moins stricte que la F-mesure mais il pénalise de façon plus importante les erreurs de classification.

Le système de référence de 2020 inclut un nombre important de changements par rapport au système de référence de 2019, tout comme le système de référence de 2019 incluait de nombreux changements par rapport au système de référence de 2018. La performance s'est améliorée, mais il est difficile de comprendre le rôle de chacun des changements dans le résultat final. Nous effectuons donc une étude ablative pour les paramètres que nous jugeons importants. Nous avons sélectionné trois paramètres concernant les données, et deux paramètres concernant le modèle :

- la proportion des données présentes par lot de données durant l'apprentissage pour connaître l'impact de chaque jeu de données dans l'apprentissage ;
- l'utilisation de changement de ton lors de la génération de données pour connaître l'impact de cette augmentation de données sur nos données isolées ;
- l'utilisation de la réverbération lors de la génération de données pour comprendre l'impact des conditions d'enregistrement ;
- le bruit ajouté aux données d'entrées du modèle professeur pour connaître l'impact de la cohérence entre données d'entrée des deux modèles pour obtenir une cohérence au niveau des sorties et pour calculer l'impact de la perturbation sur le processus d'apprentissage et l'effet de régularisation du modèle professeur ;
- l'utilisation de l'accroissement progressif du taux d'apprentissage et le poids de la cohérence pour tenter de comprendre l'impact de l'optimisation croissante et de la combinaison des différents coûts dans l'apprentissage.

Nous faisons des expériences pour chacun de ces paramètres en nous rapportant systématiquement au même modèle de référence avec les paramètres suivants :

- durant la phase d'apprentissage, les clips issus de chacun des jeux de données représentent $\frac{1}{3}$ des lots ;

- le changement de ton est utilisé à l'apprentissage et la validation ;
- la réverbération n'est pas utilisée ;
- le bruit ajouté aux données du modèle professeur est ajouté de manière à obtenir un rapport signal-à-bruit de 30 dB ;
- l'accroissement progressif est utilisé pour le taux d'apprentissage et pour le poids de cohérence.

Durant chacune des expériences suivantes nous faisons varier un ensemble de paramètres dans cette liste.

Le Tableau 4.4 indique les résultats obtenus en fonction des jeux de données utilisés lors de l'apprentissage et des proportions dans chaque lot. Les proportions présentées pour chaque cas ont été déterminées lors d'expérience préliminaires de manière à maximiser la F-mesure sur le jeu de validation synthétique et faiblement annoté. L'utilisation équilibrée de tous les jeux de donnée fournit les meilleurs résultats. Cela signifie que ces jeux de données sont complémentaires et chacun d'entre eux apporte de la connaissance lors de l'apprentissage. La dernière colonne du tableau indique la complémentarité importante entre les jeux de données synthétiques et faiblement annotées. Apprendre un système avec ces deux jeux de données permet d'obtenir une F-mesure de 31,76%, qui est la plus proche de celle obtenue en utilisant tous les jeux de données. Nous attribuons cette complémentarité au fait que les données faiblement annotées présentent une différence de granularité d'annotation avec le scénario d'évaluation qui rend l'apprentissage de la segmentation difficile mais elles correspondent au domaine cible, alors que les données synthétiques permettent l'apprentissage d'une segmentation à partir de données fortement annotées mais elles présentent une différence de domaine par rapport au domaine cible.

Jeu d'apprentissage	Ratio entre les jeux de données					
Synthétique	1/3	1	1/4			1/2
Faible	1/3			1	1/4	1/2
Non-annoté	1/3		3/4		3/4	
F-mesure	34,14%	20,41%	11,56%	16,46%	17,97%	31,76%
PSDS	0,502	0,250	0,140	0,287	0,328	0,435

TABLEAU 4.4 – F-mesure par événement et PSDS sur le jeu de test en fonction des jeux de données utilisés à l'apprentissage.

Le Tableau 4.5 montre l'impact de l'utilisation du changement de ton pour les données synthétiques. Le meilleur résultat en terme de F-mesure est obtenu en appliquant un changement de ton seulement aux données de validation alors que le meilleur résultat en terme de PSDS est obtenu en appliquant un changement de ton aux données d'apprentissage et de validation. Nous constatons que le changement de ton a un impact faible sur notre système. Cela indique que le système ne souffre pas d'un problème de généralisation ou que le changement de ton n'est pas une augmentation permettant d'améliorer la généralisation à de nouvelles données enregistrées.

Apprentissage Validation synth.	Changement de ton		
		✓	✓
F-mesure	35,15%	35,91%	34,14%
PSDS	0,487	0,495	0,502

TABLEAU 4.5 – F-mesure par événement et PSDS sur le jeu de test en fonction de l'application ou non d'un changement de ton sur les événements isolés lors de la génération de clips synthétiques.

Le Tableau 4.6 indique la performance obtenue lors de l'utilisation de la réverbération. On constate que la réverbération a un impact non négligeable sur le système. Son utilisation lors de l'apprentissage dégrade la performance en terme de F-mesure mais améliore la performance en terme de PSDS. Ceci peut indiquer que l'utilisation de la réverbération permet une meilleure généralisation en terme de classification des événements (PSDS plus élevé), cependant l'apprentissage de la segmentation précise des événements est plus difficile (F-mesure plus faible). L'utilisation de la réverbération lors de la validation uniquement nous permet de juger la différence de domaine que provoque l'utilisation de la réverbération et à quel point le système appris sans réverbération peut être robuste à la réverbération lors de la validation. Nous constatons que les performances sont très proches et que le système appris sans réverbération est donc déjà robuste à la réverbération utilisée. L'utilisation de la réverbération lors de l'apprentissage peut rendre les instants de début et de fin des événements plus « flous » à détecter et rendre la segmentation moins précise.

Apprentissage Validation synth.	Réverbération		
	✓	✓	
F-mesure	30,28%	33,66 %	34,14%
PSDS	0,513	0,524	0,502

TABLEAU 4.6 – F-mesure par événement et PSDS sur le jeu de test en fonction de l'application ou non de réverbération sur les événements isolés lors de la génération de clips synthétiques.

Le Tableau 4.7 indique l'impact du bruit appliqué en entrée du réseau professeur. Nous constatons que le bruit appliqué au modèle professeur a un impact important sur la performance en terme de F-mesure et de PSDS et que la performance se dégrade quand le bruit augmente (SNR faible). Ne pas ajouter de bruit aux données d'entrée du modèle professeur permet d'obtenir de meilleurs résultats. Ceci indique que le bruit ajouté aux données d'entrée du professeur n'aide pas l'apprentissage basé sur les coûts de cohérence. Notre hypothèse est que les différences entre les paramètres des deux modèles et le *dropout* indépendant de chaque modèle introduisent déjà une variabilité suffisante pour permettre l'apprentissage. Cette hypothèse pourrait être vérifiée avec des expériences

	SNR modèle professeur (dB)			
	0	15	30	∞
F-mesure	12,56%	26,29%	34,14%	37,80%
PSDS	0,261	0,437	0,502	0,540

TABLEAU 4.7 – F-mesure par événement et **PSDS** sur le jeu de test en fonction du rapport signal-à-bruit (SNR) du bruit appliqué à l’entrée du réseau professeur dans le modèle « professeur moyen ».

complémentaires sur le *dropout* et la fenêtre de calcul de la moyenne des paramètres pour le modèle professeur.

Le Tableau 4.8 indique l’impact de l’accroissement progressif du taux d’apprentissage et du poids du coût de cohérence. Lorsque l’accroissement progressif n’est pas appliqué au coût de cohérence, deux valeurs fixes de ce poids sont considérées. Ce tableau montre l’importance de l’accroissement progressif pour le taux d’apprentissage : il y a au minimum 6 % de différence en terme de F-mesure entre les résultats sans accroissement progressif et avec accroissement progressif du taux d’apprentissage pour les scénarios avec un poids du coût de cohérence fixé. Nous attribuons cela au fait que l’accroissement progressif du taux d’apprentissage réduit les sauts importants au début de l’apprentissage qui peuvent rendre ensuite l’apprentissage par cohérence difficile puisque les poids sont fortement modifiés. L’accroissement progressif du poids du coût de cohérence permet de limiter son impact au début de l’apprentissage. En effet, au début de l’apprentissage les paramètres du modèle élève sont très variables, ce qui peut avoir un impact néfaste sur le modèle professeur et donc sur le modèle élève via le coût de cohérence. Augmenter progressivement l’importance du coût de cohérence permet au modèle élève de converger à partir des annotations, et au modèle professeur de se stabiliser avant d’être utilisé pour régulariser le modèle élève via le coût de cohérence.

Accroissement CC			✓			✓
Accroissement TA				✓	✓	✓
Poids de cohérence	1	2	[0, 2]	1	2	[0, 2]
F-mesure	24,20%	27,96%	25,28%	31,97%	33,05%	34,14%
PSDS	0,408	0,412	0,420	0,489	0,475	0,502

TABLEAU 4.8 – F-mesure par événement et **PSDS** sur le jeu de test en fonction de l’accroissement progressif ou non du taux d’apprentissage (TA) et du poids du coût de cohérence (CC) et de la valeur de ce poids.

Chacun des paramètres considérés ayant été évalué individuellement, il faut maintenant vérifier s’ils ont un effet complémentaire ou pas. Pour ce faire, nous évaluons la combinaison des paramètres au niveau des données (la réverbération et le changement de ton) dans le Tableau 4.9. En effet, les paramètres au niveau du modèle sont ceux du système de référence (qui ont donné les meilleurs résultats en terme de F-mesure), excepté le bruit appliqué aux données d’entrée du modèle professeur qui a été supprimé (rapport signal-

Réverbération		✓	✓		
Changement ton		✓		✓	
Test	F-mesure	31,13%	36,27%	37,80%	34,46%
	PSDS	0,482	0,521	0,540	0,520
Évaluation	F-mesure	33,7%	39,9%	39,0%	36,8%
	PSDS	0,515	0,568	0,552	0,566

TABLEAU 4.9 – F-mesure par événement et PSDS sur les jeux de test et d'évaluation publics en fonction de l'application ou non de réverbération et du changement de ton sur le jeu de validation synthétique.

à-bruit infini). Le système le plus performant sur l'ensemble de test est celui utilisant le changement de ton pour les données de validation synthétiques et aucune réverbération. L'indication conjointe des performances de test et d'évaluation dans ce tableau montre que les meilleurs paramètres pour le jeu de données de test ne sont pas nécessairement les meilleurs pour le jeu de données d'évaluation. Ceci montre qu'un modèle est souvent biaisé envers le jeu de données sur lequel on a optimisé ses paramètres. Cela justifie l'importance de ne pas sur-apprendre le système sur les données de test. Dans notre cas, le jeu de données de test n'est utilisé que pour le choix de certains hyper-paramètres (ceux discutés ici).

4.2 Analyse des résultats

Dans la partie 3.3, nous avons présenté les résultats officiels de la Tâche 4 du Challenge DCASE. Dans cette partie, nous proposons une analyse plus détaillée des systèmes soumis afin d'identifier leurs avantages et inconvénients selon certains critères spécifiques, les problèmes qui semblent généralement résolus et ceux qui posent encore problème à l'ensemble des systèmes. Cette analyse nous permet aussi de questionner la définition de la tâche et les solutions pour l'améliorer et la rendre plus pertinente, aussi bien scientifiquement que du point de vue applicatif.

Dans la Figure 4.6, nous présentons un premier élément important d'analyse des résultats : la distribution des durées des événements en fonction de leur classe. Cette distribution est calculée sur le jeu de données de test. Une distribution similaire est observée pour le jeu de données d'évaluation. Deux grandes familles d'événements se distinguent : les classes d'événements courts avec en majorité des événements qui durent moins d'une seconde et les classes d'événements longs avec en majorité des événements qui durent plus de 5 s. Les événements courts regroupent les classes « alarme/sonnette/sonnerie », « chat », « plats de cuisine », « chien » et « voix ». Les événements longs regroupent les autres classes « blender », « rasoir / brosse à dent électrique », « friture », « eau qui coule » et « aspirateur ». Ces dénominations seront utilisées tout au long de ce manuscrit pour faire référence à ces deux sous-ensembles de classes.

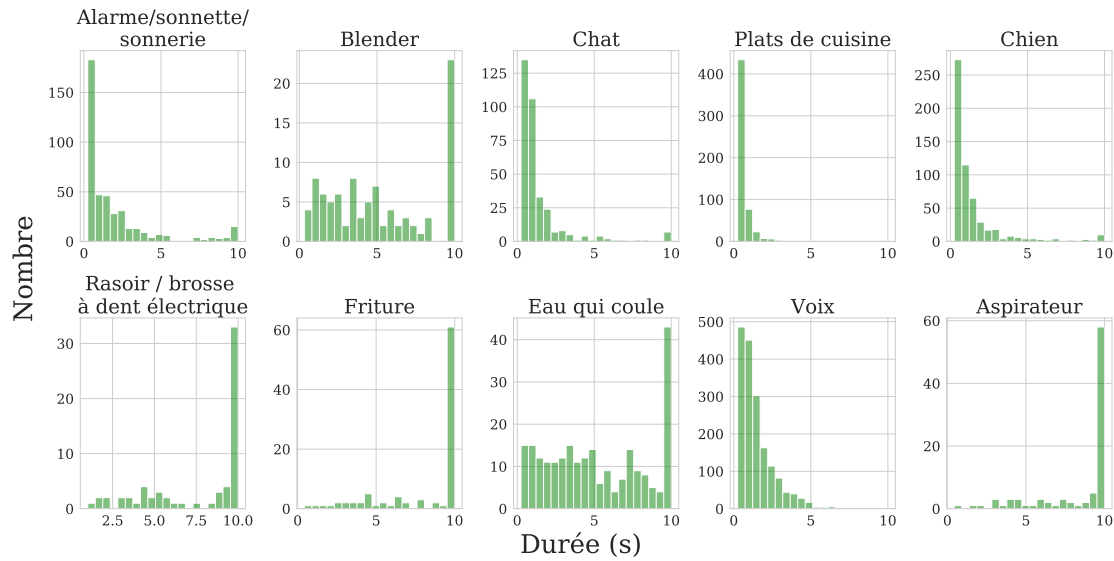


FIGURE 4.6 – Distribution des classes d'événements sonores dans le jeu de test.

4.2.1 DCASE 2018

Dans le Tableau 4.10, nous présentons les résultats par classe pour les 10 meilleurs systèmes soumis à la Tâche 4 du Challenge DCASE 2018. Nous identifions une disparité des résultats entre les différentes classes. En règle générale, les systèmes sont moins performants pour les classes d'événements courts. Lorsqu'un système détecte bien les événements des classes d'événements courts [Hou and Li, 2018, Kothinti et al., 2018], il a tendance à moins bien détecter les événements des classes d'événements longs, et inversement lorsqu'un système détecte bien les événements des classes d'événements longs, il détecte plus difficilement les événements des classes d'événements courts [JiaKai, 2018, Kong et al., 2018]. Il semble qu'aucun système ne domine clairement dans la détection d'événements à la fois sur les classes d'événements longs et les classes d'événements courts. Ceci est confirmé par les Tableaux 4.11 et 4.12 qui présentent les 5 meilleurs systèmes pour les événements courts et longs respectivement. Le Tableau 4.11 indique que la meilleure performance pour l'ensemble des classes d'événements courts est de 26,4%. Le Tableau 4.12 indique que la meilleure performance pour l'ensemble des classes d'événements longs est de 42,6%. Seuls les 2 premiers systèmes sont présents dans les deux tableaux et chacun de ces systèmes est spécialisé dans un type de classe d'événements. La difficulté des systèmes à détecter les événements courts peut s'expliquer par la difficulté qu'ils ont à segmenter les événements.

Pour confirmer cette hypothèse, la Figure 4.7 présente les résultats de segmentation (détection des bonnes frontières temporelles sans tenir compte de la classe) pour l'ensemble des systèmes soumis avec une tolérance de 0,2 s, 1 s, ou 5 s. La majorité des systèmes est capable de détecter les événements sans frontières bien définies (Figure 4.7c). En revanche, peu de systèmes sont capables de segmenter les événements avec la tolérance de

Système	Alarme	Blender	Chat	Plats	Chien	R/Bàd	Friture	Eau	Voix	Aspi
jiakai_psh	49,9	38,2	3,6	3,2	18,1	48,7	35,4	31,2	46,8	48,3
liu_ustc	46,0	27,1	20,3	13,0	26,5	37,6	10,9	23,9	43,1	50,0
kong_surrey	24,5	18,9	7,8	7,7	5,6	46,4	43,6	15,2	19,9	50,0
kothinti_jhu	36,7	22,0	20,5	12,8	26,5	24,3	0,0	9,6	34,3	37,0
harb_tug	15,4	30,0	8,1	17,5	9,7	21,0	34,7	17,3	31,1	31,5
koutini_jku	30,0	16,4	13,1	9,5	8,4	23,5	18,1	12,6	42,9	40,8
guo_thu	35,3	31,8	7,8	4,0	9,9	17,4	32,7	18,3	31,0	24,8
hou_bupt	41,4	16,4	6,4	23,5	20,2	9,8	6,2	14,0	40,6	32,3
lim_etri	11,6	21,6	7,9	5,9	17,4	27,8	14,9	15,5	21,0	60,0
avdeeva_itmo	33,3	15,2	14,9	6,3	16,3	15,8	24,6	13,3	27,2	34,8
baseline	4,8	12,7	2,9	0,4	2,4	20,0	24,5	10,1	0,1	30,2

TABLEAU 4.10 – F-mesure par événement pour chacune des classes d’intérêt pour les 10 meilleurs systèmes soumis en 2018. Alarme indique « Alarme/sonnette/sonnerie », plats indique « Plats de cuisine », R/Bàd indique « Rasoir / brosse à dent électrique », Eau indique « Eau qui coule » et Aspi indique « Aspirateur ».

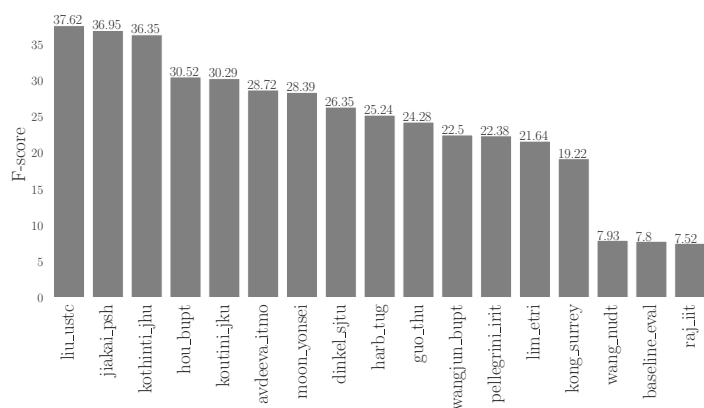
Système	Courts	Longs	Tous	Rang
liu_ustc	26,4	31,4	29,9	2
kothinti_jhu	24,1	20,8	22,4	4
hou_bupt	22,9	16,2	21,1	8
jiakai_psh	18,7	42,6	32,4	1
avdeeva_itmo	17,7	22,6	20,1	10
baseline	2,6	21,8	10,8	16

TABLEAU 4.11 – F-mesure par événement des 5 meilleurs systèmes pour les classes d’événements courts ainsi que leur classement.

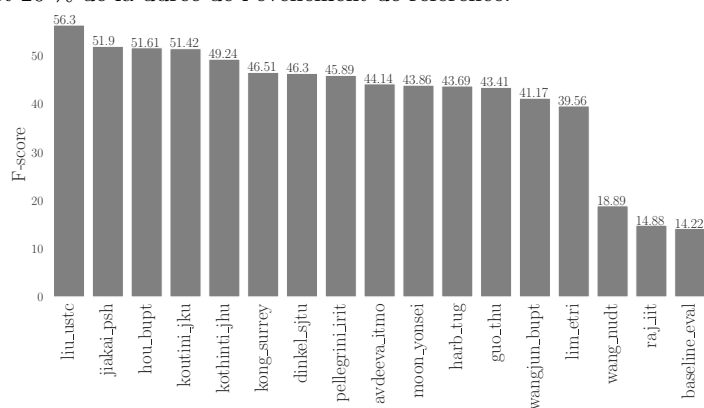
Système	Longs	Courts	Tous	Rang
jiakai_psh	42,6	18,7	32,4	1
kong_surrey	39,7	11,4	24,0	3
liu_ustc	31,4	26,4	29,9	2
lim_etri	31,1	10,7	20,4	9
harb_tug	29,3	12,7	21,6	5
baseline	21,8	2,6	10,8	16

TABLEAU 4.12 – F-mesure par événement 5 meilleurs systèmes pour les classes d’événements longs ainsi que leur classement.

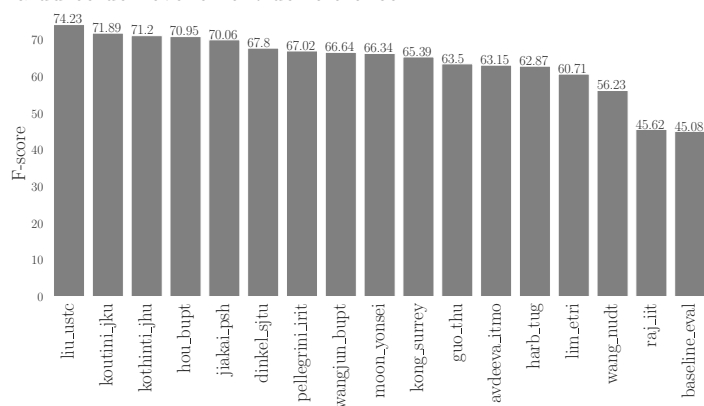
200 ms utilisée dans la tâche (Figure 4.7a). La Figure 4.7b indique que, pour l’ensemble des systèmes, l’augmentation de la tolérance à 1 s améliore la F-mesure par événement qui dépasse parfois 50%, ce qui montre qu’ils souffrent tous de problèmes de segmentation. Les meilleurs systèmes en terme de segmentation sont ceux qui intègrent une segmentation dédiée dans leur implémentation [Kothinti et al., 2018, Liu et al., 2018]. Le



- (a) Tolérance sur les instants de début de 200 ms et tolérance sur les instants de fin égale au maximum entre 200 ms et 20 % de la durée de l'événement de référence.



- (b) Tolérance sur les instants de début de 1 s et tolérance sur les instants de fin égale au maximum entre 1 s et 20 % de la durée de l'événement de référence.



- (c) Tolérance sur les instants de début de 5 s et tolérance sur les instants de fin égale au maximum entre 5 s et 20 % de la durée de l'événement de référence.

FIGURE 4.7 – Performance de segmentation (détection des frontières temporelles sans tenir compte de la classe) selon la tolérance temporelle choisie.

système classé premier a quant à lui utilisé un modèle de professeur moyen qui lui permet d'obtenir une bonne performance de classification tout en gardant une performance de segmentation correcte. Le modèle de professeur moyen est désormais un système de référence pour la détection d'événements sonores [JiaKai, 2018]. Des analyses supplémentaires ont été faites pour comparer les résultats avec une métrique par événement et une métrique par segment ainsi que des analyses de performance de détection d'instant de début et de fin mais celles-ci ne sont pas détaillées dans ce manuscrit [Serizel and Turpault, 2019]⁶. En résumé, l'analyse des systèmes soumis en 2018 nous a permis d'identifier un problème de segmentation important pour l'ensemble des systèmes. Afin de résoudre ce problème à un coût raisonnable, nous avons décidé d'intégrer des données synthétiques fortement annotées dans l'ensemble d'apprentissage en 2019.

4.2.2 DCASE 2019

En 2019, l'utilisation de données synthétiques non seulement à l'apprentissage mais à l'évaluation a permis une analyse plus détaillée des systèmes soumis qu'en 2018. En effet, nous avons créé des jeux de données d'évaluation avec des paramètres spécifiques afin d'analyser le comportement des systèmes soumis sur des problèmes isolés. Les participants à la tâche ont dû détecter les événements présents dans ces jeux de données synthétiques au même titre que dans les données enregistrées. Ces jeux de données synthétiques n'ont pas été utilisés pour le classement final des systèmes mais uniquement pour réaliser cette analyse.

Pour rappel le jeu de données synthétiques d'apprentissage est construit de la façon suivante (*cf.* Algorithme 3.2 de la partie 3.2.4) :

- (1) durée des clips de 10 s ;
- (2) bruit de fond choisi parmi la banque de bruits de fond ;
- (3) choix des événements sonores d'intérêt à partir des distributions de co-occurrence ;
- (4) positionnement temporel aléatoire des événements ;
- (5) augmentation avec changement de ton (de -3 à 3 demi-tons) ;
- (6) rapport signal-à-bruit tiré de façon uniforme entre 6 et 30 dB par événement par rapport au bruit de fond ;
- (7) sans réverbération.

Les différents jeux de données synthétiques que nous avons conçus pour l'évaluation sont décrits ci-dessous. Les numéros cités indiquent les modifications effectuées par rapport à l'algorithme de génération utilisé à l'apprentissage.

- Le jeu « **S_eval** » utilise la même procédure de génération et les mêmes paramètres que le jeu de données synthétiques d'apprentissage, mais appliqué aux données d'évaluation.

6. L'ensemble des résultats ont été partagés sur cette page pour aider les participants du challenge suivant (2019) : <https://turpaultn.github.io/dcase2018-results/>.

- Les jeux « **500 ms** », « **5500 ms** » et « **9500 ms** » sont des jeux de données pour lesquels un seul événement d'intérêt a été choisi (modifie (3)). L'instant de début de cet événement est fixé respectivement à $0,5 \text{ s} \pm 250 \text{ ms}$, $5,5 \text{ s} \pm 250 \text{ ms}$ et $9,5 \text{ s} \pm 250 \text{ ms}$ (modifie (6)).
- Les jeux « **fbsnr_30 dB** », « **fbsnr_24 dB** », « **fbsnr_15 dB** », « **fbsnr_0 dB** » sont des jeux de données pour lesquels le rapport signal-à-bruit par événement est tiré de façon uniforme entre 6 et 30 dB, entre 0 et 24 dB, entre -9 et 15 dB, ou entre -24 et 0 dB, respectivement (modifie (5)). Les jeux de données « **S_eval** » et « **fbsnr_30 dB** » sont identiques, leur nom est modifié pour une présentation plus claire des résultats.
- Les jeux de données « **phone_play** », « **phone_record** », « **clipping** », « **compression** », « **highpass** » et « **lowpass** » sont des jeux de données pour lesquels des dégradations ont été appliquées après la génération. Ces dégradations représentent respectivement un enregistrement joué par téléphone, un enregistrement par un micro de faible qualité, un écrêtage, une compression, un filtrage par un filtre passe-haut et un filtrage par un filtre passe bas⁷.

Pour analyser l'ensemble des systèmes plus en détail, nous utilisons tout d'abord les jeux de données pour lesquels le rapport signal-à-bruit varie afin de comprendre l'impact du bruit de fond sur les systèmes proposés. La Figure 4.8 présente les résultats pour les 10 meilleurs systèmes en fonction du rapport signal-à-bruit dans le jeu de données d'évaluation généré. Les performances sont plutôt similaires entre systèmes, ce qui indique qu'aucun d'entre eux ne se démarque en terme de robustesse à la diminution du signal-à-bruit. De façon intéressante, les résultats sur le jeu de données **fbsnr_15dB** sont de l'ordre de grandeur de ceux sur les données réelles. Ce jeu de données a un rapport signal-à-bruit moyen compris entre -9 et 15 dB, ce qui suggère que les données collectées depuis Audioset ont un rapport signal-à-bruit du même ordre de grandeur. Sans surprise, lorsque le rapport signal à bruit est négatif (**fbsnr_0dB**), les performances de l'ensemble des systèmes s'effondrent.

Nous proposons ensuite d'analyser la performance de segmentation des différents systèmes. La Figure 4.9 présente la performance de segmentation sur les classes d'événements longs pour les 3 meilleurs systèmes de segmentation et le système de référence (« baseline ») en fonction de la position temporelle de l'événement. De façon surprenante, les 3 meilleurs systèmes voient leur performance de détection des événements longs se dégrader fortement lorsque ceux-ci sont placés en fin de clip (**9500 ms**). Le système de référence (« baseline ») est quant à lui plus robuste aux positions des événements mais a une performance plus faible. Afin de mieux comprendre ce résultat, les instants de début et de fin des événements longs du jeu d'apprentissage synthétique sont présentés dans la Figure 4.10a. Les instants de début sont distribués presque uniformément, alors que les instants de fin sont presque exclusivement à la fin du clip. Dans la Figure 4.10b nous présentons les instants de fin des événements longs qui apparaissent dans les jeux de

7. Audio Degradation Toolbox avec les paramètres par défaut : <https://code.soundsoftware.ac.uk/projects/audio-degradation-toolbox> [Mauch and Ewert, 2013]

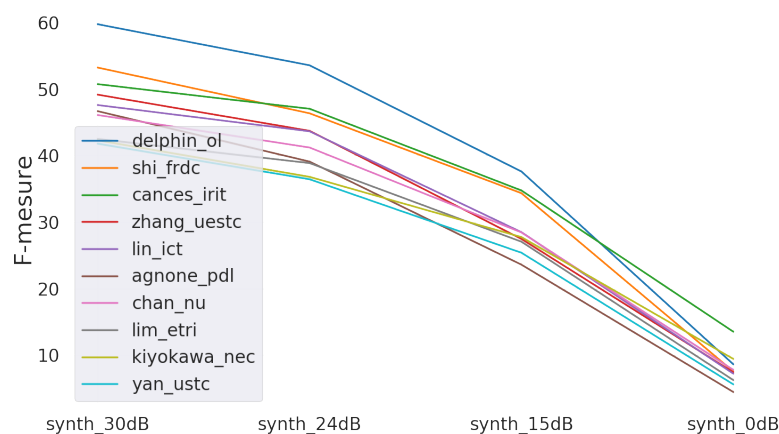


FIGURE 4.8 – F-mesure par événement (%) des 10 meilleurs systèmes soumis en 2019 en fonction du rapport signal-à-bruit.

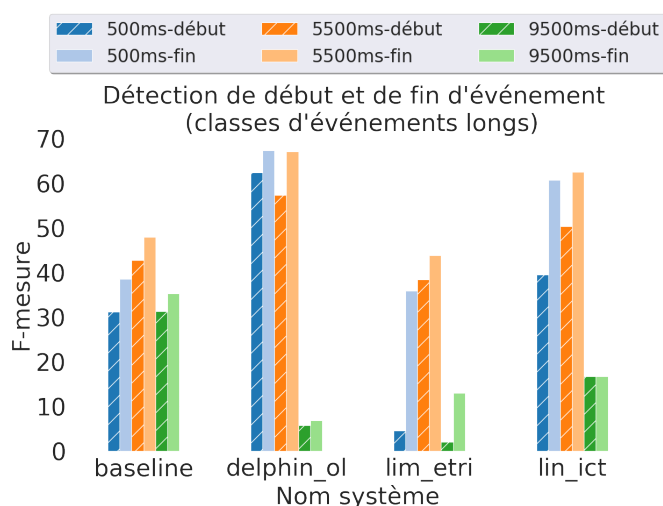
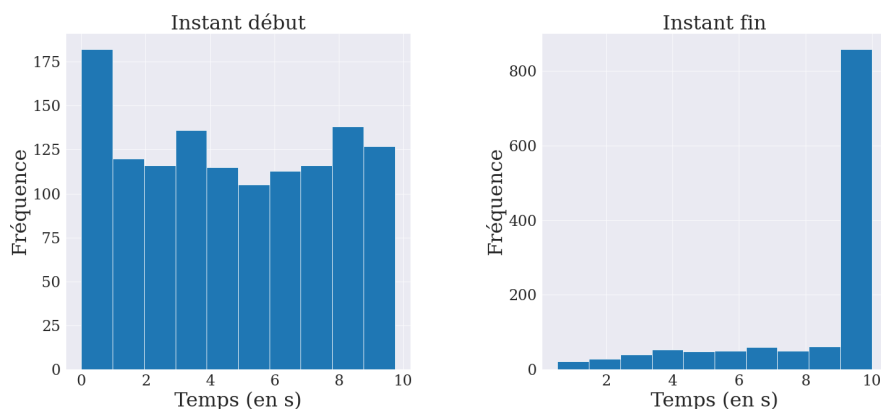


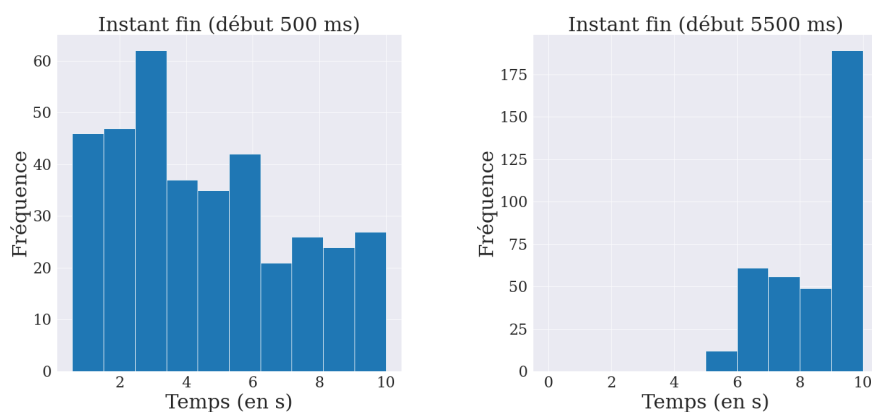
FIGURE 4.9 – Performance de segmentation des classes d'événements longs en fonction de la localisation temporelle de l'événement.

données **500 ms** et **5500 ms**. Nous constatons que, pour le jeu de données **500 ms**, les instants de fin des événements longs sont distribués tout au long du clip alors que, pour le jeu de données **5500 ms**, les instants de fin des événements longs sont concentrés à la fin des clips. Quant au jeu de données **9500 ms**, ses événements longs ont tous la fin du clip comme instant de fin. La diminution de performance observée sur le jeu de données **9500 ms** n'est donc pas due à la différence de distribution des instants de début et de fin entre les jeux d'apprentissage et d'évaluation, puisque la performance est élevée sur le jeu **500 ms** malgré une distribution des instants de fin très différente. Nous attribuons cette chute de performance à l'utilisation de méthodes de post-traitement spécifiques à chaque classe. Les paramètres choisis pour les classes d'événements longs (seuil de dé-

tection et longueur du filtre médian) font que les systèmes ne tendent à détecter que les événements longs dont la durée est strictement supérieure à la moitié de la longueur du filtre médian. Cette condition n'est pas respectée lorsque l'instant de début est proche de la fin du clip. Le système de référence emploie le même post-traitement pour l'ensemble des événements et n'est donc pas sujet à ce biais.



(a) Instants de début et de fin des événements longs dans le jeu d'apprentissage synthétique.



(b) Instants de fin des événements longs dans les jeux d'évaluation **500 ms** et **5500 ms**.

FIGURE 4.10 – Distribution des instants de début et de fin des événements longs dans les jeux d'apprentissage et d'évaluation synthétiques.

Pour finir l'analyse des systèmes soumis en 2019, le Tableau 4.13 présente les performances des systèmes soumis sur les ensembles synthétiques sujets à différentes dégradations. Dans l'ensemble, les systèmes ne semblent pas très robustes aux dégradations. Le système de Shi [2019] est le plus robuste aux dégradations appliquées. Il utilise un modèle de « professeur moyen » et l'augmentation de données « mixup », qui mélange les fichiers audio et leurs annotations [Shi et al., 2020]. Ce modèle utilise à l'apprentissage des données audio très différentes des données d'évaluation (puisque elles sont mixées entre

Système	fbsnr_24dB	phone play	phone record	clipping	compression	high pass	low pass
Agnone, PDL	39,1%	15,4%	9,2%	14,6%	29,6%	8,5%	0,9%
Cances, IRIT	47,1%	25,7%	35,8%	42,6%	44,3%	19,2%	1,2%
Chan, NU	41,2%	25,9%	17,5%	22,8%	33,4%	19,3%	1,2%
Delphin-Poulat, OL	53,6%	32,9%	23,7%	29,5%	48,2%	23,3%	4,8%
Kiyokawa, NEC	36,8%	33,9%	21,9%	35,6%	40,2%	22,1%	4,2%
Lim, ETRI	38,9%	26,9%	30,3%	39,7%	48,1%	15,4%	0,7%
Lin, ICT	43,7%	22,4%	9,3%	19,8%	35,3%	17,6%	0,5%
Shi, FRDC	46,4%	35,0%	36,4%	48,3%	54,1%	17,4%	4,0%
Yan, USTC	36,5%	22,1%	21,5%	18,3%	32,7%	16,6%	1,0%
Zhang, UESTC	43,7%	21,8%	15,3%	24,6%	41,4%	14,1%	1,7%
Score moyen (tous systèmes)	33,9%	22,0 %	16,4%	21,6%	31,4%	15,8%	1,7%

TABLEAU 4.13 – F-mesure par événement (%) des 10 meilleurs systèmes soumis en 2019 sur les données d'évaluation synthétiques dégradées.

elles, les événements et les bruits de fonds se superposent sans que ce soit nécessairement réaliste), ce qui pourrait expliquer sa bonne performance sur les jeux de données dégradées. L'ensemble des systèmes s'adapte plutôt bien à la compression (**compression**), ce qui peut être expliqué par l'utilisation de vidéos Youtube à l'apprentissage qui ont déjà pu être compressées. La faible performance des systèmes après filtrage passe-haut (**highpass**) ou passe-bas (**lowpass**) peut s'expliquer par la nature particulièrement destructive de ces dégradations (perte de certaines composantes fréquentielles). Ces résultats permettent tout de même de montrer la complémentarité entre les basses et les hautes fréquences pour reconnaître nos classes d'intérêt qui sont souvent des sons complexes. L'utilisation d'une réponse impulsionnelle d'un micro de téléphone accompagnée de compression dynamique (**phone_record**) a un impact plus négatif que l'utilisation d'une réponse impulsionnelle de haut-parleur de téléphone (**phone_play**) ce qui peut paraître surprenant car de nombreuses vidéos sont enregistrées à partir d'un téléphone. Finalement, certains systèmes ([Cances et al., 2019, Kiyokawa et al., 2019, Lim et al., 2019, Shi et al., 2019]) semblent robustes à la saturation du signal (**clipping**) alors que les autres semblent fortement impactés par les artefacts introduits par la saturation.

En résumé, la comparaison des différents systèmes sur des jeux de données spécifiques nous a permis de montrer les spécificités ou les lacunes de certains systèmes. Cette analyse est particulièrement utile pour le choix d'un système qui nécessiterait certaines spécificités afin d'être utilisé dans un scénario réel.

4.2.3 DCASE 2020

En 2020, nous avons proposé d'utiliser la séparation de sources pour traiter le problème de la polyphonie. Nous avons également proposé d'utiliser les réponses impulsionnelles des salles du jeu de données FUSS pour réverbérer chaque événement du clip et créer une scène sonore plus réaliste. Nous avons enfin proposé d'utiliser le PSDS comme métrique secondaire. Comme en 2019, nous avons proposé de multiples jeux de données

synthétiques permettant l'évaluation sur des problèmes isolés. Les jeux de données sont inspirés de l'analyse de 2019 qui a identifié des problèmes de localisation temporelle et de segmentation des événements en fonction de leur durée. Les jeux de données créés sont les suivants.

- Le jeu de données « **S_eval** » utilise les mêmes paramètres de génération que les données d'apprentissage. C'est notre jeu d'évaluation synthétique de référence.
- Les jeux de données « **500 ms** », « **5500 ms** » et « **9500 ms** » restent identiques à ceux proposés en 2019 afin d'analyser l'impact de la position des événements dans les clips audio.
- Le jeu de données « **60 s** » utilise le même nombre d'événements par clip et le même intervalle de rapport signal-à-bruit que le jeu de données **S_eval**, mais dans des clips de durée 60 s (modifie le (1) de l'Algorithme 3.2). Ce jeu de données permet d'identifier les problèmes surgissant lors de l'analyse de clips audio de durée plus longue et de densité temporelle d'événements plus faible, comme ce peut être le cas lorsque l'on cherche à détecter des événements toute la journée ou à différents moments cibles de la journée.
- Le jeu de données « **one event** » contient un seul événement par clip (modifie (3)), à une position aléatoire qui nous permet de faire une analyse précise sur la durée des événements [Turpault et al., 2020a].
- Le jeu de données « **S_eval** » se présente sous 8 variantes, qui incluent du bruit de fond (modifie (2)), de la réverbération (modifie (7)) ou les deux et des événements non-cibles issus de **FUSS** en plus. Les variantes « **TNTSNR_15** » et « **TNTSNR_0** » contiennent des événements non-cibles avec un rapport de puissance entre les événements cibles et les événements non-cibles de 15 dB et 0 dB, respectivement. Les variantes « **low_reverb** » et « **high_reverb** » correspondent respectivement à une réverbération avec une réponse impulsionnelle de salle tronquée à 200 ms et une réverbération avec la réponse impulsionnelle non tronquée.

Le Tableau 4.14 présente les résultats des 10 meilleures équipes et du système de référence (« baseline ») sur le jeu d'évaluation synthétique de référence et le jeu contenant des clips de 60 s. La dernière colonne du tableau indique que la majorité des systèmes ont des difficultés à détecter des événements sonores dans des clips audio de longue durée. Trois systèmes ont des performances comparables sur les clips de 10 s et de 60 s [Ebberts and Haeb-Umbach, 2020, Huang et al., 2020a, Liu et al., 2020] et un système améliore même sa performance [Hao et al., 2020]. Cette différence de comportement peut s'expliquer par l'utilisation de seuils de détection permettant de limiter l'impact de la densité des événements dans le clip. Une hypothèse naturelle pour expliquer la faible performance des autres systèmes sur le jeu **60 s** serait l'apparition d'un nombre important de faux positifs. Cette hypothèse est réfutée par la Figure 4.11, qui indique au contraire que les systèmes possèdent une précision élevée mais un rappel faible, ce qui traduit un nombre important de faux négatifs. Ce rappel faible peut être attribué à la mauvaise performance de segmentation de ces systèmes ou au fait que F-mesure par événement n'accepte pas de coupure dans les événements détectés et que, plus l'événement est long, plus une

Nom d'équipe	F-mesure par événement		Différence 60s-synth_eval
	S_eval	60 s	
Miyazaki	56,5	2,9	-53,6
Hao	38,4	53,0	14,7
Ebbers	54,4	53,0	-1,5
Koh	51,4	3,3	-48,1
Yao	52,9	2,7	-50,2
CTK	50,9	39,9	-11,1
Liu	45,3	41,7	-3,6
Zhenwei	36,4	0,1	-36,3
Huang	36,2	35,8	-0,4
Cornell	47,8	3,4	-44,4
Baseline	46,3	3,0	-43,3

TABLEAU 4.14 – F-mesure par événement (%) des 10 meilleurs systèmes soumis en 2020 sur les jeux de données synthétiques d'évaluation **S_eval** et **60 s** et différence entre ces deux valeurs.

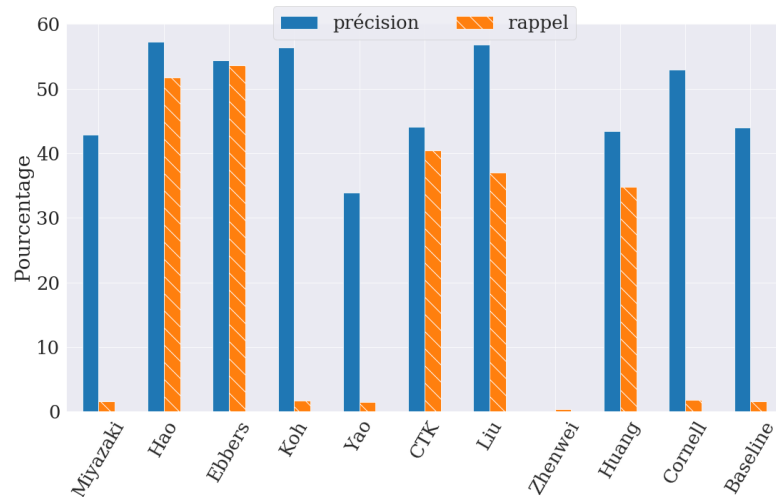


FIGURE 4.11 – Précision et rappel sur le jeu de données **60 s**.

telle coupure devient probable. La réponse à cette question nécessiterait une expérience complémentaire, qui sort du cadre de l'analyse de 2020.

La Figure 4.12 présente les résultats des systèmes sur les jeux de données **500 ms**, **5500 ms** et **9500 ms** pour les classes d'événements longs. Cette analyse permet de savoir si les systèmes soumis en 2020 sont plus robustes que ceux soumis en 2019 à la position des événements dans le clip, car le même jeu de données est utilisé. La perte de performance identifiée en 2019 concernant la détection des événements longs situés en fin de clip a été réduite de manière générale. Cette perte de performance est néanmoins toujours présente, et très importante pour le vainqueur de la tâche [Miyazaki et al., 2020].

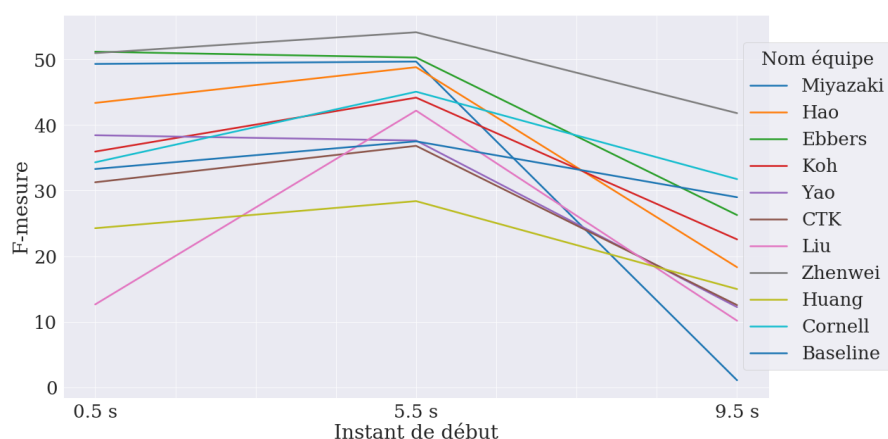


FIGURE 4.12 – F-mesure des classes d'événements longs macro-moyennée en fonction de la localisation temporelle de l'instant de début des événements.

Ce problème continue de s'expliquer par le post-traitement appliqué par les participants (filtrage médian de longueur variable en fonction des classes d'événement par exemple) qui peut être biaisé lorsqu'il est optimisé sur les clips enregistrés.

La Figure 4.13 présente la F-mesure des 10 premières équipes et du système de référence sur le jeu de données **one_event** en fonction de la durée des événements analysés. Cette analyse permet de grouper les événements par durée, peu importe leur classe. Dans ce cas, la F-mesure n'est pas macro-moyennée par classe mais par taille d'événement. Notons que la classe « voix » peut être très présente dans les événements courts et donc engendrer un biais si celle-ci est bien ou mal reconnue par rapport aux autres classes. Cette figure illustre globalement la difficulté des systèmes à détecter les événements de

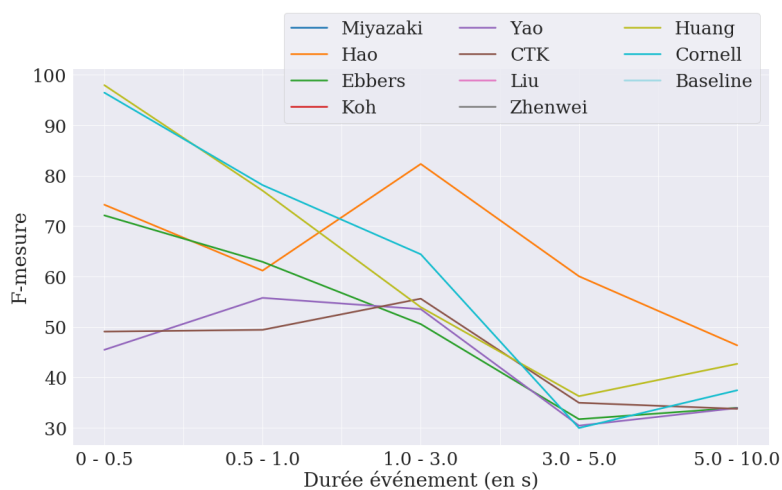


FIGURE 4.13 – F-mesure en fonction de la durée des événements sonores dans le jeu de données **one_event**.

durée supérieure à 3 s, ce qui contraste avec les résultats sur les données enregistrées qui indiquent de meilleures performances pour les classes d'événements longs. Ce problème pourrait s'expliquer par la polyphonie présente dans les enregistrements réels qui rend la détection des événements courts difficile. Certains systèmes sont très performants pour les événements courts [Cornell et al., 2020, Huang et al., 2020a, Koh et al., 2020, Liu et al., 2020] et parmi eux 2 systèmes utilisent la séparation de sources. Le système de Hao et al. [2020] a une performance supérieure à 60% pour l'ensemble des événements de durée inférieure à 5 s et obtient la meilleure performance pour les événements de durée comprise entre 5 et 10 s. Les autres systèmes ont des courbes plutôt semblables, reconnaissant un peu mieux les événements courts que les événements longs avec souvent un point d'inflexion pour les événements d'une durée comprise entre 3 et 5 s.

La Figure 4.14 représente la F-mesure macro-moyennée par événement des 11 systèmes présentés précédemment en fonction du rapport de puissance entre les événements cibles et les événements non-cibles ajoutés. Le jeu de données **S_eval** n'a pas d'événements non-cibles, c'est notre référence. On voit que l'ajout d'événements non-cibles dégrade la performance de tous les systèmes. Ceci peut indiquer une confusion des événements non-cibles et des événements cibles ou bien l'apparition de polyphonie plus importante avec les événements non-cibles. La F-mesure se dégrade de 5 à 10% en présence d'événements non-cibles avec un rapport de puissance de 15 dB et de 15 à 25% en présence d'événements non-cibles de même niveau sonore que les événements cibles.

La Figure 4.15 représente la F-mesure des systèmes en faisant varier la réverbération appliquée aux événements cibles pour les jeux de données ne contenant pas d'événements non-cibles. De manière générale, la performance se dégrade en présence de réverbération. Il y a cependant très peu de différence entre la réverbération tronquée (*low reverb*) et complète (*high reverb*) ce qui semble indiquer que les réflexions précoces sont responsables

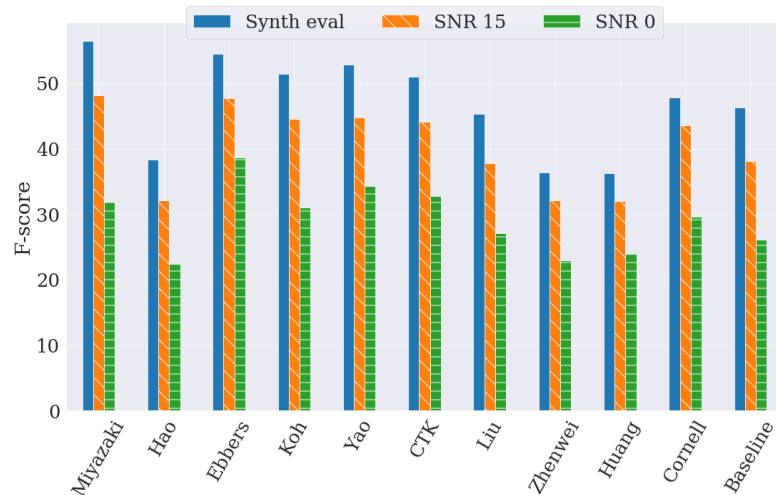


FIGURE 4.14 – F-mesure (%) en fonction du rapport de puissance entre les événements cibles et les événements non-cibles.

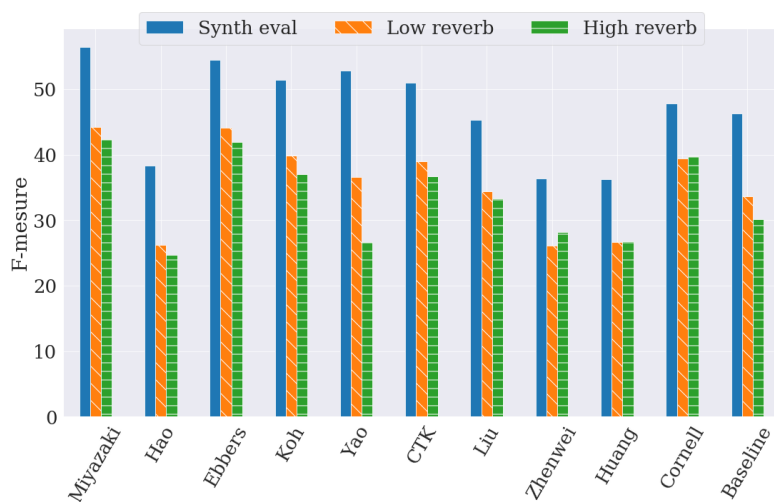


FIGURE 4.15 – F-mesure (%) en fonction de la réverbération.

en grande partie de la dégradation.

En résumé, l’analyse détaillée des systèmes soumis à la tâche chaque année nous a permis de partager avec les participants les points d’amélioration possibles. Cela nous a également permis de faire évoluer la définition du problème, de proposer de nouvelles solutions chaque année et d’inspirer de multiples nouvelles recherches [Chan and Chin, 2020, de Benito-Gorrón et al., 2021, Sudo et al., 2020].

4.3 Conclusion

Dans ce chapitre, nous avons présenté une partie des contributions scientifiques réalisées dans le cadre de l’organisation de la Tâche 4 du Challenge DCASE. La tâche a été définie de manière à promouvoir des solutions à un grand nombre de problèmes pouvant apparaître dans les scénarios de détection d’événements sonores réels. L’identification de nouveaux problèmes passe par le développement d’un système de référence et la réflexion autour de pistes d’amélioration, notamment grâce l’analyse des systèmes soumis chaque année. La campagne d’évaluation nécessite une réflexion importante au préalable permettant la définition d’un jeu de données qui permet de créer un classement des systèmes (ce n’est pas le but principal). Nous avons fait le choix de définir le jeu de données [DESED](#) de manière à ce qu’il permette une analyse détaillée des différentes soumissions pour faire ressortir les problèmes communs ou spécifiques aux différents systèmes soumis. Les problèmes communs identifiés par l’analyse des soumissions nous ont permis de faire évoluer la définition du problème et l’utilisation des données du corpus [DESED](#), de proposer de nouvelles données et d’introduire de nouvelles approches de résolution de la tâche (comme la séparation de sources) et de nouvelles métriques d’évaluation (comme le [PSDS](#) qui devient métrique officielle en 2021).

5 Apprentissage de représentation semi-supervisé

Ce chapitre est consacré à l'apprentissage de représentation de façon semi-supervisée. Dans les applications réelles, le scénario semi-supervisé est courant puisqu'il est trop coûteux d'annoter l'ensemble des données disponibles. Un moyen de réduire les coûts est d'utiliser une représentation de haut niveau commune à diverses applications et permettant de réduire la complexité du classifieur utilisé. En effet, ce dernier est appris uniquement sur les données de la tâche d'intérêt et bien souvent de manière supervisée ce qui réduit la quantité de données exploitables. L'apprentissage d'une telle représentation permet d'exploiter des données non spécifiques à la tâche d'intérêt pouvant apporter des informations supplémentaires au classifieur. Dans ce chapitre nous nous concentrons sur l'apprentissage de représentation sur un jeu de données de petite taille spécifique à nos applications visées par une méthode semi-supervisée basée sur l'apprentissage par triplets. Nous comparons cette approche à une méthode d'apprentissage supervisée sur un jeu de données très important contenant des classes d'événements générales non spécifiques à notre application et à une méthode utilisant le même modèle que l'apprentissage par triplets mais basée sur la méthode du « professeur moyen ». L'apprentissage par triplets est inspiré du travail réalisé par [Jansen et al. \[2017\]](#) qui apprend des représentations de façon supervisée ou non supervisée. Nous proposons d'utiliser cette méthode de façon semi-supervisée. Nous proposons également une nouvelle méthode non supervisée de tirage de l'exemple positif. La représentation obtenue par apprentissage d'un classifieur général est celle de VGGish appris sur Audioset [[Hershey et al., 2017](#)]. Il faut noter qu'il s'agit ici d'un abus commun de langage puisque cette représentation est en réalité apprise sur le corpus Youtube-8M contenant plus de 6 millions de clips audio alors qu'Audioset contient « seulement » 2 millions de clips audio.

5.1 Description du problème semi-supervisé

Dans ce chapitre, nous nous intéressons à un problème d'étiquetage d'événements sonores. Nous avons un petit jeu de données faiblement annotées $\mathcal{I}_A = \{(\mathbf{X}_l, \mathbf{w}_l)\}_{l=1}^L$ et un grand jeu de données non annotées $\mathcal{I}_{NA} = \{(\mathbf{X}_u)\}_{u=1}^U$ où L et U sont respectivement le nombre de données annotées et non annotées. $\mathbf{X} \in \{\mathcal{I}_A, \mathcal{I}_{NA}\}$ est la représentation de bas niveau (temps-fréquence) d'un clip audio et $\mathbf{w} = [w_1, \dots, w_C]$ est le vecteur d'annotation de ce clip avec $w_c \in \{0, 1\}$ indiquant si la classe c est présente dans le clip ou non. Notons que plusieurs classes peuvent être présentes. On définit $\mathcal{I} = \mathcal{I}_A \cup \mathcal{I}_{NA}$ comme le jeu de données contenant les données annotées et non annotées. L'apprentissage par

triplets pourrait être fait au niveau du clip mais, dans notre cas, nous divisons le clip audio en segments de 0,96 s comme cela a été fait par [Jansen et al. \[2017\]](#). Pour l'utilisation des annotations, nous faisons l'hypothèse que chaque segment contient les mêmes classes d'événements que le clip entier, ce qui n'est pas vrai en pratique (surtout pour les événements courts) et donc implique des erreurs d'annotation des segments. L'analyse détaillée de ce type de problème sera présentée dans le chapitre 6. On définit \mathbf{X}_τ un segment de 0,96 s extrait du clip \mathbf{X} où $\tau \in \{1, \dots, 10\}$ représente l'indice temporel du segment dans le clip. Les fonctions de coût et autres formules introduites dans ce chapitre ne dépendant pas de τ , nous omettons cet indice par la suite par souci de clarté.

5.1.1 Apprentissage par triplets

Premièrement, nous cherchons à apprendre une représentation à partir des données non annotées et annotées en utilisant l'apprentissage par triplets. Le but de l'apprentissage par triplets [[Wang et al., 2014](#)] est d'apprendre une représentation de haut niveau discriminante. La représentation est optimisée afin de rapprocher chaque exemple issu du jeu de données appelé « ancre » (\mathbf{X}^a) d'un exemple « positif » (\mathbf{X}^p) et de l'éloigner d'un exemple « négatif » (\mathbf{X}^n). Les exemples positifs et négatifs peuvent être issus du jeu de données ou non et servent optimiser la position de l'ancre dans l'espace de représentation. La fonction de coût à minimiser est égale à

$$\sum_{\mathbf{X}_i \in \mathcal{J}} [\|E(\mathbf{X}_i^a) - E(\mathbf{X}_i^p)\|_F^2 - \|E(\mathbf{X}_i^a) - E(\mathbf{X}_i^n)\|_F^2 + \delta]_+ \quad (5.1)$$

où $E(\mathbf{X})$ est la représentation de haut niveau de \mathbf{X} , $[\cdot]_+$ est le coût charnière (*hinge loss*), $\|\cdot\|_F$ est la norme de Frobenius et δ est un paramètre de marge. Cette fonction de coût diffère du coût par paires utilisé dans les réseaux siamois [[Bromley et al., 1993](#)] par le fait que dans les triplets le nombre d'exemples positifs et négatifs associés à chaque exemple du jeu de données est équilibré.

5.1.2 Stratégie de tirage supervisée

Différentes stratégies de tirage ont été proposées dans la littérature afin d'exploiter les données et les annotations [[Jati et al., 2019](#), [Schroff et al., 2015](#), [Wang et al., 2014](#)]. Chaque stratégie mène à une représentation de haut niveau différente. Une stratégie simple appliquée par [Jansen et al. \[2017\]](#) repose sur l'utilisation de données annotées seulement. Dans cette stratégie, les exemples positifs sont choisis de façon aléatoire parmi l'ensemble des exemples avec au moins une classe d'événement commune avec l'exemple ancre. Les exemples négatifs sont choisis aléatoirement parmi l'ensemble des exemples sans aucune classe d'événement commune avec l'exemple ancre. La Figure 5.1 représente un exemple de tirage de triplets de façon supervisée. Notons que l'utilisation de segments de 0,96 s dont les annotations sont similaires à celles du clip implique que les triplets peuvent avoir des exemples ancres et positifs ne contenant en réalité pas la même classe. Cependant, les exemples négatifs sont bien négatifs. Le pire scénario serait que l'exemple

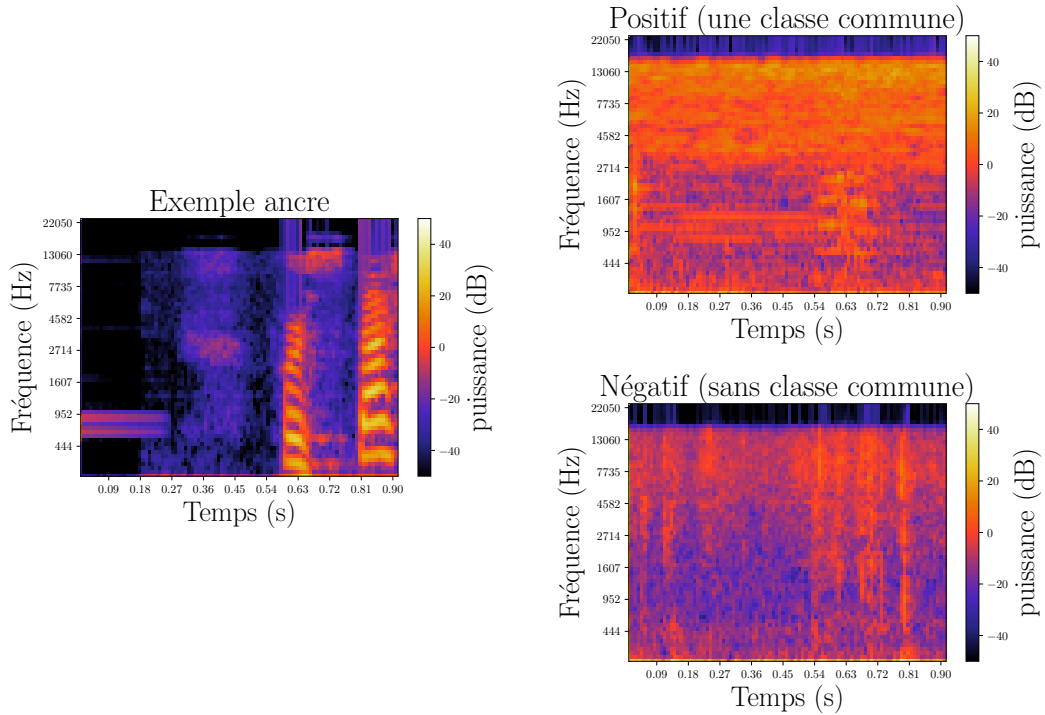


FIGURE 5.1 – Création des triplets de façon supervisée.

ancre ne contienne que du bruit de fond et l'exemple négatif aussi alors que l'exemple positif contiendrait l'événement d'intérêt.

5.1.3 Stratégie de tirage non supervisée basée sur des transformations

Jansen et al. [2017] ont aussi proposé d'utiliser une stratégie de tirage non supervisée. Dans cette stratégie, l'exemple positif est une version transformée de l'exemple ancre et le *semi-hard mining* est utilisé pour tirer l'exemple négatif. Nous détaillons ci-dessous les quatre transformations proposées pour l'exemple positif et le processus de *semi-hard mining* pour l'exemple négatif. Ces quatre transformations sont illustrées dans la Figure 5.2.

Bruit gaussien L'exemple positif est créé en ajoutant du bruit gaussien de moyenne 0 et d'écart type σ à l'exemple ancre.

Translation temporelle et fréquentielle L'exemple positif est créé par translation circulaire de l'exemple ancre sur l'axe temporel. Le nombre de fenêtres de translation est tiré aléatoirement de manière uniforme sur $[0, T_s]$ où T_s est le nombre maximal de fenêtres de translation possible. Cet exemple subit ensuite une translation sur l'axe fréquentiel. Le nombre de bandes fréquentielles de translation est tiré aléatoirement de manière uniforme sur $[-S, S]$ où S est le nombre maximal de bandes fréquentielles de translation possible.

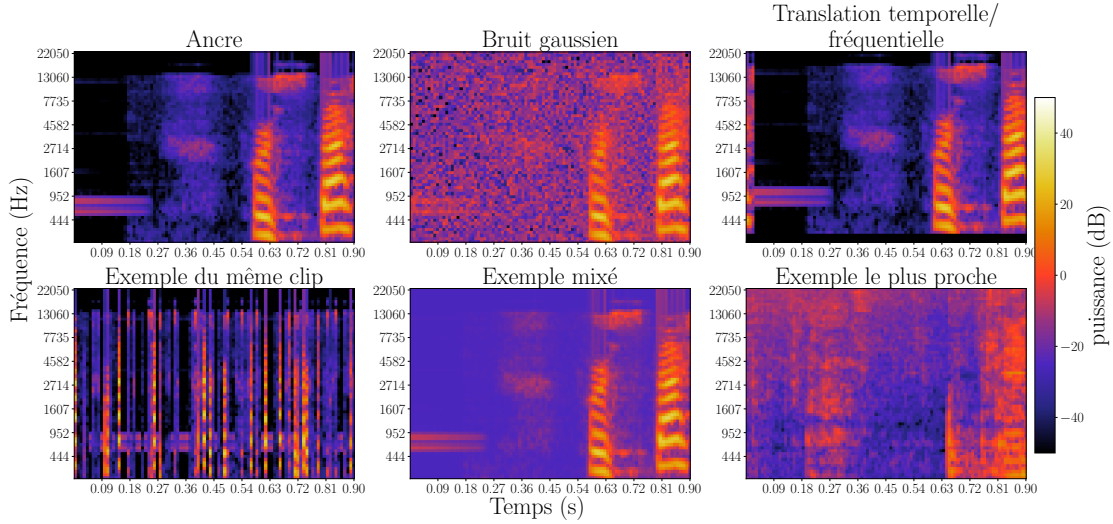


FIGURE 5.2 – Exemples positifs obtenus par les stratégies appliquées aux données non annotées.

Les bandes fréquentielles manquantes sont remplacées par des 0.

Proximité temporelle L'exemple positif est un exemple issu du même clip audio que l'exemple ancre avec avec une différence de temps inférieure à Δt entre les deux.

Mélange d'exemples L'exemple positif est un mélange de l'ancre et d'un autre exemple $\mathbf{X} \in \mathcal{J}$ tel que $\mathbf{X}^p = \mathbf{X}^a + \alpha[\text{énergie}(\mathbf{X}^a)/\text{énergie}(\mathbf{X})] \mathbf{X}$ où $\text{énergie}(\cdot)$ est l'énergie du signal. L'exemple en question est choisi de façon aléatoire dans le jeu de données.

Semi-hard mining L'exemple négatif tiré est le plus proche de l'ancre dans l'espace de représentation de haut niveau, tout en restant plus éloigné de l'ancre que l'exemple positif [Schroff et al., 2015] :

$$\mathbf{X}^n = \underset{\substack{\mathbf{X} \in \mathcal{J}_{\text{NA}} \setminus \{\mathbf{X}^a, \mathbf{X}^p\} \\ \|E(\mathbf{X}^a) - E(\mathbf{X}^p)\|_2^2 < \|E(\mathbf{X}^a) - E(\mathbf{X})\|_2^2}}{\operatorname{argmin}} \|E(\mathbf{X}) - E(\mathbf{X}^a)\|_2^2. \quad (5.2)$$

5.2 Stratégies semi-supervisées proposées

La stratégie supervisée proposée par Jansen et al. [2017] utilise seulement les données annotées, alors que leur approche non supervisée n'utilise aucune annotation. Nous proposons deux stratégies semi-supervisées de tirage des exemples, où les deux jeux de données annotées et non annotées sont utilisés. Lorsque l'annotation de l'ancre est disponible, l'exemple positif est tiré de façon supervisée et l'exemple négatif est tiré de façon supervisée ou par *semi-hard mining*. Lorsque l'annotation de l'ancre n'est pas disponible,

l'exemple positif est choisi de façon non supervisée soit en transformant l'exemple ancre par l'une des quatre transformations ci-dessus (bruit gaussien, translation, proximité, mélange) soit en choisissant l'exemple le plus proche dans l'espace d'entrée, et l'exemple négatif est nécessairement tiré par *semi-hard mining*.

Le choix de l'exemple le plus proche dans l'espace d'entrée consiste à prendre comme exemple positif l'exemple le plus proche de l'ancre au sens de la norme L_2 sur les représentations de bas niveau :

$$\mathbf{X}^P = \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{X}^a\|_2^2. \quad (5.3)$$

Ce choix est dans la lignée des travaux de [Thulasidasan and Bilmes \[2017\]](#) qui ont utilisé la distance entre les représentations de bas niveau pour régulariser l'apprentissage de représentation de haut niveau. Dans notre cas, on peut considérer que la représentation de bas niveau d'un exemple transformé par une des méthodes ci-dessus est proche de sa représentation de bas niveau originelle. Utiliser l'exemple dont la représentation de bas niveau est la plus proche de celle de l'ancre traduit donc implicitement la contrainte que deux points proches dans l'espace de représentation de bas niveau devraient l'être aussi dans l'espace de représentation de haut niveau.

La Figure 5.3 illustre la motivation sous-jacente à l'usage de l'exemple le plus proche dans l'espace d'entrée au lieu des exemples transformés. Lorsque nous avons de nombreux exemples dans une petite partie de l'espace d'entrée, prendre une version transformée de l'exemple ancre comme exemple positif est intéressant mais, lorsque les données d'entrée sont éparpillées dans l'espace d'entrée, il peut être intéressant d'utiliser l'exemple le plus proche dans l'espace d'entrée. En effet, l'un des problèmes possibles de l'apprentissage par triplets est que l'exemple positif est parfois trop proche de l'ancre dans l'espace de représentation de haut niveau, ce qui empêche de trouver un exemple négatif suffisamment proche (dont la distance à l'ancre soit proche de la distance entre l'ancre et l'exemple positif plus la marge) pour permettre l'optimisation du modèle. Lorsque les données sont éparpillées dans l'espace d'entrée, l'utilisation de l'exemple le plus proche dans l'espace d'entrée augmente la probabilité de trouver un exemple négatif suffisamment proche pour

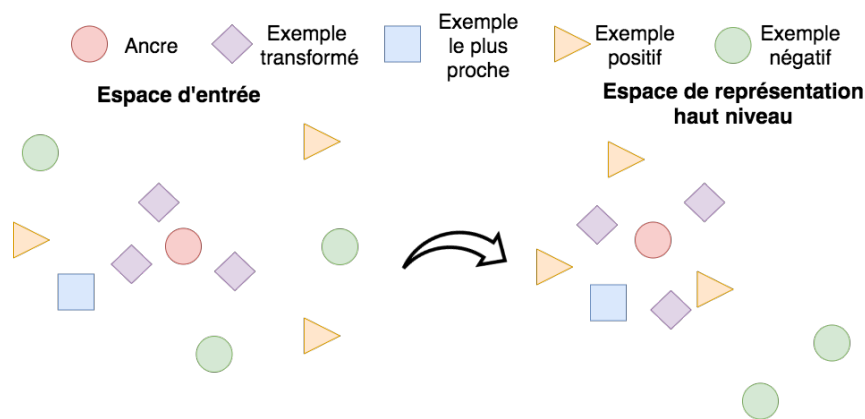


FIGURE 5.3 – Illustration des stratégies de tirage.

permettre l'optimisation. L'inconvénient de cette approche est cependant qu'il n'y a pas de garantie que l'exemple positif partage une même classe d'événement avec l'exemple ancre. Dans la Figure 5.3, c'est ce deuxième cas qui est représenté.

5.3 Protocole expérimental

5.3.1 Jeu de données

Dans les expériences de ce chapitre, nous utilisons le jeu de données [DESED](#) dans sa version de 2018 [[Serizel et al., 2018](#)], c'est-à-dire un jeu de données contenant uniquement des données réelles. Le jeu de données d'apprentissage est constitué des données faiblement annotées (1578 clips) et des données non annotées du domaine (14412 clips). Les données de test contiennent 288 clips et les données d'évaluation contiennent 880 clips issus du jeu d'évaluation d'AudioSet. La tâche considérée étant l'étiquetage d'événements sonores, les annotations fortes des données de test et d'évaluation sont transformées en annotations faibles (toute classe d'événement présente dans une ou plusieurs fenêtres d'un clip est incluse dans l'annotation faible du clip). Nous utilisons les données de test pour valider l'apprentissage et les données d'évaluation pour évaluer les performances des systèmes.

5.3.2 Stratégies de tirage utilisées

Lors de l'apprentissage par triplets, nous comparons 5 stratégies de tirage des exemples avec un système de référence simple. Nous décrivons les 5 stratégies de tirage des exemples dans le Tableau 5.1. Le tirage de l'exemple positif peut se faire par transformation de l'exemple ancre, en prenant l'exemple le plus proche dans l'espace d'entrée ou en utilisant un exemple ayant une classe d'événement commune avec l'exemple ancre. Le tirage de l'exemple négatif peut se faire en utilisant un exemple sans aucune classe d'événement commune avec l'exemple ancre ou par *semi-hard mining*. Les différentes stratégies combinant ces possibilités sont les suivantes.

- La stratégie S1 est la stratégie de tirage non supervisée proposée par [Jansen et al. \[2017\]](#). Dans cette expérience, nous utilisons les données annotées et non annotées de notre jeu de données mais les annotations ne sont pas utilisées.
- La stratégie S2 est la stratégie semi-supervisée qui utilise les transformations pour choisir l'exemple positif lorsque l'exemple ancre n'a pas d'annotation. Cette stratégie nous permet de déterminer l'impact de l'apprentissage semi-supervisé. Nous utilisons les données annotées pour tirer les exemples de manière supervisée et les données non annotées pour tirer les exemples de manière non supervisée.
- La stratégie S3 est la stratégie semi-supervisée utilisant le plus proche voisin dans l'espace d'entrée pour choisir l'exemple positif lorsque l'exemple ancre n'a pas d'annotation. Cette stratégie nous permet de déterminer la performance de l'approche proposée par rapport à l'utilisation de transformations. Nous utilisons les données annotées pour tirer les exemples de manière supervisée et les données non annotées pour tirer les exemples de manière non supervisée.

Stratégie		Non supervisée	Semi-supervisée		Supervisée	
		S1 [Jansen et al., 2017]	S2	S3	S4	S5
Positif	Annotation		X	X	X	X
	Transformation	X	X			
	Plus proche			X		
Négatif	Annotation		X	X		X
	<i>Semi-hard mining</i>	X	X	X	X	

TABLEAU 5.1 – Stratégies de tirage des triplets.

- La stratégie S4 est une stratégie hybride qui utilise les annotations pour tirer l'exemple positif et le *semi-hard mining* pour tirer l'exemple négatif. Cette stratégie nous permet de déterminer l'impact du *semi-hard mining* dans l'apprentissage semi-supervisé. Les données non annotées ne sont pas utilisées.
- La stratégie S5 est une stratégie de tirage supervisée. Elle nous permet de déterminer l'impact des annotations dans l'apprentissage semi-supervisé. Les données non annotées ne sont pas utilisées.

5.3.3 Représentation de bas niveau et transformations

Les clips audio sont échantillonnés à 44,1 kHz. La représentation de bas niveau utilisée est un spectrogramme log-Mel. La **STFT** de chaque clip est calculée sur des trames de 25 ms avec un pas de 10 ms avant d'appliquer un banc de 64 filtres en échelle Mel et de calculer le logarithme des valeurs ainsi obtenues. Lors de l'utilisation de transformations, la transformation est faite sur le spectrogramme en échelle Mel où le logarithme n'est pas calculé et le logarithme est calculé après l'application des transformations.

Les paramètres des transformations utilisées sont similaires à ceux de Jansen et al. [2017] :

- lors de l'application du bruit gaussien $\sigma = 0,5$;
- lors de la translation temporelle, $T_s = (T - 1) \times 0,096$ le nombre de fenêtres dans notre exemple ;
- lors de la translation fréquentielle $S = 10$;
- lors du mélange d'exemples $\alpha = 0,25$;
- lors de l'utilisation d'un exemple du même clip $\Delta t = 10$ s, ce qui correspond à l'ensemble du clip.

5.3.4 Modèle

5.3.4.1 Modèles de référence

Le système de référence utilisé comme classifieur sur les données annotées utilise la même architecture que le système de référence de la Tâche 4 du Challenge DCASE 2018 décrit dans la partie 4.1.1. Cependant, seule la première passe est utilisée ici. Le modèle est un **CRNN** dont la sortie du **CNN** peut être utilisée comme représentation de haut niveau et le **RNN** composé d'une couche bidirectionnelle de 64 **GRU** est suivi d'une couche linéaire totalement connectée de 10 neurones et d'une agrégation moyenne comme classifieur.

5.3.4.2 Triplets

Pour l'apprentissage de représentation de haut niveau par triplets, un **CNN** similaire au système de référence est utilisé et optimisé avec le coût de triplet. La marge δ est égale à 1 pour l'ensemble des modèles. Une fois la représentation de haut niveau apprise, un classifieur est appris pour évaluer la performance d'étiquetage d'événements sonores. Nous utilisons un **RNN** à deux couches¹ bidirectionnelles de 64 **GRU** suivi d'une couche linéaire totalement connectée de 10 neurones et d'une agrégation moyenne comme classifieur. Le classifieur est appris sur les 1578 données faiblement annotées du jeu de données.

5.3.4.3 RNN appliqué au spectrogramme

Afin de comparer les approches utilisant des représentations de haut niveau à une approche d'étiquetage qui n'utilise pas une telle représentation, nous proposons d'appliquer le classifieur **RNN** à deux couches bidirectionnelles de 64 **GRU** suivi d'une couche linéaire totalement connectée de 10 neurones et d'une agrégation moyenne directement à la représentation de bas niveau. Ce modèle est utilisé par la suite de la même façon dont nous l'appliquons pour les triplets, et nous l'appelons « **RNN** spectrogramme ».

5.3.4.4 VGGish

Nous comparons aussi la représentation de haut niveau apprise spécifiquement sur notre jeu de données de façon semi-supervisée à des représentations de génériques apprises sur Audioset. Ces représentations sont obtenues par l'apprentissage d'un classifieur appelé VGGish sur le corpus Youtube-8M² [Hershey et al., 2017]. Ce corpus contient des millions de clips audio extraits de Youtube et 1000 classes d'événements. Le modèle VGGish est composé d'un **CNN** à 6 couches convolutionnelles et 3 couches d'agrégation suivies de 2 couches totalement connectées. Il est appris de façon supervisée sur des segments de 0,96 s, en dupliquant l'annotation au niveau du clip sur chaque segment. Après apprentissage, une représentation de haut niveau de taille 128 est classiquement extraite en

1. Un **RNN** à une couche étant trop peu profond pour réaliser une classification à partir des représentations de haut niveau, nous avons décidé d'utiliser deux couches, ce qui reste un réseau avec peu de paramètres. Nous avons aussi essayé une architecture de **RNN** à deux couches pour le système de référence mais la performance n'était pas aussi bonne qu'avec une seule couche.

2. <https://github.com/tensorflow/models/tree/master/research/audioset/vggish>

supprimant les couches totalement connectées. Nous ne nous limitons pas à ce choix et analysons les représentations extraites par différentes couches du modèle. Pour évaluer ces représentations, nous utilisons un classifieur RNN à deux couches bidirectionnelles de 64 GRU suivi d'une couche linéaire totalement connectée de 10 neurones et d'une agrégation moyenne, appris à l'aide des données annotées de notre jeu de données.

5.3.4.5 Professeur moyen

Le modèle de professeur moyen permet de comparer l'apprentissage semi-supervisé d'une représentation de haut niveau à l'apprentissage semi-supervisé d'un classifieur opérant directement sur la représentation de bas niveau. Nous utilisons la méthode de professeur moyen décrite dans le chapitre 4 de manière semi-supervisée. Le modèle utilisé suit l'architecture de CRNN décrite dans la partie 5.3.4.1 et les spectrogrammes d'entrée sont tels que décrits dans la partie 5.3.3. Les autres paramètres d'apprentissage sont identiques à ceux décrits dans la partie 4.1.3 pour le système de référence de la Tâche 4 du Challenge DCASE 2020. En particulier, lors de l'apprentissage, nous accroissons progressivement le taux d'apprentissage et le poids de la cohérence.

5.3.5 Métrique et validation

La performance est évaluée en utilisant la F-mesure à l'échelle du clip pour chaque classe et macro-moyennée. La validation des modèles d'apprentissage de représentation de haut niveau s'effectue en apprenant le classifieur sur les données faiblement annotées toutes les 2 époques et en calculant la performance sur le jeu de données de validation. Le modèle atteignant la meilleure performance sur le jeu de données de validation est sélectionné.

5.4 Analyse de performance de l'apprentissage par triplets

La première expérience, présentée dans le Tableau 5.2, compare les performances des cinq stratégies d'apprentissage par triplets en fonction du nombre de triplets non annotés. En effet, nous avons plus de 9 fois plus de données non annotées que de données annotées (14 412 clips contre 1 578), et nous voudrions connaître l'impact de l'utilisation des clips annotés dans l'apprentissage.

Sur la première ligne du tableau, nous utilisons 31 560 triplets par époque. Rappelons qu'un clip contient 10 exemples, donc le nombre d'exemples uniques dans le jeu de données faiblement annotées est 15 780. Lorsque nous utilisons 31 560 triplets par époque, pour S2 et S3 cela revient à utiliser le même nombre triplets annotés et non annotés. Dans le jeu de données non annotées, nous utilisons 1578 clips pour créer 15 780 triplets afin d'avoir un apprentissage similaire à l'apprentissage supervisé. Pour S1, cela correspond à utiliser les mêmes données que S2 et S3, sans utiliser les annotations pour les données faiblement annotées. Pour S4 et S5, nous utilisons les exemples annotés deux fois comme exemple ancre par époque mais avec des exemples positifs et négatifs différents. Dans les deuxième et troisième lignes, nous utilisons 85 780 et 159 900 triplets respectivement.

Pour S1, cela revient à utiliser un plus grand nombre de clips non annotés, jusqu'à utiliser l'ensemble des clips dans le troisième cas. Pour S2 et S3, cela signifie utiliser 15 780 triplets issus des 1578 clips annotés et le reste des triplets est issu de clips non annotés, ce qui crée un déséquilibre entre les données annotées et non annotées. Pour S4 et S5, cela revient à utiliser chaque exemple comme exemple ancre quatre fois et demi et neuf fois dans le deuxième et le troisième cas, respectivement, mais avec des exemples positifs et négatifs différents.

Nombre de triplets	Stratégie				
	S1	S2	S3	S4	S5
31 560	52,32	54,35	50,96	42,47	53,59
85 780	51,44	54,57	46,13	42,72	51,85
159 900	51,62	52,27	38,59	42,19	49,95

TABLEAU 5.2 – F-mesure (%) obtenue par les cinq stratégies d'apprentissage par triplets en fonction du nombre de triplets.

Dans le Tableau 5.2, nous constatons que la stratégie utilisant une transformation de l'exemple ancre comme exemple positif (S2) fonctionne mieux que d'utiliser l'exemple le plus proche dans l'espace d'entrée (S3). Ceci peut s'expliquer par le manque de variabilité lors de l'apprentissage, qui pourrait être résolu en utilisant l'un des exemples les plus proches dans l'espace d'entrée. La comparaison de S4 et S5 indique que le *semi-hard mining* n'est pas souhaitable lorsque nous tirons un exemple positif grâce aux annotations. Nous pouvons aussi constater les bénéfices de l'apprentissage semi-supervisé. En effet, la stratégie S2 est la plus performante et elle dépasse la stratégie non supervisée S1. De plus, la stratégie S2 dépasse la stratégie S5, en particulier lorsque le nombre de triplets augmente, ce qui pourrait indiquer l'utilité de la transformation de l'ancre qui peut être considérée comme une forme d'augmentation de données. Ceci permet l'augmentation du nombre d'exemples dans le jeu de données et, dans ce cas, il est certain que l'exemple positif contient la même classe que l'exemple ancre (contrairement à S3).

Une autre explication serait que la stratégie de tirage utilisant une transformation pour définir l'exemple positif opère comme une régularisation qui permet au modèle de généraliser, ce qui peut expliquer la performance élevée de l'approche semi-supervisée utilisant cette stratégie (S2). Dans le cas du tirage du plus proche voisin dans l'espace d'entrée pour définir l'exemple positif (S3), l'augmentation du nombre de triplets implique que ces exemples sont plus proches les uns des autres, ce qui peut expliquer la dégradation de performance lorsque nous augmentons le nombre d'exemples utilisés. En effet, lorsque le nombre d'exemples est important, l'exemple le plus proche de l'ancre peut être plus proche dans l'espace d'entrée que ne le serait un exemple transformé. Pour traiter ce problème, une alternative serait de tirer l'exemple le plus proche sous la contrainte qu'il soit plus éloigné que les exemples transformés. Ceci sort du cadre de cette étude.

Les deuxième et troisième cas utilisant 85 780 et 159 900 triplets par époque nous per-

mettent de comparer les approches utilisant les données non annotées. En effet, on peut voir que S2 bénéficie des données non annotées jusqu'à un certain point mais que lorsque les données annotées sont très importantes (159 900 triplets), les performances se dégradent. Cependant, c'est le cas pour les autres approches aussi. Une explication peut être que, chaque époque optimisant un nombre plus important de lots, la sélection du meilleur modèle lors de la validation qui est réalisée toutes les 2 époques sera différente. Lors de l'utilisation des approches supervisées S4 et S5, l'utilisation d'un plus grand nombre de triplets par époque indique une différence dans le calcul de la validation (qui est réalisée toute les deux époques). Nous aurions donc pu augmenter le nombre d'époques plutôt que le nombre de triplets par époque, ce qui pourrait également éviter les problèmes impliqués par la validation mentionnés précédemment.

Dans la Figure 5.4, nous visualisons les représentations de haut niveau apprises avec les différentes stratégies de tirage des exemples. Pour simplifier, nous avons choisi de représenter seulement les 3 classes les plus distinctes pour chaque stratégie. Cette figure utilise les représentations de haut niveau obtenues avec 31 560 triplets. Ces représentations semblent cohérentes avec les résultats présentés dans le Tableau 5.2. En effet, nous pouvons voir une distinction des trois classes représentées avec l'ensemble des stratégies, excepté S4. Cependant, d'après cette figure, nous pouvons constater que les représentations ont une marge d'amélioration assez importante.

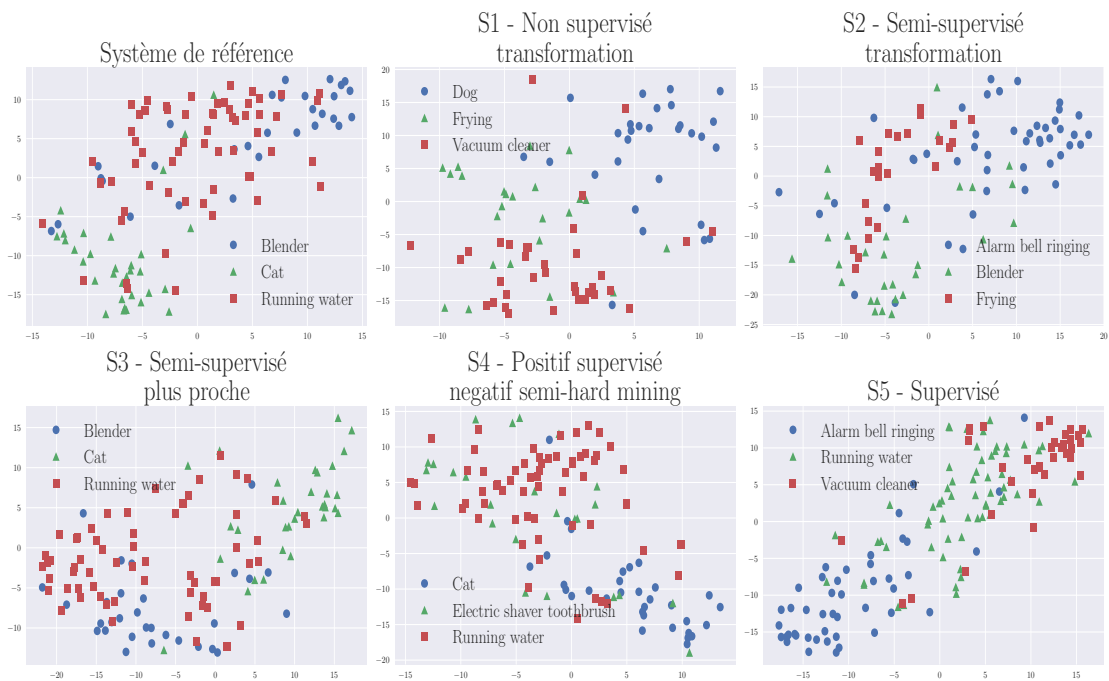


FIGURE 5.4 – Visualisation par t-SNE des représentations de haut niveau de 3 classes obtenues par le système de référence et par les cinq stratégies d'apprentissage par triplets.

Le Tableau 5.3 quantifie l'impact du ratio entre les données annotées et non annotées.

	NA	7,890	15,780	19,725	23,670
	A	23,670	15,780	11,835	7,890
S2		55,16±0,7	54,35±0,7	47,3±6,2	17,42±26,8
S3		51,58±2,2	50,96±2,0	30,02±19,8	0,0±0,0

TABLEAU 5.3 – F-mesure (%) obtenue par les stratégies d’apprentissage par triplets S2 et S3 en fonction du nombre de triplets issus de données annotées (A) et non annotées (NA). Les expériences sont réalisées 3 fois pour un intervalle de confiance de 95%.

Nous utilisons 31 560 triplets avec une proportion différente de triplets issus de données annotées et de données non annotées. Seulement 1578 clips non annotés ont été tirés aléatoirement pour avoir 15 780 exemples ancres possibles par jeu de données. Lorsque le nombre de triplets issus de données annotées ou non annotées est supérieur à 15 780, nous utilisons plusieurs fois certains exemples comme exemple ancre au sein d’une même époque. Nous pouvons voir l’importance d’utiliser les données annotées dans cette expérience. S3 est plus sensible au nombre de données annotées que S2. Nous constatons aussi que, lorsque nous n’avons pas assez de données annotées, la performance se dégrade de façon importante. Une explication possible pour S2 est que, le nombre de triplets annotés étant très faible, le modèle sur-apprend ces exemples qui peuvent avoir une erreur très importante par rapport aux exemples utilisant les données non annotées. L’intervalle de confiance indique que ce modèle est très instable à l’apprentissage. Pour S3, la chute de performance est probablement très importante à cause des nombreux triplets non annotés qui ne contribuent pas au coût d’apprentissage (lorsque l’exemple le plus proche dans l’espace d’entrée est aussi le plus proche dans l’espace de sortie) et le nombre de triplets annotés n’est pas suffisant à l’apprentissage du modèle. La différence de performance entre la dernière case du Tableau 5.3 et la première case du Tableau 5.2 pour S2 peut paraître surprenante. Cela amène à la conclusion qu’il est préférable de ne pas utiliser les annotations lorsque nous avons un nombre de données annotées trop faible pour S2. D’après les résultats par classe, le nombre de données annotées dont on a besoin par classe semble dépendre du nombre d’événements, de la durée des événements, et de la superposition de ceux-ci avec les autres classes. Lorsque nous utilisons un nombre plus important de triplets issus de données annotées (23 670), la performance est proche de la stratégie supervisée (S5) du Tableau 5.2. Dans ce cas les données non annotées peuvent servir de régularisation, ce qui peut expliquer le gain de performance dans ce cas.

5.5 Comparaison de l’apprentissage par triplets avec l’apprentissage par classifieur

Dans cette partie, nous comparons les représentations de haut niveau obtenues avec l’apprentissage par triplets aux représentations obtenues par l’apprentissage d’un classifieur. Nous nous concentrons sur les représentations obtenues avec VGGish appris sur Au-

Couche du modèle	Taille du vecteur de sortie
conv1	393 216
maxpool1	98 304
conv2	196 608
maxpool2	49 152
conv3_1	98 304
conv3_2	98 304
maxpool3	24 576
conv4_1	49 152
conv4_2	49 152
maxpool4	12 288
fc1_1	4 096
fc1_2	4 096
embedding	128

TABLEAU 5.4 – Taille des représentations extraites de chaque couche du modèle VGGish. Les couches sont dans l'ordre d'apparition : « conv » représente une couche de convolution, « maxpool » une couche d'agrégation maximum et « embedding » la représentation de haut niveau de taille 128 communément utilisée.

dioset puis nous comparons les différentes approches supervisées et enfin les différentes approches semi-supervisées.

5.5.1 Représentation de haut niveau apprise sur Audioset

Pour commencer, nous considérons les représentations de haut niveau extraites des différentes couches du modèle VGGish appris sur Audioset, dont la taille est indiquée dans le Tableau 5.4. La Figure 5.5 présente les résultats obtenus avec ces représentations pour l'étiquetage d'événements sonores, grâce à un classifieur appris comme expliqué dans la partie 5.3.4. Notons que, comme ces représentations n'ont pas la même taille, la complexité du classifieur utilisé varie également d'une couche à l'autre. Nous constatons tout d'abord une différence absolue de 5 à 18% entre les F-mesures obtenues sur le jeu de validation et sur le jeu d'évaluation. De manière générale les performances sont meilleures dans la deuxième moitié du réseau, ce qui confirme l'utilité d'une représentation de haut niveau. Nous constatons que la performance obtenue avec la représentation de haut niveau issue de la dernière couche n'est pas la meilleure. Les meilleures performances sur le jeu d'évaluation sont obtenues à la sortie des couches totalement connectées (« fc1_1 » et « fc1_2 »). Si nous nous concentrons sur les résultats obtenus sur le jeu de données de validation, nous constatons que les performances augmentent jusqu'à la sortie du

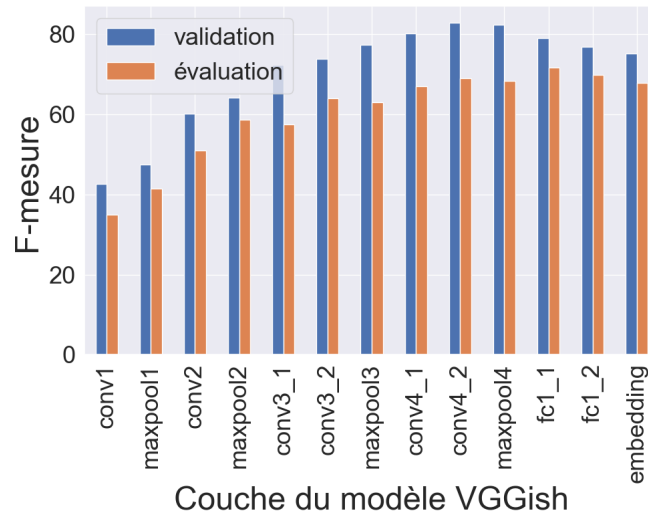


FIGURE 5.5 – F-mesure (%) obtenue à partir des représentations extraites de chaque couche du modèle VGGish.

CNN (« maxpool4 ») et diminuent ensuite. Si nous comparons les performances obtenues avec les représentations issues des dernières couches convolutionnelles (« conv4_1 », « conv4_2 », « maxpool4 ») et les performances obtenues avec les représentations issues des dernières couches totalement connectées (« fc1_1 », « fc1_2 », « embedding »), nous constatons une différence de performance entre le jeu de validation et le jeu d'évaluation. Ceci peut s'expliquer par la complexité du classifieur qui est plus importante lorsqu'il est utilisé avec les représentations issues des dernières couches convolutionnelles que lorsqu'il est utilisé avec les représentations issues des couches totalement connectées. Une autre explication serait que le **CNN** est utile pour extraire une représentation riche, et nous permet donc d'obtenir de bonnes performances. Les couches totalement connectées permettent ensuite de compresser l'information afin d'avoir une représentation plus facilement utilisable. Lorsque nous observons les performances sur le jeu de données d'évaluation, cette hypothèse semble se confirmer, car la performance augmente entre la couche « maxpool4 » et « fc1_2 » alors qu'elle diminue sur le jeu de validation, pouvant indiquer une meilleure généralisation du modèle qui pourrait être induite par la compression. Cependant, cette compression a ses limites, puisque les performances sur le jeu de données d'évaluation diminuent elles aussi après la couche « fc1_2 ». Une explication de cette baisse de performance pourrait être que, plus VGGish utilisent de couches totalement connectées et donc plus on se rapproche de la sortie, plus la représentation devient spécifique à l'application utilisée lors de l'apprentissage.

5.5.2 Comparaison des méthodes d'apprentissage supervisé

Nous comparons maintenant les différents modèles appris de façon supervisée à l'aide des données faiblement annotées uniquement, c'est-à-dire :

- le classifieur (RNN à 2 couches) appris sur les spectrogrammes ;
- le modèle de classification décrit dans la partie 5.3.4, qui est un CRNN appris à partir des spectrogrammes ;
- le modèle qui utilise les triplets issus des données annotées uniquement (S5) avec 31 560 triplets ;
- le modèle de classification décrit dans la partie 5.3.4 mais appris en utilisant la méthode du professeur moyen décrite au chapitre 4. Ce modèle est appris uniquement avec les données faiblement annotées.

Les performances de ces modèles sont présentées dans le Tableau 5.5. Les trois premières lignes du tableau confirment l'intérêt d'extraire une représentation de haut niveau apprise sur notre jeu de données. En effet, l'apprentissage de bout-en-bout permet une amélioration de 14% entre le CNN et le CRNN. L'apprentissage par triplets (S5) améliore encore la performance par rapport à l'apprentissage de bout-en-bout utilisant le CRNN. Enfin, l'apprentissage par la méthode de professeur moyen fournit les meilleurs résultats. Sa supériorité pourrait s'expliquer par l'utilisation d'un coût de cohérence au niveau des sorties fortes du modèle qui peut compenser le problème des annotations faibles. La présence des événements dans une partie du clip peut en effet induire un biais sur l'apprentissage par triplets, qui sera discuté dans la partie 6.6.1.

Méthode	F-mesure
RNN spectrogramme	35,3
CRNN	49,6
Triplets (S5)	53,6
Professeur moyen (faibles)	62,0

TABLEAU 5.5 – F-mesure (%) obtenue à partir des différentes méthodes d'apprentissage supervisé.

5.5.3 Comparaison des méthodes d'apprentissage semi-supervisé

Pour terminer, nous comparons les différents modèles appris de façon semi-supervisée à l'aide des données faiblement annotées et des données non annotées, c'est-à-dire :

- le modèle qui définit l'exemple positif par une transformation lors de l'utilisation des données non annotées (S2), basé sur 31 560 triplets issus de façon équilibrée des données annotée et non annotées ;
- le classifieur appris sur la représentation issue de la dernière couche de VGGish (« embedding ») ;
- le modèle « professeur moyen » de la section précédente (5.5.2) appris de manière semi-supervisée.

Méthode	F-mesure
Triplets (S2)	54,4
VGGish Audioset	67,8
Professeur moyen (NA + A)	63,3

TABLEAU 5.6 – F-mesure (%) obtenue à partir des différentes méthodes d’apprentissage semi-supervisé.

Les performances de ces modèles sont présentées dans le Tableau 5.6. Nous constatons que l’utilisation d’une représentation issue de l’apprentissage sur un grand jeu de données contenant des classes très variées pouvant être éloignées de notre domaine d’intérêt est préférable à l’apprentissage de modèle sur le jeu de données spécifique à notre application. Cela signifie que la représentation générale qui a été apprise sur le grand jeu de données contient des informations intéressantes pour discriminer les sons. Cependant, il faut noter que le [CRNN](#) appris avec la méthode du professeur moyen a 133 706 paramètres alors que VGGish a 64 millions de paramètres jusqu’à la représentation de taille 128 issue de la dernière couche considérée ici. Au delà de la taille du jeu de données, la complexité du modèle pourrait avoir un impact. Les performances présentées confirment également que l’apprentissage par professeur moyen est plus performant que l’apprentissage par triplets. Cela est probablement dû au biais introduit par les annotations faibles lors de l’apprentissage par triplets, mentionné dans la partie 5.5.2.

Dans ces expériences, il aurait été intéressant de comparer ces approches avec la représentation issue du modèle de [Jansen et al. \[2017\]](#) apprise sur le corpus Audioset. Cependant, le code n’est pas disponible et l’apprentissage de ce modèle n’est pas réaliste en raison de la complexité du modèle et du temps d’apprentissage nécessaire avec notre infrastructure.

5.6 Conclusion

Dans ce chapitre, nous avons présenté une analyse concernant l’apprentissage de représentation de manière semi-supervisée. Nous avons présenté une méthode d’apprentissage de représentation semi-supervisée par triplets. Cette méthode a été comparée avec différentes méthodes d’apprentissage de représentation par triplets qui étaient non supervisées et supervisées. Nous avons proposé une nouvelle stratégie de tirage de l’exemple positif lors de la création de triplets de manière non supervisée, qui consiste à tirer l’exemple le plus proche dans l’espace d’entrée. Cette stratégie n’est pas aussi performante qu’une stratégie reposant sur l’utilisation de transformations et que nous avons adaptée pour l’apprentissage semi-supervisé. Cependant, ces stratégies pourraient être utilisées de manière complémentaire. Nous avons ensuite comparé l’apprentissage par triplets avec différents classifieurs utilisant l’apprentissage supervisé ou semi-supervisé. Nous avons pu montrer l’intérêt d’apprendre une représentation, quelle que soit la méthode. Nous avons montré que la représentation issue d’un classifieur appris sur un jeu de données très

important permet une bonne classification de nos événements d'intérêt, mais la complexité du modèle utilisé peut être limitante dans certaines applications. Finalement, nous émettons l'hypothèse que l'apprentissage par triplets n'est pas très efficace à cause de l'utilisation des annotations faibles qui rend les méthodes d'apprentissage basées sur des distances difficiles en raison de l'incertitude sur la présence de l'événement dans le segment sélectionné. Les résultats par classe, non présentés dans ce chapitre, semblent confirmer cette hypothèse. C'est pourquoi nous proposons de traiter le problème des annotations faibles plus en détail dans le chapitre 6.

6 Impact des annotations faibles

Ce chapitre est consacré à l'analyse de l'impact des annotations faibles sur l'apprentissage d'un système d'analyse de sons ambiants.

Pour l'apprentissage d'un système de détection d'événements sonores, l'idéal serait d'utiliser une approche complètement supervisée. Cependant, l'annotation forte des données est coûteuse et difficile. En pratique, des jeux de données faiblement annotés qui peuvent être obtenus de façon moins coûteuse par *crowdsourcing* [Fonseca et al., 2017] sont donc régulièrement utilisés. Cela implique d'estimer la segmentation des événements lors de la phase d'évaluation alors qu'on ne dispose d'aucune information de segmentation lors de la phase d'apprentissage. Plusieurs méthodes faiblement supervisées ont été proposées pour la détection d'événements sonores et nous avons passé en revue les performances de différents systèmes de détection d'événements sonores appris de façon non supervisée et faiblement supervisée avec différentes tolérances de segmentation [Serizel and Turpault, 2019]. Une partie de cette analyse a été présentée dans le chapitre 4.

Lorsque l'on cherche à faire de l'étiquetage d'événements sonores, il n'est pas nécessaire de segmenter les événements lors de la phase d'évaluation. Cette tâche est donc plus simple que la détection d'événements sonores et l'utilisation de données faiblement annotées paraît plus naturelle à priori. Afin de résoudre cette tâche, le modèle doit tout de même être capable de discriminer différentes classes d'événements sans savoir à quel endroit elles apparaissent dans les clips audio d'apprentissage.

De nombreux travaux essayent d'exploiter les annotations faibles pour l'étiquetage ou la détection d'événements sonores, dont certains sont basés sur l'apprentissage multi-instance [Kumar and Raj, 2016], l'utilisation de filtres gaussiens [Su et al., 2017], une méthode d'attention [Kim and Ghaffarzadegan, 2019] ou bien une méthode basée principalement sur un post-traitement [Dinkel et al., 2021]. Tandis que ces méthodes semblent efficaces pour améliorer les performances d'étiquetage ou de détection d'événements sonores, elles tentent de résoudre le problème sans isoler le problème des annotations faibles ni faire une analyse de l'impact des annotations faibles dans le résultat final. C'est pourquoi il est difficile d'établir si les améliorations de performance observées sont liées à une meilleure exploitation des données faiblement annotées ou à la résolution d'autres problèmes présents (polyphonie, événements non-cibles, données non équilibrées, etc.). Shah et al. [2018] ont essayé d'analyser le problème des annotations faibles. Cependant, ils se sont basés sur des données issues d'AudioSet qui présentent d'autres problèmes que celui des annotations faibles pour lesquelles des solutions indépendantes ont pu être proposées : annotations non-fiables ou manquantes [Fonseca et al., 2019a], polyphonie [Bisot et al., 2017a], rapport de puissance variable entre événements sonores et bruit de fond [Benetos et al., 2016], présence d'événements sonores non-cibles ou encore jeu de données non équilibré [Xu et al., 2017]. De plus, Shah et al. [2018] ont utilisé des

clips audio de 10 s avec des annotations faibles puis ils ont étendu ces clips à 30 s et à 60 s en utilisant une partie plus longue du clip audio original issue de Youtube. Leur analyse ne prend pas en compte la durée de l'événement cible dans le clip original qui affecte fortement l'impact des annotations faibles [Turpault et al., 2020a], ni le fait que les parties du clip audio ajoutées peuvent elles aussi contenir l'événement cible.

Dans ce chapitre, nous proposons une analyse détaillée de l'impact des annotations faibles pour la tâche d'étiquetage d'événements sonores. Nous définissons d'abord le problème et proposons une solution pour le traiter indépendamment des autres problèmes habituellement présents. Nous montrons l'impact des annotations faibles sur différents systèmes permettant de calculer des représentations de haut niveau. Nous montrons aussi que pour limiter l'impact des annotations faibles sur l'apprentissage, les clips d'apprentissage doivent être au moins aussi longs que les clips d'évaluation et que des clips d'apprentissage plus longs ont un impact faible. Finalement, nous montrons qu'une bonne agrégation temporelle aide à réduire l'impact des annotations faibles au moment de l'évaluation et nous donnons des indications de la granularité d'annotation nécessaire dans un corpus en fonction du scénario visé.

6.1 Définition du problème

Nous faisons l'hypothèse que chaque clip \mathbf{X} contient un ou plusieurs événements sonores de \mathcal{C} accompagnés d'un bruit de fond stationnaire. L'annotation forte du clip est représentée sous forme de matrice \mathbf{Y} de taille $C \times T$ où C est le nombre de classes de \mathcal{C} . Chaque $y_{c,t} \in \{0, 1\}$ représente l'absence ou la présence d'une classe c dans la fenêtre temporelle t . Le vecteur $\mathbf{y}_t = [y_{1,t}, \dots, y_{C,t}]^T$ de taille $C \times 1$ représente l'activité des classes d'intérêt dans cette fenêtre. De manière similaire, l'annotation faible de ce clip peut être représentée sous la forme d'un vecteur $\mathbf{w} = [w_1, \dots, w_C]^T$ de taille $C \times 1$ avec w_c défini par

$$w_c = \max_t(y_{c,t}). \quad (6.1)$$

Cette équation exprime le fait que, s'il n'y a pas d'erreur d'annotation, une classe c est considérée comme présente dans le clip ($w_c = 1$) si elle est présente dans au moins une fenêtre temporelle. Ceci est représenté dans la Figure 6.1.

Étant intéressés par les annotations faibles, nous définissons la densité d'événement ainsi :

$$D_c = \frac{1}{T} \sum_{t=1}^T y_{c,t}. \quad (6.2)$$

Plus un événement est présent longtemps dans le clip, plus D_c est élevé. Lorsque la densité d'une classe est égale à 0 ou 1, les annotations fortes et faibles sont équivalentes. Dans les autres cas, l'utilisation des annotations faibles correspond à introduire du bruit d'annotation qui est quantifié par $w_c - D_c$.

Nous sommes intéressés ici par la tâche d'étiquetage d'événements sonores où l'on cherche quels événements sonores apparaissent dans un clip audio sans se soucier du nombre de fois et à quel moment ils apparaissent. Nous cherchons donc à estimer un vecteur $\hat{\mathbf{w}}$

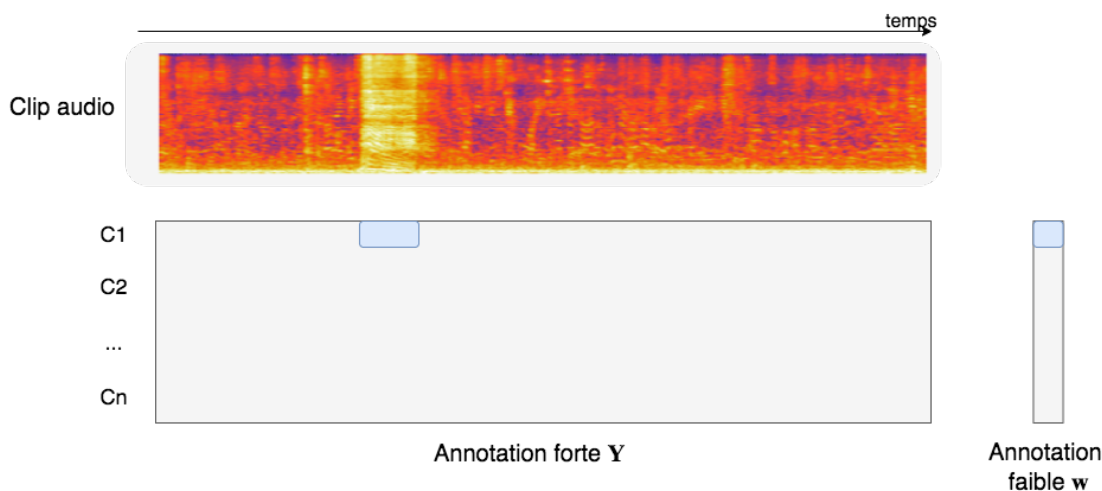


FIGURE 6.1 – Illustration de la différence entre annotation forte et faible.

représentant les probabilités de présence des classes d'événements sonores dans un clip à partir de sa représentation temps-fréquence \mathbf{X} . Ces scores au niveau du clip sont généralement obtenus en estimant les probabilités au niveau des fenêtres $\hat{\mathbf{y}}_t$ ¹ et en les agrégeant selon l'axe temporel en utilisant une fonction d'agrégation fixe ou paramétrique. En fonction des données d'apprentissage utilisées (faiblement ou fortement annotées), la phase d'agrégation peut être appliquée durant l'apprentissage ou seulement à l'évaluation. De plus, dans le cas d'une fonction d'agrégation paramétrique, les paramètres peuvent être appris ou fixes.

Pour étudier l'impact des annotations faibles sur l'étiquetage d'événements sonores, nous proposons de varier la densité d'événement et d'analyser l'impact du bruit introduit par les annotations faibles lors de l'apprentissage et de l'évaluation. Nous notons par \mathcal{I} l'ensemble d'apprentissage et nous faisons l'hypothèse qu'il est constitué de données (\mathbf{X}, \mathbf{w}) faiblement annotées ou (\mathbf{X}, \mathbf{Y}) fortement annotées sans erreur d'annotation.

6.2 Isolation du problème

Afin d'analyser le problème des annotations faibles, il nous a fallu définir un scénario et créer un corpus permettant d'isoler le problème des annotations faibles. Nous définissons trois nouveaux jeux de données appelés « analyse des annotations faibles » (AAF), « événements tronqués, analyse des annotations faibles tronquées » (ETA AF) et « densité fixe, analyse des annotations faibles » (DAAF). Ces jeux de données sont des jeux de données synthétiques créés à partir du corpus [DESED](#). Les clips audio de ces jeux de données durent 10 s et sont générés en mélangeant un événement sonore d'une classe

1. Pour simplifier les notations, nous faisons l'hypothèse que les entrées et les sorties du modèle ont la même résolution temporelle définie par $t \in \{1, \dots, T\}$. En réalité, la résolution temporelle des entrées et des sorties peut être différente.

Classe	Développement		Évaluation
	Apprentissage	Validation	
Alarme/sonnette/sonnerie	177	13	63
Blender	89	9	27
Chat	78	10	26
Plats de cuisine	99	10	34
Chien	121	15	43
Rasoir / brosse à dent électrique	51	5	17
Friture	56	8	17
Eau qui coule	59	9	20
Voix	117	11	47
Aspirateur	62	10	20
Total	909	100	314

TABLEAU 6.1 – Nombre d'événements Freesound isolés utilisés dans chaque ensemble de données.

d'intérêt et du bruit de fond avec un rapport de puissance compris entre 6 et 30 dB. Ces jeux de données reposent sur les données d'apprentissage qui ont été séparées en 90% d'apprentissage et 10% de validation et les données d'évaluation de la banque de données DESED. Nous rappelons les événements sonores des classes d'intérêt utilisées dans le Tableau 6.1.

6.2.1 AAF

Le corpus de données AAF contient 2700 clips audio d'apprentissage, 300 de validation et 750 d'évaluation. Chacun de ces ensembles est équilibré pour que la distribution des classes ne soit pas un problème. La durée effective de chaque événement sonore dépend de la durée de l'événement isolé utilisé et de son instant de début dans le clip qui est choisi de façon aléatoire (les événements sonores qui sont plus longs que la durée restante du clip sont tronqués). La Figure 6.2 présente la distribution de la durée des événements sonores pour chacune des classes dans le jeu d'apprentissage. On distingue les deux catégories de classes introduites dans le chapitre 3 : « alarme/sonnette/sonnerie », « chat », « plats de cuisine », « chien » et « voix » représentant les classes d'événements courts et « blender », « rasoir / brosse à dent électrique », « friture », « eau qui coule » et « aspirateur » correspondant aux classes d'événements longs.

Afin de contrôler à quel point les annotations sont faibles dans les jeux de données créés, on coupe les clips audio de 10 s en clips plus courts. Les clips sont coupés à 5 différentes durées $d_{\text{clip}} = 0, 2, 0, 5, 1, 3, \text{ or } 5$ s. Lorsque nous coupons les clips, nous nous assurons de garder l'événement d'intérêt. Le segment choisi débute au début de l'événement si l'événement est plus long que d_{clip} , sinon le segment choisi est pris aléatoirement autour de l'événement pour faire apparaître l'événement à une position aléatoire dans le nouveau clip. Nous illustrons le second scénario dans la Figure 6.3 avec une durée de clip de 5 s et un événement de durée inférieure à 5 s. Lorsque la durée du clip est courte, la plupart

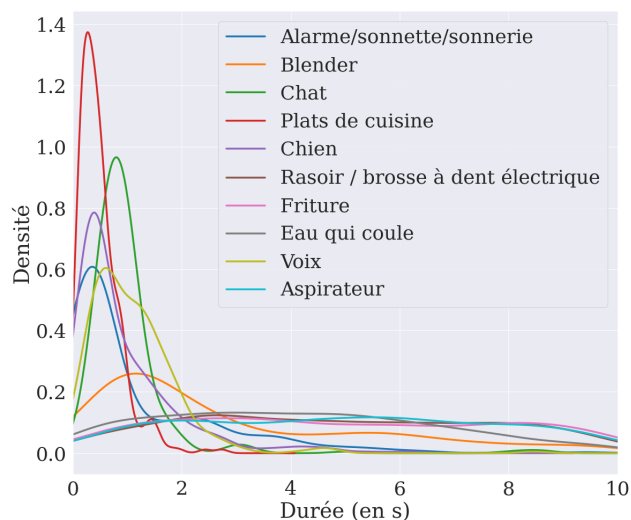


FIGURE 6.2 – Distribution (estimée à l'aide d'un estimateur par noyau) de la durée des événements sonores par classe dans l'ensemble d'apprentissage.

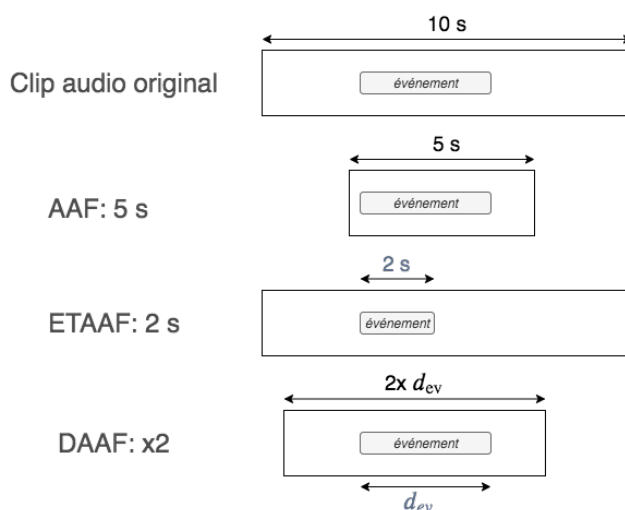


FIGURE 6.3 – Représentation des différents jeux de données synthétiques créés pour analyser les annotations faibles. d_{ev} représente la durée de l'événement.

des fenêtres temporelles contiennent l'événement sonore et l'annotation peut donc être considérée comme forte. Quand la durée du clip est plus longue, le nombre de fenêtres temporelles où l'événement sonore est absent augmente, et la densité d'événement devient donc plus faible.

À titre d'exemple pour comprendre les jeux de données, 122 événements sonores durent moins de 0,2 s et parmi ces événements on compte 60 « plats de cuisine », 18 « voix », 18 « chien », 13 « alarme/sonnette/sonnerie », 10 « eau qui coule » et 3 « chat ». Cela

signifie que, même dans les clips audio coupés à 200 ms, environ $\frac{1}{4}$ des bruits de plats peuvent être considérés comme annotés de manière faible. Cependant, le bruit introduit dans ce cas reste faible (97 événements font plus de 0,1 s et aucun ne fait moins de 50 ms). Dans la suite de ce manuscrit, nous faisons donc l’hypothèse que, dans le cas particulier des clips de durée 200 ms, $D_c = 1$ car lors de l’annotation des événements dans les clips audio réels de DESED, la durée minimale des événements annotés était de 250 ms.

6.2.2 ETAAF

Afin d’analyser plus en détail les annotations faibles et permettre un contrôle plus important sur la densité d’événement dans les clips audio, nous créons le jeu de données ETAAF. Le jeu de données ETAAF utilise les mêmes données initiales que le jeu de données AAF. Cependant, les événements sont tronqués dans les clips audio utilisés. Les événements sont tronqués à 5 durées différentes : $d_{\text{tronq}} = 0, 2, 0, 5, 1, 3, \text{ or } 5$ s. Si un événement sonore est plus long que la durée de troncature choisie, l’événement est tronqué, sinon on garde l’événement original. Ce scénario est illustré dans la Figure 6.3 avec une durée d’événement tronqué de 2 s et un événement initial plus long que cette durée.

6.2.3 DAAF

Les jeux de données AAF et ETAAF permettent de faire apparaître des scénarios avec un contrôle différent sur la « faiblesse » des annotations. Cependant, aucun d’eux ne permet de fixer la densité des événements pour l’ensemble des événements. Pour résoudre ce problème, nous créons le jeu de données DAAF. Le jeu de données DAAF reprend les mêmes associations de bruits de fond et d’événements isolés que les autres jeux de données, mais cette fois-ci ajoute du bruit autour de l’événement en fonction de la durée de l’événement original d_{ev} . Dans ce jeu de données, la durée des clips varie mais la densité d’événement D_c est fixe entre tous les clips. Trois durées de clips sont utilisées à partir de l’événement présent dans le clip : $d_{\text{clip}} = d_{\text{ev}}, d_{\text{ev}} \times 2, d_{\text{ev}} \times 10$. Lorsque $d_{\text{clip}} = d_{\text{ev}}$ les annotations sont fortes car toutes les fenêtres temporelles contiennent l’événement à retrouver. Lorsque $d_{\text{clip}} = d_{\text{ev}} \times 2$ ou $d_{\text{ev}} \times 10$, la densité d’événement est respectivement de $D_c = 0,5$ et $D_c = 0,1$. Si la durée du clip est supérieure à la durée du bruit de fond, le bruit de fond est dupliqué afin d’atteindre la durée de clip correspondante. Le scénario $d_{\text{clip}} = d_{\text{ev}} \times 2$ est représenté dans la Figure 6.3.

6.3 Impact attendu des annotations faibles sur l’apprentissage de représentation

Dans cette partie, nous analysons l’impact des annotations faibles sur l’apprentissage d’une représentation de haut niveau. Notre but est d’apprendre un modèle E calculant une représentation de haut niveau qui sera utilisée comme entrée par un classifieur G pour l’étiquetage d’événements sonores. Nous considérons trois méthodes d’apprentissage

différentes : apprentissage de bout-en-bout, par triplets ou par réseau prototype. Nous avons identifié la difficulté d'apprendre une telle représentation dans le chapitre 5. Ici nous analysons si cette difficulté est principalement causée par les annotations faibles.

6.3.1 Apprentissage de bout-en-bout

L'apprentissage de bout-en-bout consiste à apprendre conjointement les modèles E et G en minimisant un coût de classification qui est par exemple la somme des entropies croisées binaires sur les C classes

$$\sum_{\mathbf{X} \in \mathcal{J}} \sum_c -w_c \log(G(E(\mathbf{X}))_c) - (1 - w_c) \log(1 - G(E(\mathbf{X}))_c) \quad (6.3)$$

où $G(\cdot)_c$ représente la sortie du classifieur pour la classe c . Dans ce cas, les annotations faibles ont un impact à la fois sur l'apprentissage de E et de G . L'impact est principalement dû à la capacité de G à distinguer les événements sonores du bruit de fond. Cependant, même si G est capable de détecter les fenêtres contenant le bruit, l'apprentissage de la représentation n'étant pas explicite, il n'est pas garanti que la représentation extraite par E sépare bien les événements du bruit.

6.3.2 Apprentissage par triplets

Nous rappelons la méthode d'apprentissage par triplets décrite en 5.1.1, utilisée ici de manière supervisée. Nous rappelons que l'apprentissage par triplets utilise des triplets $(\mathbf{X}^a, \mathbf{X}^p, \mathbf{X}^n)$ où l'ancre \mathbf{X}^a est un exemple du jeu d'apprentissage, l'exemple positif \mathbf{X}^p est un exemple aléatoire contenant la même annotation que l'ancre et l'exemple négatif \mathbf{X}^n est un exemple aléatoire avec une annotation différente de celle de l'ancre. Le réseau E est appris en minimisant le coût de triplet [Wang et al., 2014]

$$\sum_{\mathbf{X}^a \in \mathcal{J}} [\|E(\mathbf{X}^a) - E(\mathbf{X}^p)\|_F^2 - \|E(\mathbf{X}^a) - E(\mathbf{X}^n)\|_F^2 + \delta]_+, \quad (6.4)$$

où $[\cdot]_+$ est le coût charnière, $\|\cdot\|_F$ est la norme de Frobenius et δ est un paramètre de marge. Le coût de triplet a pour but de trouver une représentation de haut niveau discriminante dans laquelle l'ancre et l'exemple positif sont plus proches que l'ancre et l'exemple négatif. Dans ce cas, les annotations faibles sont problématiques puisque la distance entre les représentations de haut niveau de deux clips ayant une densité D_c faible dépend majoritairement de la distance entre leurs bruits de fond.

6.3.3 Réseau prototype

Dans le réseau prototype, les données sont là aussi tirées d'une façon spécifique [Snell et al., 2017]. Chaque lot de données d'apprentissage contient un certain nombre de classes et un certain nombre m_c d'exemples \mathbf{X} par classe. Parmi ces exemples, m_c^s sont appelés exemples « supports » et sont utilisés pour calculer un *prototype* de chaque classe. Le

vecteur prototype de la classe c est la moyenne des représentations de haut niveau des exemples supports :

$$\mathbf{proto}_c = \frac{1}{m_c^s} \sum_{\mathbf{X} \in \mathcal{J}_c^s} E(\mathbf{X}) \quad (6.5)$$

où \mathcal{J}_c^s est l'ensemble des exemples supports de la classe c dans le lot considéré. Les $m_c^q = m_c - m_c^s$ exemples restants sont appelés exemples « requêtes » et sont utilisés pour apprendre le modèle E . Pour chaque exemple requête, des pseudo-probabilités de classes sont définies par le softmax des distances entre la représentation de haut niveau de cet exemple et les prototypes de chacune des classes :

$$\mathbf{pseudo}(\mathbf{X}) = \text{softmax}(\|E(\mathbf{X}) - \mathbf{proto}_1\|_F, \dots, \|E(\mathbf{X}) - \mathbf{proto}_C\|_F). \quad (6.6)$$

Le coût à minimiser est la somme des entropies croisées binaires pour chaque classe entre l'annotation de l'exemple requête et ces pseudo-probabilités :

$$\sum_{\mathbf{x} \in \mathcal{J}} \sum_c -w_c \log(\mathbf{pseudo}_c(\mathbf{X})) - (1 - w_c) \log(1 - \mathbf{pseudo}_c(\mathbf{X})). \quad (6.7)$$

Dans notre cas, les C classes sont présentes dans chaque lot et nous utilisons le même nombre d'exemples $m_c = 10$ pour chacune, avec $m_c^s = 7$ et $m_c^q = 3$. Les annotations faibles ont un impact à plusieurs moments du calcul. Lors du calcul des vecteurs prototypes, la densité d'événement D_c est moyennée entre les différents exemples utilisés, de sorte que la densité D_c des vecteurs prototypes n'a pas une valeur extrême (ni 0, ni 1). Ensuite, nous ne nous intéressons pas à la distance absolue entre un exemple requête et un vecteur prototype, mais à sa distance relative par rapport à chacun des autres vecteurs prototypes. Cela peut réduire l'impact des annotations faibles si les classes d'événements sont distinctes entre elles sur cet aspect. Par exemple, dans le cas où une classe d'événement ne contient que des événements courts alors que les autres classes contiennent toutes des événements longs, un exemple requête de la classe d'événements courts aura une distance plus faible avec le vecteur prototype de la classe d'événements courts si la distance entre les bruits de fonds est moins importante que la distance entre le bruit de fond et les événements longs.

6.3.4 Métrique de validation

Dans le cas de l'apprentissage par triplets ou par réseau prototype, la fonction de coût à l'apprentissage ne correspond pas à notre application finale. Il nous faut donc définir une métrique de validation indépendante du coût d'apprentissage afin de juger de la même façon la qualité des représentations de haut niveau pour les différents modèles et de déterminer quand arrêter l'apprentissage. Alors que les coûts d'apprentissage doivent être différentiables, la métrique de validation n'a pas besoin de l'être. Dans le chapitre 5, nous avons utilisé un classifieur appris toutes les deux époques. Ce principe est très coûteux puisque l'apprentissage du classifieur est parfois plus long que les deux époques

d'apprentissage de la représentation de haut niveau elles-mêmes. Pour résoudre ce problème, nous considérons simplement la moyenne des représentations de haut niveau de chaque classe :

$$\mathbf{cent}_c = \frac{1}{|\mathcal{J}_c|} \sum_{\mathbf{X} \in \mathcal{J}_c} E(\mathbf{X}) \quad (6.8)$$

où \mathbf{cent}_c est le centroïde de la classe c , \mathcal{J}_c est le sous-ensemble de \mathcal{J} contenant les exemples \mathbf{X} de la classe c et $|\mathcal{J}_c|$ représente la taille de cet ensemble. Notons que ce calcul est similaire au calcul des prototypes (6.5) mais appliqué ici sur l'ensemble du jeu de données. Nous définissons la métrique de validation suivante qui indique si, pour chaque exemple (\mathbf{X}, \mathbf{w}) , le centroïde \mathbf{cent}_c le plus proche de \mathbf{X} est celui correspondant à la classe d'événement présente dans \mathbf{w} :

$$F(\mathbf{X}) = \begin{cases} 1 & \text{si } \arg \min_c \|E(\mathbf{X}) - \mathbf{cent}_c\|_F^2 = \arg \max_c w_c \\ 0 & \text{sinon.} \end{cases} \quad (6.9)$$

Cette métrique est motivée par le fait que nous voulons que les représentations de haut niveau des différentes classes forment des clusters distincts. En effet, si chaque point est plus proche du cluster de sa classe que des clusters des autres classes, nous devrions être capables de le classifier de façon correcte. Il serait possible d'étendre cette métrique à un cas avec plusieurs événements par clip grâce à l'utilisation de centroïdes pour chaque combinaison de classes ou bien en cherchant à attribuer les exemples aux clusters les plus proches parmi les C présents devant correspondre aux différentes classes présentes dans le clip audio. Nous utilisons l'équation (6.9) pour déterminer quand arrêter l'apprentissage du modèle E .

6.4 Impact attendu des annotations faibles sur l'agrégation temporelle

L'obtention des scores au niveau du clip $\hat{\mathbf{w}} = [\hat{w}_1, \dots, \hat{w}_C]^T$ à partir des scores au niveau des fenêtres nécessite une agrégation temporelle. Dans cette partie, nous présentons des méthodes d'agrégation populaires et une nouvelle méthode d'agrégation L_p . Nous discutons l'impact attendu des annotations faibles sur les scores agrégés en fonction de la densité d'événement. Les méthodes étudiées sont illustrées dans la Figure 6.4, où \mathbf{O} de taille $N \times T$ est la représentation de haut niveau en sortie du modèle E et $\hat{\mathbf{Y}}$ de taille $C \times T$ est la matrice représentant les scores au niveau des fenêtres $\hat{y}_{c,t} = P(y_{c,t} = 1 | \mathbf{X})$. La matrice $\hat{\mathbf{Y}}$ a une dimension identique à celle de \mathbf{Y} et est obtenue en appliquant à \mathbf{O} une couche linéaire distribuée dans le temps suivie d'une activation sigmoïde. Sauf indication contraire, comme nous cherchons à comprendre l'impact des annotations faibles, le coût est toujours calculé au niveau des clips après agrégation.

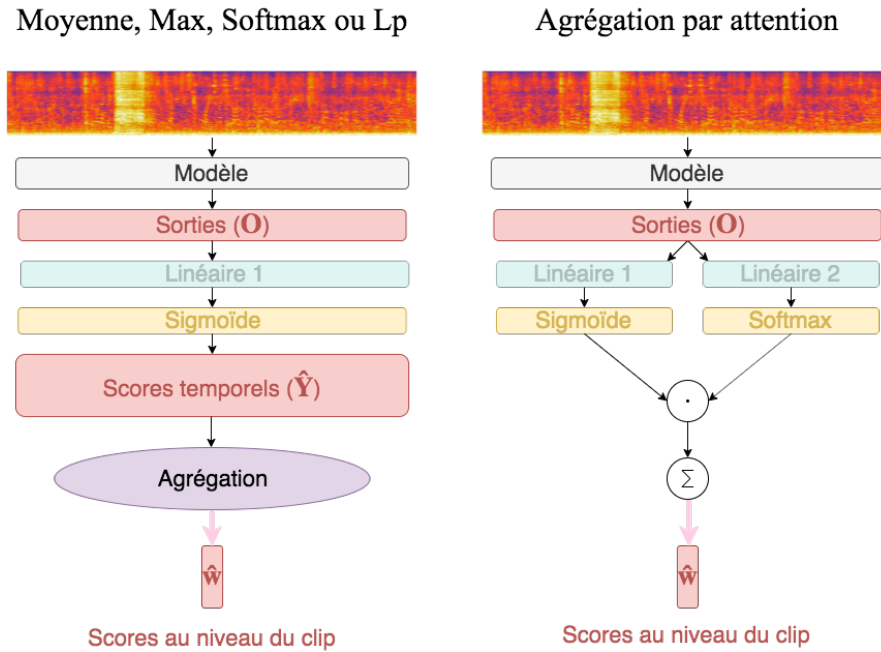


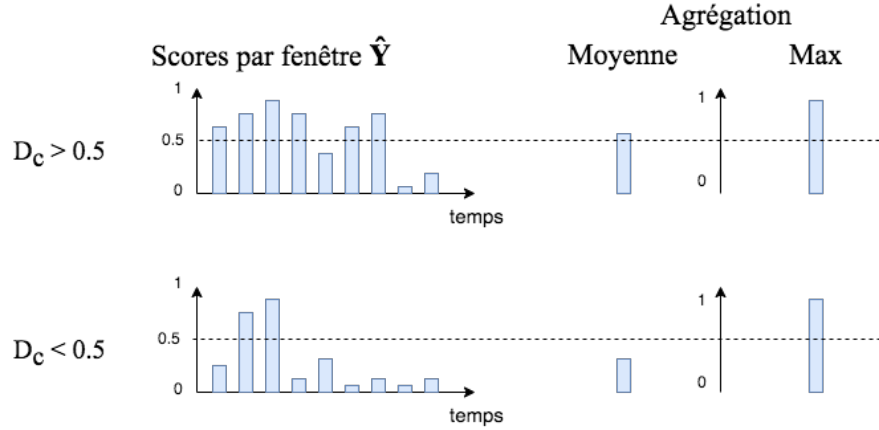
FIGURE 6.4 – Méthodes d'agrégation utilisées.

6.4.1 Agrégation moyenne

L'agrégation moyenne est une méthode d'agrégation courante qui moyenne les scores par fenêtre selon l'axe temporel :

$$\hat{w}_c = \frac{1}{T} \sum_{t=1}^T \hat{y}_t. \quad (6.10)$$

Lorsque $D_c \neq 0$, cette méthode présente un inconvénient majeur : plus D_c est faible, plus le score \hat{w}_c sera faible puisqu'il est borné par D_c qui est également calculée en moyenne sur le clip. Afin de mieux comprendre ce phénomène, nous pouvons considérer un scénario où le système estime parfaitement les scores par fenêtre $\hat{y}_{c,t} = y_{c,t}$, de sorte que $\hat{w}_c = D_c$. Nous notons par τ le seuil de décision et $\hat{w}_c \geq \tau$ définit les événements qui sont considérés comme actifs. Dans ce cas, lorsque $D_c < \tau$, l'événement est faussement étiqueté comme absent (le taux de FN est supérieur à 0), bien que les scores au niveau des fenêtres soient exacts. En théorie, la valeur $D_c = 0,5$ est le point de basculement, mais en pratique, la valeur de D_c qui implique ce phénomène varie étant donné que les scores au niveau des fenêtres ne sont pas parfaits. Abaisser le seuil de décision τ abaisse le taux de FN mais augmente le taux de FP. C'est pourquoi, l'agrégation moyenne n'est pas un bon choix, excepté pour des événements très longs qui ont donc une haute densité $D_c \simeq 1$.

FIGURE 6.5 – Agrégation moyenne et max ($\tau = 0,5$).

6.4.2 Agrégation max

L'agrégation max est définie ainsi :

$$\hat{\mathbf{w}} = \max_{t \in \{1, \dots, T\}} \hat{\mathbf{y}}_t \quad (6.11)$$

où le maximum est calculé par classe, c'est à dire que le score au niveau du clip pour chaque classe correspond au score de la fenêtre ayant la plus grande valeur, comme nous le représentons dans la Figure 6.5. Cette méthode d'agrégation semble appropriée à première vue puisqu'elle est utilisée pour convertir les annotations fortes en annotations faibles dans l'équation (6.11). Elle présente cependant deux inconvénients en pratique. Le premier est sa sensibilité aux FP : un FP au niveau d'une ou plusieurs fenêtres se traduit systématiquement par un FP au niveau du clip. Le deuxième inconvénient concerne la non-différentiabilité des sorties par rapport à l'ensemble des entrées. Lors de l'apprentissage, le gradient est seulement propagé aux entrées dans la fenêtre correspondant au score maximal, ce qui peut causer des problèmes d'apprentissage comme l'expliquent McFee et al. [2018].

6.4.3 Agrégation softmax

L'agrégation softmax est définie par :

$$\hat{\mathbf{w}} = \sum_{t=1}^T \hat{\mathbf{y}}_t \odot \frac{\exp(\hat{\mathbf{y}}_t)}{\sum_{t'=1}^T \exp(\hat{\mathbf{y}}_{t'})} \quad (6.12)$$

où \odot représente la multiplication terme-à-terme (produit de Hadamard), et les opérations de division et l'exponentielle sont aussi calculées terme-à-terme. Cette méthode peut être vue comme un compromis entre l'agrégation moyenne et l'agrégation max : elle est différentiable par rapport à l'ensemble des entrées comme l'agrégation moyenne et se

rapproche de l'agrégation max dans la façon d'agrèger les scores des fenêtres. Malgré son utilisation répandue au sein de la communauté, cette agrégation manque de flexibilité puisqu'elle ne permet pas de gérer le compromis entre la moyenne et le max. Lorsque $D_c \simeq 1$, nous nous attendons à ce que l'agrégation softmax fonctionne bien tout comme l'agrégation moyenne. Lorsque $D_c < \tau$, l'agrégation softmax continue à donner des scores corrects au niveau du clip alors que l'agrégation moyenne n'est plus adaptée. Cependant, lorsque D_c est trop faible, l'agrégation softmax devient elle aussi sujette aux FN alors que l'agrégation max peut rester adaptée. Lors de l'utilisation de l'agrégation softmax, les fenêtres FP avec un score élevé peuvent avoir un impact important dans le score agrégé du clip (comme pour le max), mais un nombre important de fenêtres FP avec des scores plus faibles peu aussi présenter un impact négatif (comme pour la moyenne).

6.4.4 Agrégation L_p

L'agrégation L_p est inspirée du calcul de la norme L_p . La différence se situe dans l'utilisation de la moyenne des valeurs à la puissance p plutôt que la somme. Cette agrégation peut être vue comme un autre compromis entre les agrégations moyenne et max. Nous avons introduit cette méthode d'agrégation dans le cadre du Challenge DCASE pour fusionner les scores issus de différents signaux lorsque la détection d'événements sonores est précédée d'une étape de séparation de sources [Turpault et al., 2020b]. Nous proposons de l'utiliser ici comme fonction d'agrégation temporelle :

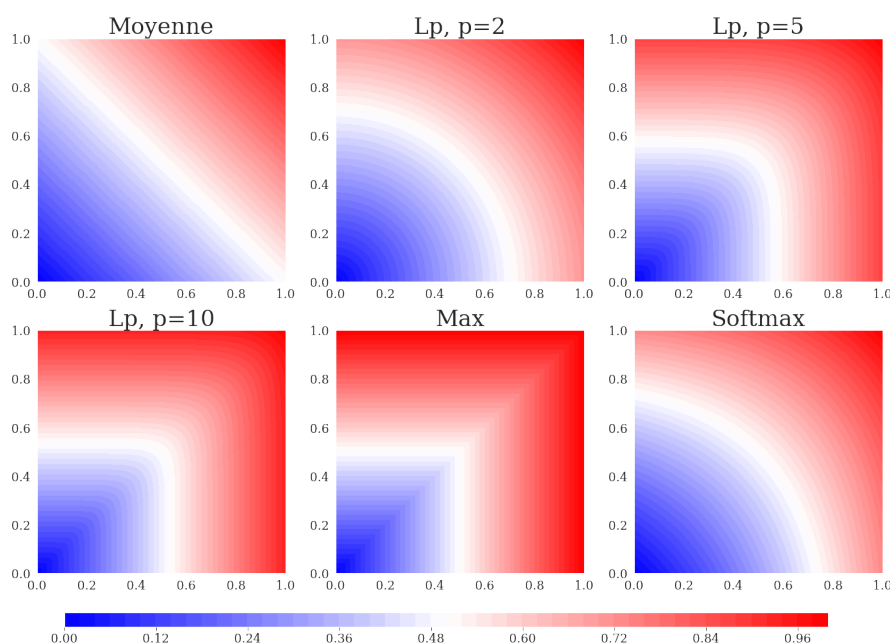
$$\hat{\mathbf{w}} = \left(\frac{1}{T} \sum_{t=1}^T \hat{\mathbf{y}}_t^p \right)^{\frac{1}{p}} \quad (6.13)$$

où $p > 0$ et les exponentielles sont calculées terme-à-terme. Lorsque $p = 1$, l'agrégation L_p est équivalente à l'agrégation moyenne. Lorsque $p \rightarrow \infty$, son comportement devient semblable à l'agrégation max tout en restant différentiable par rapport à l'ensemble des entrées. Cette solution offre une certaine flexibilité puisqu'il est possible de faire varier p en fonction de la classe d'intérêt et de l'application visée. Dans la Figure 6.6 nous représentons la fonction d'agrégation L_p pour différentes valeurs de p ainsi que les agrégations présentées précédemment. Nous pouvons observer que l'agrégation softmax est proche de l'agrégation L_p pour $p = 2$. Nous nous attendons à ce qu'une faible valeur de p soit préférable lorsque D_c est grand, et inversement.

6.4.5 Agrégation par attention

L'agrégation par attention est devenue une méthode d'agrégation populaire dans les travaux récents [Adavanne et al., 2019a, Kim and Ghaffarzadegan, 2019, Kong et al., 2019, Yu et al., 2018], et a été déclinée en de multiples variantes. Dans ce travail, nous nous intéressons à la variante la plus courante :

$$\hat{\mathbf{w}} = \sum_{t=1}^T \hat{\mathbf{y}}_t \odot \text{softmax}_t(\mathbf{A}) \quad (6.14)$$

FIGURE 6.6 – Comparaison de l’agrégation L_p avec les autres fonctions d’agrégation fixes.

où

$$\text{softmax}_t(\mathbf{A}) = \frac{\exp(\mathbf{a}_t)}{\sum_{t'=1}^T \exp(\mathbf{a}_{t'})}, \quad (6.15)$$

toutes les opérations sont terme-à-terme et $\mathbf{a}_t = \mathbf{W}_2 \mathbf{o}_t + \mathbf{b}_2$ est obtenue à partir d’une couche linéaire dont les poids \mathbf{W}_2 et biais \mathbf{b}_2 sont différents de ceux utilisés pour obtenir $\hat{\mathbf{y}}_t$. Par rapport aux méthodes d’agrégation ci-dessus qui n’ont pas de paramètres appris² l’agrégation d’attention a un nombre important de paramètres appris $\{\mathbf{W}_2, \mathbf{b}_2\}$. Il est donc attendu qu’elle soit bénéfique dans un nombre plus important de situations et pour différents valeurs de D_c [Adavanne et al., 2019a, Kim and Ghaffarzadegan, 2019].

6.5 Description des expériences

6.5.1 Apprentissage de représentation

Dans la partie 6.6.1, nous évaluons l’impact des annotations faibles sur les trois méthodes d’apprentissage de représentation de la partie 6.3. L’ensemble des expériences est basé sur les architectures de réseaux E et G décrites ci-dessous. Le code des expériences est disponible publiquement³.

2. autres que les poids et biais $\{\mathbf{W}_1, \mathbf{b}_1\}$ utilisés pour obtenir $\hat{\mathbf{Y}}$ à partir de \mathbf{O} , qui ne font pas partie de l’agrégation elle-même.

3. <https://github.com/turpaultn/walle>

Représentation de bas niveau L'ensemble des clips a une fréquence d'échantillonnage de 16 kHz. La représentation de bas niveau utilisée est un spectrogramme log-Mel. La transformée de Fourier à court terme de chaque clip est calculée sur des trames de 25 ms avec un pas de 10 ms avant d'appliquer un banc de 64 filtres en échelle Mel et de calculer le logarithme des valeurs ainsi obtenues.

Représentation de haut niveau Le réseau E permettant d'extraire la représentation de haut niveau est un CNN avec 4 couches composées respectivement de 16, 16, 32 et 65 filtres de taille 3×3 . Il opère sur des fenêtres constituées de 20 trames successives (200 ms). Une agrégation max est utilisée entre chaque couche sur des voisinages de taille (2, 2), (2,2), (1, 4) et (1, 2), où la première dimension correspond au facteur temporel et la deuxième dimension au facteur fréquentiel. Entre l'entrée et la sortie, le nombre de trames est donc divisé par 4 et le nombre de bandes fréquentielles par 32. À la sortie, les 5 trames restantes sont moyennées, de sorte à obtenir un vecteur de taille 130 (constitué des 65 canaux dans les 2 bandes fréquentielles restantes) pour chaque fenêtre de 200 ms.

Classifieur L'architecture de classifieur G utilisée pour l'apprentissage bout-en-bout et pour l'évaluation des représentations apprises par triplets ou par réseau prototype est composée d'une couche totalement connectée de taille 32 suivie d'une couche de taille 10 avec une fonction d'activation sigmoïde. Bien que nos clips ne comportent qu'une seule classe d'événement, nous utilisons la fonction sigmoïde plutôt que softmax afin que le modèle corresponde au cas général où chaque clip comporte plusieurs classes d'événements. La taille de la couche totalement connectée a été optimisée grâce à l'algorithme de bisection d'Orion⁴ [Li et al., 2017a].

Agrégation Ce classifieur détecte la présence des 10 classes d'intérêt sur chaque fenêtre. Les scores au niveau des fenêtres sont agrégés au niveau du clip en utilisant l'agrégation moyenne.

6.5.2 Impact de l'agrégation et de la durée des clips

Dans les parties 6.6.2 et 6.6.3, nous comparons les différentes méthodes d'agrégation. En raison des limites de l'apprentissage par triplets ou par réseau prototype montrées dans la partie 6.6.1, nous nous concentrons sur l'apprentissage de représentation de bout-en-bout. Pour cela, nous utilisons le système de référence de la Tâche 4 du Challenge DCASE 2020 décrit dans la partie 4.1.3. Ce système est un CRNN qui diffère de celui utilisé pour l'apprentissage de bout-en-bout dans la partie 6.6.1 sur plusieurs points. Par exemple, la représentation de bas niveau a 128 bandes au lieu de 64, le CNN comporte 7 couches au lieu de 4, et le classifieur est un RNN bidirectionnel à 2 couches au lieu d'un réseau totalement connecté sur chaque fenêtre. Nous optons pour ce modèle car ses résultats sont proches de l'état de l'art et nous l'avons déjà analysé en détail dans le cadre du Challenge DCASE.

4. <https://github.com/Epistimio/orion>

Dans un premier temps, nous utilisons le modèle pré-appris pour la tâche de détection d'événements sonores sur les données hétérogènes de la Tâche 4 qui comprennent des données enregistrées⁵, et nous l'évaluons sur les jeux de données AAF, ETAAF et DAAF afin d'analyser l'impact des annotations faibles. Le fait que ce modèle soit pré-appris pour une tâche de détection d'événements sonores favorise une bonne segmentation, ce qui permet d'évaluer le rôle de l'agrégation lors de la phase d'évaluation.

Dans un second temps, nous ré-apprenons ce modèle pour la tâche d'étiquetage d'événements sonores sur les jeux de données AAF, ETAAF ou DAAF. La fonction d'agrégation utilisée dans ce cas peut différer entre les phases d'apprentissage et d'évaluation afin d'évaluer le rôle de l'agrégation dans chacune des deux phases.

6.5.3 Métrique d'évaluation

La métrique d'évaluation est la F-mesure par clip. Cependant, le calcul de l'intervalle de confiance change entre les expériences concernant l'apprentissage de représentation (partie 6.6.1) et celles concernant l'impact de l'agrégation et de la durée des clips (parties 6.6.2 et 6.6.3). Pour l'évaluation de l'apprentissage de représentation, chaque modèle est entraîné 3 fois pour chaque expérience afin de déterminer un intervalle de confiance de 90%. Pour l'évaluation de l'impact de l'agrégation et de la durée des clips, afin d'éviter l'apprentissage de 3 modèles pour chaque expérience, l'intervalle de confiance de 90% est calculé en utilisant la méthode de « *bootstrap* » [DiCiccio and Efron, 1996] sur 200 itérations avec 80% des données par itération. Dans le premier cas, l'intervalle de confiance représente l'incertitude due à l'initialisation et à l'optimisation du modèle alors que, dans le deuxième cas, il représente l'incertitude due à la variété des exemples d'évaluation.

6.6 Résultats et discussion

6.6.1 Apprentissage de représentation

Dans la partie 6.3, nous avons décrit trois méthodes pour apprendre des représentations de haut niveau. Dans cette partie, nous évaluons la performance d'étiquetage d'événements sonores obtenue sur le jeu d'évaluation par le classifieur final G utilisant ces représentations en entrée. Les marges d'erreur sont calculées à partir de 3 apprentissages du modèle E , comme expliqué dans la partie 6.5.3. Nous ne sommes pas intéressés par la comparaison des F-mesures absolues obtenues par les trois méthodes, mais plutôt par leur comportement envers les annotations faibles.

Dans le Tableau 6.2, nous présentons les résultats des différents modèles sur le jeu de données ETAAF avec des événements tronqués à 0,2 s en faisant varier la durée des clips audio à l'apprentissage et l'évaluation. Les lignes représentent l'apprentissage sur une durée de clip fixe en faisant varier la durée des clips d'évaluation. Les colonnes représentent l'évaluation sur une durée de clip fixe en faisant varier la durée des clips d'apprentissage. Les durées de clips évalués sont de 200 ms, 1 s et 10 s, ce qui correspond

5. Voir le chapitre 3 pour une description détaillée.

Méthode	Durée du clip d'apprentissage	Durée du clip d'évaluation		
		200 ms	1 s	10 s
Classifieur	200 ms	45,8±2,9	29,6±1,7	3,7±0,5
	1 s	44,2±1,8	47,4±3,2	12,7±2,4
	10 s	39,8±1,9	49,3±3,2	36,7±3,8
Triplets	200 ms	42,5±1,0	2,6±0,4	0,0±0,0
	1 s	39,1±2,4	28,9±2,7	0,1±0,1
	10 s	0,0±0,0	0,0±0,0	0,0±0,0
Prototypes	200 ms	41,2±3,5	9,4±2,7	0,0±0,0
	1 s	38,8±1,8	36,1±2,1	1,1±1,3
	10 s	0,0±0,0	0,0±0,0	0,0±0,0

TABLEAU 6.2 – F-mesure (%) obtenue par les trois méthodes d'apprentissage de représentation sur le jeu de données ETAAF selon la durée des clips d'apprentissage et d'évaluation.

respectivement à $D_c = 1$, $D_c = 0,2$ et $D_c = 0,02$. Lorsque les modèles sont appris sur des clips de 200 ms (première ligne pour chaque méthode) et évalués sur des clips de 1 s ou 10 s, nous observons de faibles performances, ce qui indique une faible robustesse au bruit de fond lors de l'évaluation. Cela signifie qu'avoir des annotations fortes et apprendre un modèle uniquement sur les événements segmentés ne permet pas de détecter correctement ces événements dans des clips audio non segmentés. Apprendre les modèles avec des annotations faibles a aussi un impact lorsqu'on les évalue avec des données segmentées (première colonne). Cet impact est bien plus visible avec l'apprentissage par triplets ou par réseau prototype qu'avec l'apprentissage de bout-en-bout. Il est possible que, dans le cas de l'apprentissage par triplets ou par réseau prototype, la présence de clips ayant une densité d'événement D_c faible se traduise par l'apprentissage d'une représentation qui discrimine les bruits de fonds plutôt que les événements sonores cibles. Au contraire, l'apprentissage de bout-en-bout sur des clips de 10 s améliore la performance quand les données d'évaluation ne sont pas segmentées (1 s et 10 s). Ceci peut signifier que le bruit de fond a un effet de régularisation sur le modèle qui est appris sur une faible quantité de données (ceci est encore plus vrai pour des clips courts) ou que l'observation de fenêtres contenant uniquement du bruit de fond à l'apprentissage aide à reconnaître les événements dans un tel bruit à l'évaluation.

Le Tableau 6.3 présente les performances obtenues sur le jeu de données AAF lorsque l'on varie la taille des clips à l'apprentissage et à l'évaluation. Pour chaque modèle, l'apprentissage du système sur des clips d'1 s et son évaluation sur des clips d'1 s donne les meilleurs résultats. Ceci est à mettre en parallèle avec la Figure 6.2 qui montre que la durée de la plupart des événements courts se situe autour d'1 s, donc le biais des annotations introduit lors de l'apprentissage avec des clips d'1 s est faible.

Nous pouvons comparer les résultats obtenus entre les Tableaux 6.2 et 6.3. Sur le jeu de

Méthode	Durée du clip d'apprentissage	Durée du clip d'évaluation		
		200 ms	1 s	10 s
Classifieur	200 ms	45,8 ± 2,9	49,0±4,1	26,8±3,1
	1 s	46,9±1,2	57,5±2,5	38,0±1,9
	10 s	40,2±2,0	54,2±0,7	51,0±2,3
Triplets	200 ms	42,5±1,0	38,2±3,6	11,7±3,2
	1 s	41,7±7,0	44,8±10,9	18,3±7,3
	10 s	9,1±3,2	10,2±2,0	2,8±0,7
Prototypes	200 ms	41,2±3,5	36,1±7,3	9,5±4,3
	1 s	45,2±0,4	52,4±3,9	22,0±3,4
	10 s	29,9±6,2	35,8±10,9	28,6±11,0

TABLEAU 6.3 – F-mesure (%) obtenue par les trois méthodes d'apprentissage de représentation sur le jeu de données AAF selon la durée des clips d'apprentissage et d'évaluation.

données ETAAF avec des événements tronqués à 200 ms, augmenter la durée des clips d'apprentissage de 200 ms à 1 s dégrade la performance sur les clips d'évaluation de 200 ms, ce qui s'explique par le fait que 4 fenêtres sur 5 contiennent seulement du bruit de fond. Sur le jeu de données AAF avec des événements non-tronqués, l'apprentissage sur des clips de 1 s fonctionne aussi bien (triplets) voire mieux (prototypes, bout-en-bout) sur les clips d'évaluation de 200 ms que l'apprentissage sur des clips de 200 ms. Ceci tend à indiquer que les clips de 200 ms ne sont probablement pas assez longs pour identifier correctement les différentes classes. Nous pouvons aussi formuler l'hypothèse qu'à l'inverse 1 s est une durée suffisante pour avoir assez d'information sur les événements longs. Lorsque l'on apprend les modèles avec des clips de 10 s, les performances des méthodes d'apprentissage par triplets ou par réseau prototype se dégradent sévèrement quelle que soit la durée des clips d'évaluation alors que la méthode d'apprentissage de bout-en-bout maintient une performance correcte. L'explication est la même que pour le Tableau 6.2 : les clips longs contiennent probablement trop de bruit de fond pour permettre de distinguer les événements sonores à l'aide d'une simple distance. L'apprentissage de bout-en-bout peut faire plus facilement abstraction du bruit de fond en apprenant à donner des poids plus importants aux fenêtres qui contiennent nos événements sonores. Le classifieur réalise donc une segmentation implicite.

En résumé, ces deux premières expériences montrent que l'apprentissage de représentation par triplets ou par réseau prototype est fortement affecté par une faible densité d'événement, que ce soit à l'apprentissage ou à l'évaluation. Le modèle appris de bout-en-bout est plus robuste, particulièrement à l'apprentissage.

6.6.2 Impact de l'agrégation

Afin d'analyser plus en détail l'impact des annotations faibles sur un modèle appris de bout-en-bout, nous nous concentrons à partir de maintenant sur ce type de modèle et nous évaluons l'impact de la méthode d'agrégation choisie pour passer des scores au niveau des fenêtres aux scores au niveau du clip, à l'apprentissage comme à l'évaluation. Pour cela, nous utilisons le système de référence de la Tâche 4 du Challenge DCASE 2020, soit pré-appris sur le corpus DESED de la Tâche 4 soit ré-appris uniquement sur les données synthétiques de ce chapitre.

6.6.2.1 Impact de l'agrégation à l'évaluation

Notre première expérience utilise les scores au niveau des fenêtres estimés par le système de référence pré-appris sur le corpus DESED. La Figure 6.7 évalue les fonctions d'agrégation-

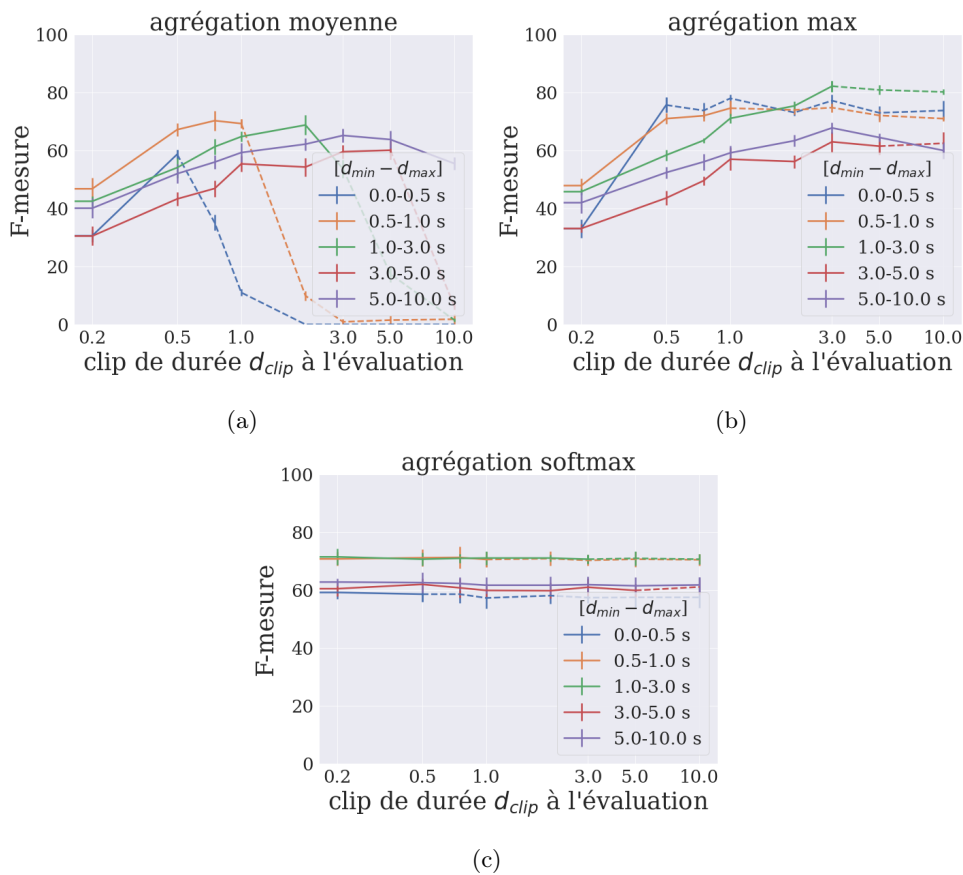


FIGURE 6.7 – F-mesure (%) du système de référence pré-appris sur le corpus DESED et évalué sur le jeu de données AAF selon la durée de l'événement ($d_{min}-d_{max}$) et du clip (d_{clip}) pour les fonctions d'agrégation moyenne, max et softmax. Les lignes pointillées indiquent que $D_c < 1$ pour tous les clips d'évaluation.

tion les plus classiques (moyenne, max, softmax) en fonction de la durée de l'événement et du clip d'évaluation sur le jeu de données AAF. Les lignes pointillées indiquent la présence d'annotations faibles ($D_c < 1$). Les résultats confirment le comportement attendu discuté dans la partie 6.4. L'agrégation moyenne (Figure 6.7a) n'est pas efficace lorsque $D_c < 1$. L'agrégation max est très efficace dans ce scénario car elle n'est utilisée qu'à l'évaluation (le problème de non-différentiabilité n'apparaît qu'à l'apprentissage). Le meilleur résultat est obtenu pour $d_{\text{clip}} = 3$ s et, sa performance se dégrade sur des clips plus longs car la probabilité que le maximum soit le score d'une fenêtre FP augmente. L'agrégation softmax est peu affectée par la durée des clips. Cela en fait une fonction d'agrégation intéressante lorsqu'on ne connaît pas D_c a priori. Cependant sa performance absolue n'est pas toujours aussi bonne que l'agrégation max, en particulier pour les événements de durée inférieure à 3 s.

La Figure 6.8 compare l'agrégation par attention, l'agrégation L_p pour différentes valeurs de p variant de 1 (agrégation moyenne) à 100 (proche de l'agrégation max), et l'agrégation max. L'agrégation par attention est notée « att ». Cette couche d'attention est celle utilisée par le système de référence développé pour la détection d'événements sonores et n'a été apprise que sur les données faiblement annotées du corpus DESED. Ces données ne correspondent qu'à 10% de l'ensemble d'apprentissage, ce qui peut expliquer la faible performance obtenue. En ce qui concerne l'agrégation L_p et l'agrégation max, les Figures 6.8a et 6.8b montrent que, lorsque $d_{\text{clip}} \lesssim 0,5$ s, la fonction d'agrégation utilisée a peu d'impact, ce qui est normal puisque pour la majorité des clips $D_c \simeq 1$. De plus, $d_{\text{clip}} = 0,2$ s semble être une durée trop courte pour permettre de reconnaître les événements d'intérêt. Dans la Figure 6.8c correspondant à $d_{\text{clip}} = 1$ s, nous observons un comportement similaire pour les clips contenant des événements de durée supérieure à 0,5 s avec une performance globale plus élevée. Pour les clips contenant des événements de durée inférieure à 0,5 s en revanche, une valeur faible de p dégrade la performance, ce qui était également le cas avec la moyenne. L'explication se situe dans le fait que ces clips ont une densité d'événement $D_c < 0,5$ avec une moyenne pour l'ensemble de ces clips de $D_c = 0,32$, ce qui implique que, même si les scores au niveau des fenêtres sont bons, les scores au niveau du clip sont erronés avec p faible. Lorsque $d_{\text{clip}} > 1$ s, la valeur de p commence à devenir plus importante, en particulier pour les événements courts, puisque ceux-ci nécessitent une valeur de p importante pour être détectés correctement en raison de la faible densité d'événement des clips. D'après l'ensemble de ces figures, nous constatons que l'agrégation L_p avec $p = 5$ ou $p = 10$ est une bonne alternative à l'agrégation max pour l'étiquetage des événements courts dans des clips longs.

6.6.2.2 Impact de la segmentation à l'apprentissage

Afin d'évaluer l'impact des annotations faibles à l'apprentissage, nous ré-apprenons le système de référence sur le jeu de données AAF et nous définissons tout d'abord un scénario « idéal » au sens où il ne contient pas d'annotations faibles ($D_c = 1$). Pour définir un tel scénario, une façon naturelle de procéder serait de segmenter les données afin de ne conserver que l'événement d'intérêt dans chaque clip. Techniquement, l'apprentissage sur des lots contenant des signaux de durées différentes requiert l'accumulation du gradient

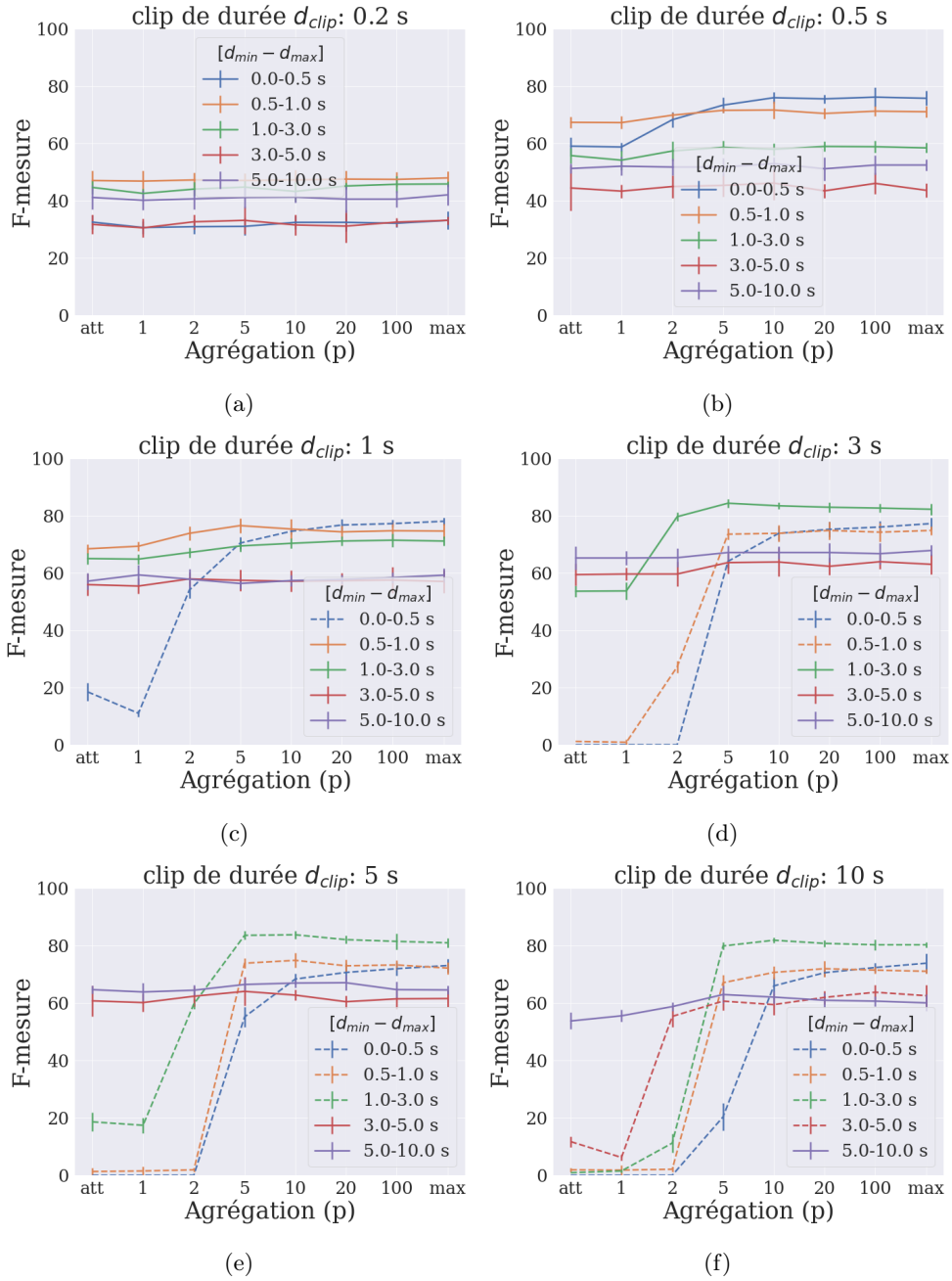


FIGURE 6.8 – F-mesure (%) du système de référence pré-appris sur le corpus DESED et évalué sur le jeu de données AAF selon la durée de l'événement ($d_{min}-d_{max}$) et du clip (d_{clip}) pour les fonctions d'agrégation max, L_p pour différentes valeurs de p , et par attention (« att »). Les lignes pointillées indiquent que $D_c < 1$ pour tous les clips d'évaluation.

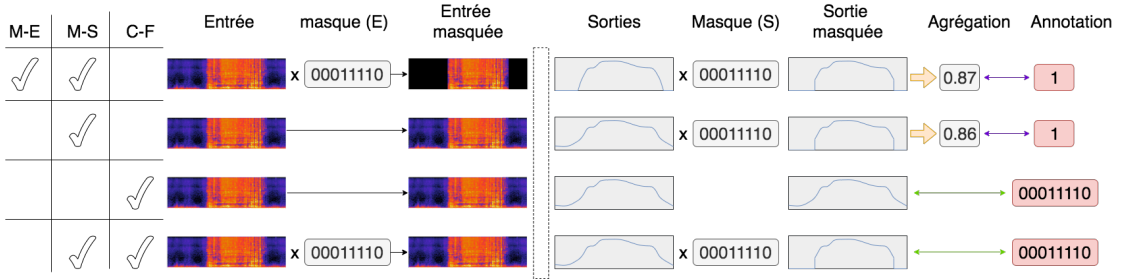


FIGURE 6.9 – Variantes possibles d'utilisation des annotations fortes. M-E signifie masque sur l'entrée, M-S signifie masque sur la sortie, C-F signifie coût au niveau des fenêtres et l'agrégation utilisée est l'agrégation moyenne. Le rectangle en pointillé représente le modèle. Les flèches jaunes représentent l'agrégation des fenêtres dans le masque, les flèches violettes représentent le coût au niveau du clip et les flèches vertes représentent le coût au niveau des fenêtres. La première ligne correspond au scénario idéal.

	Sans normalisation par lots	Avec normalisation par lots
Masquage	$54,9 \pm 1,2$	$70,2 \pm 1,2$
Segmentation	$40,1 \pm 1,4$	N/A

TABLEAU 6.4 – F-mesure (%) du système de référence ré-appris et évalué sur des données segmentées ou masquées (entrées + sorties) du jeu de données AAF, avec ou sans normalisation par lots.

sur des sous-lots de taille 1 et empêche l'usage de la normalisation par lot, ce qui dégrade la performance. Pour cette raison, nous ne modifions pas la durée des signaux et nous définissons le scénario idéal en appliquant un masque temporel binaire (égal à l'annotation forte de l'événement considéré) sur les entrées et les sorties du réseau à l'apprentissage et à l'évaluation, comme illustré sur la première ligne de la Figure 6.9. Le Tableau 6.4 valide expérimentalement cette définition. En effet, nous constatons que la segmentation a une performance inférieure au masquage sans normalisation par lots, elle-même inférieure au masquage avec normalisation par lots. L'importance de la normalisation par lots pour notre système pourrait s'expliquer par l'utilisation du modèle de « professeur moyen » : l'accroissement du taux d'apprentissage au fil des époques va de pair avec l'augmentation du nombre d'époques, ce que permet la normalisation par lots.

Ce scénario idéal a été utilisé pour définir la taille du jeu d'apprentissage. Dans nos expériences préliminaires, nous avons identifié qu'au-delà de 1000 clips l'apprentissage donne des résultats semblables mais un nombre supérieur de clips donne des résultats plus stables. À partir de maintenant, nous considérons donc une version étendue du jeu d'apprentissage AAF comportant 5000 clips.

En partant de ce scénario idéal, diverses variantes illustrées dans la Figure 6.9 peuvent être explorées. Premièrement, en sus ou au lieu de leur utilisation pour le masquage, les annotations fortes peuvent être utilisées pour apprendre le modèle avec un coût au niveau des fenêtres au lieu d'un coût au niveau du clip. Le Tableau 6.5 évalue cette variante.

	Masquage	Clips de 10 s
Coût par clip (agrégation moyenne)	70,2 ± 1,2	62,2 ± 1,3
Coût par fenêtre	66,0 ± 1,1	56,5 ± 1,5

TABLEAU 6.5 – F-mesure (%) du système de référence ré-appris et évalué sur le jeu de données AAF avec un coût d'apprentissage au niveau des fenêtres ou au niveau du clip et un scénario où $D_c = 1$ (masquage entrées + sorties) ou $D_c \leq 1$ (clips de 10 s non masqués).

Les résultats indiquent que, indépendamment du masquage, l'utilisation d'un coût au niveau du clip est préférable. Ceci peut être expliqué par son optimisation plus facile et par l'effet de régularisation induit par le bruit de fond. En effet, lors de l'utilisation d'un coût au niveau du clip, l'optimisation du coût se fait après agrégation et ne nécessite pas une sortie parfaite pour chaque fenêtre. Lors de l'utilisation d'un coût au niveau des fenêtres, la minimisation du coût pour chaque fenêtre rend la tâche plus compliquée pour le modèle qui doit différencier chacune des fenêtres où la présence de l'événement est incertaine. La différence de performance plus importante lors de l'utilisation de clips non masqués ($D_c \leq 1$) que masqués ($D_c = 1$) indique que le biais introduit par bruit de fond autour de l'événement est augmenté.

6.6.2.3 Masquage des entrées et/ou des sorties à l'apprentissage et/ou à l'évaluation

Une autre variante illustrée dans la Figure 6.9 concerne le fait que le masque obtenu à partir des annotations fortes peut être appliqué aux entrées, aux sorties ou aux deux. Enfin, l'application du masque peut se faire à l'apprentissage, à l'évaluation ou aux deux. Le Tableau 6.6 évalue ces variantes. Nous constatons que les meilleurs résultats sont obtenus en masquant les entrées et les sorties lors de l'apprentissage et de l'évaluation. Ce résultat était attendu puisque c'est notre scénario idéal. Le masquage lors de l'apprentissage uniquement dégrade les performances de façon importante. En effet, si les données sont segmentées à l'apprentissage, le modèle n'apprend pas à distinguer les événements

		F-mesure
Masquage entrées + sorties	Apprentissage	33,8 ± 0,9
	Évaluation	69,8 ± 1,2
	Apprentissage + Évaluation (scénario idéal)	70,2 ± 1,2
Masquage sorties uniquement	Apprentissage	44,5 ± 1,4
	Évaluation	67,6 ± 1,1
	Apprentissage + Évaluation	70,2 ± 1,0

TABLEAU 6.6 – F-mesure (%) du système de référence ré-appris et évalué sur le jeu de données AAF en appliquant le masquage aux entrées et/ou aux sorties à l'apprentissage et/ou à l'évaluation.

sonores du bruit de fond qui sont tous deux présents au moment de l'évaluation. Ces résultats étaient attendus mais ne sont généralement pas indiqués quantitativement dans les articles. Ils prouvent que, du point de vue applicatif, il est plus important de savoir segmenter les données d'évaluation que celles d'apprentissage. Pour prendre un exemple concret, un modèle appris sur le corpus FSD50K [Fonseca et al., 2020], qui inclut peu de bruit autour des événements, ne sera pas capable de reconnaître des événements sonores épars dans des enregistrements continus, ce qui est commun dans les cas d'usage réels. De façon surprenante, le masquage des sorties uniquement (à l'apprentissage et à l'évaluation) obtient des performances aussi élevées que le scénario idéal. La différence entre le masquage à la sortie uniquement et le scénario idéal réside dans la quantité de données vue par le modèle lors de l'apprentissage. Cela signifie que le léger biais introduit par le bruit de fond entourant les événements à l'entrée ne semble pas être un problème. En effet, lors du masquage des sorties seulement, l'utilisation de convolutions et d'agrégations dans le modèle implique que le champ réceptif des sorties contient certaines fenêtres de bruit de fond autour des événements. Cette information de bruit de fond utilisée à l'apprentissage est même bénéfique lors de l'évaluation sur des données contenant du bruit de fond comme nous le constatons grâce aux lignes « Apprentissage » du tableau. Cependant, lors de l'évaluation, il est plus intéressant de masquer les entrées et les sorties que les sorties uniquement, ce qui tend à indiquer que la segmentation est importante au moment de l'évaluation, peu importe la méthode d'apprentissage utilisée.

6.6.2.4 Impact de l'agrégation à l'apprentissage

Une dernière variante du scénario idéal consiste à utiliser une fonction d'agrégation à l'apprentissage différente de l'agrégation moyenne utilisée dans les parties 6.6.2.2 et 6.6.2.3. Le Tableau 6.7 évalue l'usage de l'agrégation L_p à l'apprentissage pour différentes valeurs de p . L'agrégation moyenne est employée à l'évaluation dans tous les cas. Nous présentons les résultats pour le jeu de validation et le jeu d'évaluation lors de l'utilisation de clips de 10 s non masqués ou dans le scénario idéal. Nous constatons que le choix de la valeur de p a un impact limité lors de l'apprentissage. De plus, il existe une différence de performance importante entre les résultats sur le jeu de validation et le jeu d'évaluation,

p lors de l'apprentissage	Scénario			
	Clips de 10 s		Masquage	
	F-mesure (Validation)	F-mesure (Évaluation)	F-mesure (Validation)	F-mesure (Évaluation)
1	82,5 ± 1,0	62,2 ± 1,3	80,8 ± 0,9	70,2 ± 1,2
2	80,8 ± 1,1	61,8 ± 1,2	80,5 ± 0,9	71,1 ± 1,0
5	80,3 ± 0,9	61,4 ± 1,4	80,1 ± 1,3	73,8 ± 1,3

TABLEAU 6.7 – F-mesure (%) du système de référence ré-appris et évalué sur le jeu de données AAF dans le scénario idéal ($D_c = 1$) ou sur des clips de 10 s non masqués ($D_c \leq 1$), en utilisant l'agrégation L_p à l'apprentissage avec différentes valeurs de p . L'agrégation moyenne est employée à l'évaluation dans tous les cas.

qui est plus importante pour le jeu de données utilisant les clips de 10 s non masqués. Nous attribuons cette différence au fait que le modèle sur-apprend les bruits de fond utilisés à l'apprentissage. En effet, les bruits de fond utilisés à l'apprentissage et à la validation sont issus du même corpus initial. Les bruits de fond utilisés pour générer les clips d'évaluation proviennent eux d'un corpus différent. Dans la suite des expériences de ce chapitre, nous conservons l'usage de l'agrégation moyenne à l'apprentissage, qui fournit les meilleurs résultats sur le jeu de validation ($p = 1$).

6.6.3 Impact de la durée du clip et de la densité d'événement

6.6.3.1 Impact de la durée du clip

Dans ces expériences nous cherchons à connaître l'impact de la durée des clips sur le classifieur. Afin de comprendre ce qu'il se passe lors de l'utilisation d'annotations faibles, nous présentons différentes conditions. Premièrement, nous varions la durée des clips à l'apprentissage et à l'évaluation sans tronquer les événements (en utilisant le jeu de données AAF) afin de comprendre l'impact de la durée du clip sur le modèle. Comme indiqué dans la partie 6.6.2.4, nous utilisons l'agrégation moyenne lors de l'apprentissage. Cela permet au modèle de donner un poids égal à chaque fenêtre et aide à l'apprentissage d'une segmentation. Lors de l'évaluation, nous utilisons l'agrégation L_p avec $p = 5$ car nous avons identifié dans la partie 6.6.2.1 que cette valeur permet une agrégation proche d'une agrégation max sans être trop sensible aux FP. Cela évite les problèmes liés à l'agrégation moyenne et peut aider à palier les problèmes de segmentation du modèle. Des expériences préliminaires (non rapportées ici) nous a permis de valider ces combinaisons d'agrégation lors de l'apprentissage et de l'évaluation comme une des paires les plus performantes.

La Figure 6.10 représente la performance du modèle en fonction de la durée des clips d'apprentissage et d'évaluation. Les expériences sont réalisées sur le jeu de données AAF pour lequel les événements ne sont pas tronqués donc, selon que la durée initiale de

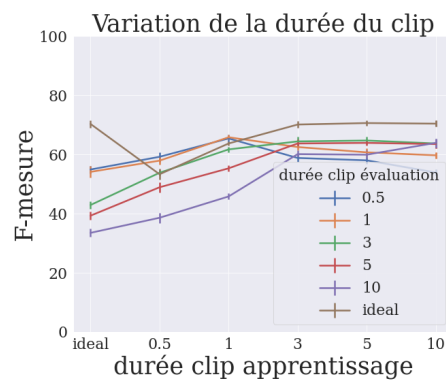


FIGURE 6.10 – F-mesure (%) du système de référence ré-appris et évalué sur le jeu de données AAF en fonction de la durée des clips d'apprentissage et d'évaluation.

l'événement est plus courte ou plus longue que la durée du clip, la quantité de bruit de fond peut augmenter en augmentant la durée du clip ou la quantité d'information peut elle aussi augmenter. Le premier point sur l'axe des abscisses correspond au scénario idéal lors de l'apprentissage. Nous constatons que la performance en ce point est la plus faible quel que soit le scénario d'évaluation, excepté lorsque celui-ci est également un scénario d'évaluation idéal. Ceci indique que l'apprentissage d'un modèle sur un jeu de données segmenté ne permet pas de généraliser à des conditions d'évaluation contenant du bruit de fond autour de l'événement ou lors desquelles seule une partie de l'événement d'intérêt est présent (événement long). La figure ne permettant pas de trancher entre ces deux conclusions, nous analysons cet aspect en détail dans les expériences suivantes. Nous observons aussi que le système obtient une meilleure performance lorsque les durées de clips sont proches à l'apprentissage et à l'évaluation. Finalement, nous identifions un point optimal pour une majorité des classes d'événements correspondant à une durée de clip de 3 s environ. Ce résultat avait déjà été identifié par [Salamon et al. \[2014\]](#), qui avaient montré que 4 s est une durée suffisante pour reconnaître un événement (même si celui-ci est plus long). C'est pour cette raison qu'ils ont défini le jeu de données Urban-Sound8k avec des clips de 4 s. Cependant, notons que la performance se dégrade pour les événements très courts ($\leq 0,5$ s) lorsque la durée du clip dépasse 1 s.

Pour aller plus loin, nous nous focalisons sur les durées de clip d'évaluation de 0,5 s, 3 s et une durée de clip égale à la durée de l'événement (cas idéal). Dans la Figure 6.11, nous rapportons la performance du modèle dans ces trois cas en fonction de la durée des clips d'apprentissage et de la durée effective des événements dans les clips d'évaluation. Notre but ici est de comprendre si les mauvaises performances obtenues en utilisant des clips courts lors de l'apprentissage sont dues au bruit des annotations (événements courts) ou bien à la troncature des événements (événements longs).

La Figure 6.11a présente les résultats pour des clips d'évaluation de 0,5 s. Nous constatons que la durée du clip d'apprentissage a peu d'importance lorsque l'évaluation est réalisée sur des clips courts, quelle que soit la durée des événements considérés. Ce n'est pas le cas lorsque l'évaluation est réalisée sur des clips plus longs. En effet, dans les Figures 6.11b et 6.11c nous observons que les modèles appris sur des clips courts peinent à reconnaître les événements sonores dans des clips longs, mais l'impact n'est pas le même en fonction de la durée effective des événements. Si nous nous concentrons sur la Figure 6.11b, nous observons que les événements courts (≤ 1 s) sont difficiles à reconnaître dans des clips de 3 s lorsque le modèle a été appris sur les événements segmentés (idéal) ou sur des clips de 0,5 s (qui contiennent peu de bruit). Cela signifie que la présence de bruit de fond à l'évaluation ($D_c \ll 1$) mais pas à l'apprentissage ($D_c \simeq 1$) dégrade la performance. Lorsque les clips sont de durée supérieure à 1 s lors de l'apprentissage, la performance reste stable jusqu'à des durées de clips d'apprentissage de 10 s quelle que soit la durée des événements. La Figure 6.11c confirme que, lorsque l'évaluation est effectuée dans un cas idéal, le meilleur système utilise des données d'apprentissage dans un scénario idéal, et ce, quelle que soit la durée des événements. L'utilisation des annotations faibles à l'apprentissage (lignes pointillées) tend à ne pas être un problème lorsque l'évaluation utilise des événements segmentés. En revanche l'utilisation d'événements tronqués à l'ap-

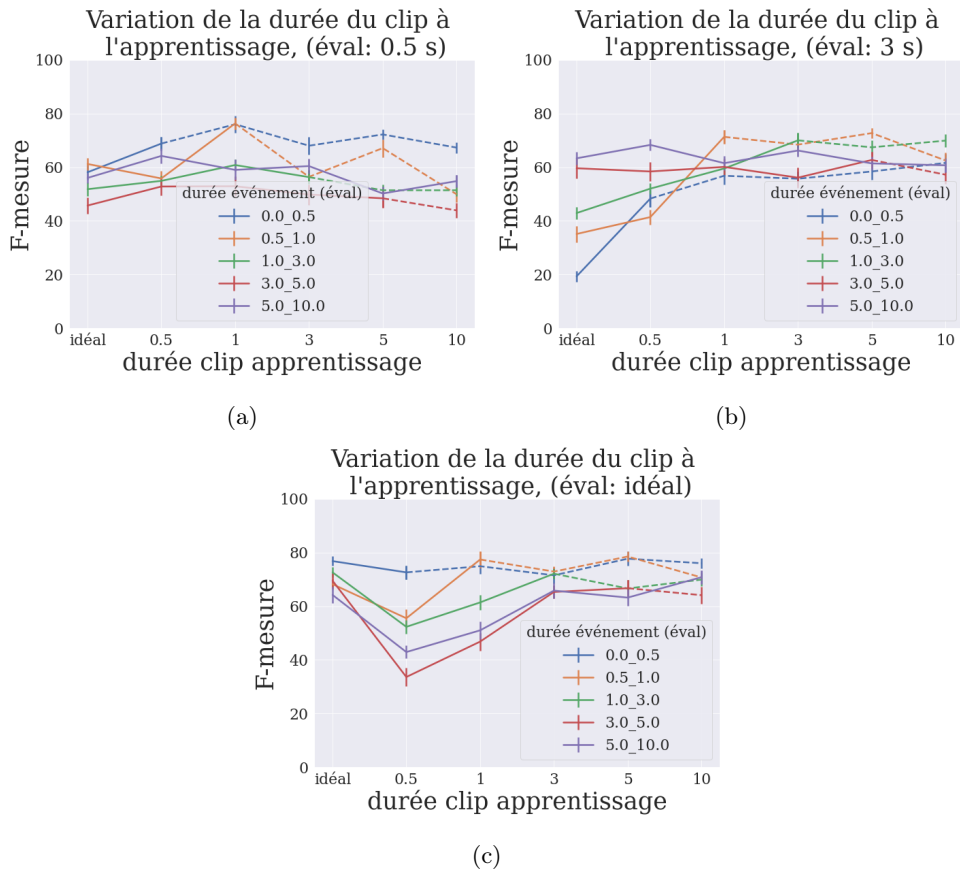


FIGURE 6.11 – F-mesure (%) du système de référence ré-appris et évalué sur le jeu de données AAF en fonction de la durée des clips d'apprentissage et de la durée des événements d'évaluation. Chaque figure représente une durée de clip d'évaluation (une courbe de la Figure 6.10). Les lignes pointillées indiquent que cette durée d'événements a été apprise uniquement avec des annotations faibles.

prentissage semble poser problème lorsque le système est évalué sur des clips segmentés. Ceci se traduit par les courbes qui croissent jusqu'au point où les clips d'apprentissage deviennent plus longs que l'événement d'intérêt.

Cette observation s'explique en comparant la position des courbes les unes par rapport aux autres dans les trois sous-figures. Si nous nous focalisons sur les deux premiers points de chaque sous-figure qui représentent une durée de clip égale à la taille de l'événement (segmenté) ou à 0,5 s, nous constatons que ces scénarios d'apprentissage sont favorables aux événements courts (≤ 1 s). Lorsque l'évaluation s'effectue avec des clips de 3 s, l'ordre des courbes change, ce qui signifie qu'il est plus difficile de reconnaître les événements courts et plus simple de reconnaître les événements longs ($D_c \simeq 1$). Dans l'ensemble, nos observations tendent à indiquer que, plus la densité d'événement ($D_c < 1$) diminue lors de l'évaluation alors qu'elle reste élevée lors de l'apprentissage, plus la performance

diminue. De plus, lorsque les clips sont de durée plus longue à l'évaluation qu'à l'apprentissage, les résultats diminuent. Autrement dit, le modèle est incapable de traiter des données d'évaluation avec un bruit de fond important (même avec une bonne méthode d'agrégation) s'il n'a pas observé ce type de données à l'apprentissage.

6.6.3.2 Impact de la densité d'événement

Impact de la valeur maximum de D_c à l'apprentissage Pour comprendre plus en détail l'impact de la quantité de bruit de fond à l'apprentissage lorsque du bruit de fond est présent à l'évaluation, nous ré-apprenons le modèle sur le jeu de données ETAAF en faisant varier la durée de troncature des événements pour contrôler la valeur maximale de D_c ⁶ et nous l'évaluons sur le jeu de données AAF (sans troncature des événements). Dans cette expérience, les clips ont une durée fixe de 10 s à l'apprentissage et à l'évaluation ; seule la durée des événements cibles est tronquée à une durée maximale lors de l'apprentissage. La Figure 6.12 présente les performances du modèle en fonction de la durée de troncature à l'apprentissage (l'événement peut tout de même être plus court que cette durée) et de la durée originelle de l'événement (avant troncature). Nous observons un contraste entre les événements très courts ($\leq 0,5$ s) et les autres événements. La troncature à 0,2 s et 0,5 s lors de l'apprentissage a peu d'impact quand les événements d'origine sont courts. Les événements plus longs sont eux difficiles à reconnaître lorsqu'ils sont tronqués à 0,2 s à l'apprentissage. Cela peut provenir du manque d'information pour les reconnaître comme nous l'avons constaté dans la partie 6.6.3.1. Ceci confirme l'hypothèse émise dans la partie 6.6.1 que 0,2 s d'événement lors de l'apprentissage est trop court pour l'apprentissage d'une représentation des événements. La meilleure performance observée pour les

6. Contrairement aux expériences précédentes où certains clips pouvaient avoir une densité $D_c = 1$ pour l'ensemble des expériences.

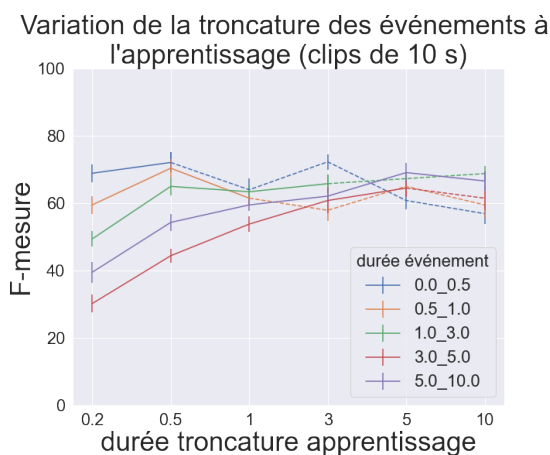


FIGURE 6.12 – F-mesure (%) du système de référence ré-appris sur le jeu de données ETAAF et évalué sur le jeu de données AAF en fonction de la durée de troncature des événements à l'apprentissage et de leur durée originelle avant troncature.

événements courts que pour les événements longs lorsque tous les événements sont tronqués à 0,5 s ($D_c \simeq 0,05$) peut montrer la difficulté à segmenter les événements courts dans de longs clips audio. En effet, lorsque nous biaisons le système à l'apprentissage en tronquant tous les événements à une durée de 0,5 s, leur reconnaissance peut devenir plus facile puisque le problème de segmentation et le nombre de fenêtres à détecter sont quasiment les mêmes pour l'ensemble des événements (nous cherchons 0,5 s d'événement sonore). Les courbes bleue et jaune (événements d'une durée ≤ 1 s) sont majoritairement décroissantes lorsque la durée maximale des événements sonores augmente. Cela indique qu'utiliser des événements plus longs dans l'apprentissage biaise probablement le système vers ces classes et rend la reconnaissance des événements courts plus difficile (segmentation, déséquilibre du nombre de fenêtre actives). Cependant, nous pouvons tout de même noter qu'il est moins pénalisant d'avoir du bruit de fond (annotations faibles) que des événements tronqués à l'apprentissage.

Impact de la valeur de D_c à l'apprentissage et à l'évaluation Pour finir, nous proposons d'analyser l'impact des annotations faibles en fonction de la densité d'événement et non plus de leur durée (D_c fixe). Notre but est de valider que les conclusions émises dans les expériences précédentes ne dépendent pas des classes d'événements cibles choisies, de leur durée ou de la durée variable du bruit de fond introduit. Nous avons vu précédemment que tronquer tous les événements à une courte durée n'est pas souhaitable, en particulier pour les événements longs. Pour éviter ce problème et avoir une densité d'événement D_c fixée pour l'ensemble des événements, nous décidons d'ajouter du bruit de fond autour de l'événement sonore en quantité proportionnelle à sa durée (la durée des clips est donc variable et proportionnelle à la durée des événements). Nous fixons la durée maximale d'un événement à 3 s puisque nous avons observé dans la Figure 6.10 que cette durée est suffisante pour obtenir des résultats semblables à ceux obtenus avec l'événement complet dans le cas des événements longs.

Le Tableau 6.8 présente les résultats obtenus sur le jeu de données DAAF avec $D_c = 1$, $D_c = 0,5$ ou $D_c = 0,1$ à l'apprentissage ou à l'évaluation. De manière générale, la performance diminue quand la densité à l'évaluation diminue. Si nous nous concentrons sur

		Évaluation		
		Idéal ($D_c = 1$)	$d_{ev} \times 2$ ($D_c = 0,5$)	$d_{ev} \times 10$ ($D_c = 0,1$)
Apprentissage	Idéal ($D_c = 1$)	66,2 $\pm 1,2$	49,5 $\pm 1,9$	33,2 $\pm 1,6$
	$d_{ev} \times 2$ ($D_c = 0,5$)	60,6 $\pm 1,2$	55,7 $\pm 1,2$	39,5 $\pm 1,1$
	$d_{ev} \times 10$ ($D_c = 0,1$)	60,1 $\pm 1,0$	61,5 $\pm 1,0$	54,8 $\pm 0,9$

TABLEAU 6.8 – F-mesure (%) du système de référence ré-appris et évalué sur le jeu de données DAAF en fonction de la durée des clips d'apprentissage et d'évaluation.

les lignes, nous observons qu'utiliser une valeur de D_c plus faible à l'évaluation qu'à l'apprentissage diminue la performance, à l'exception de la dernière ligne où le modèle est appris avec des clips de densité $D_c = 0,1$ pour lesquels une évaluation avec des clips de densité $D_c = 0,5$ ou $D_c = 1$ donne des résultats similaires. Lorsque nous analysons les colonnes, nous pouvons définir comment annoter nos données d'apprentissage si nous avons une estimation de la densité d'événement attendue à l'évaluation (grâce par exemple à une pré-segmentation des événements). Cela peut aussi revenir à définir le budget nécessaire pour annoter les données d'apprentissage en fonction du scénario visé. Si la segmentation est parfaite lors de l'évaluation (qu'elle soit manuelle ou automatique), ce qui correspond à la première colonne du tableau, le mieux est de segmenter également les données d'apprentissage. Cependant, si nous n'avons pas le budget pour segmenter les données d'apprentissage et que nous devons annoter faiblement les données, il n'y a pas de différence entre l'annotation des données avec $D_c = 0,5$ ou $D_c = 0,1$. Ceci indique que nous pouvons cibler une résolution égale à dix fois la durée des événements, ce qui peut accélérer considérablement le processus d'annotation. Pour prendre un exemple concret, si nous ciblons une classe d'événements sonores qui sont généralement d'une durée de 3 s, il n'y a pas de différence entre annoter des clips de 6 s ou de 30 s. De façon intéressante, si nous n'avons pas accès à une segmentation parfaite à l'évaluation ou si nous avons des contraintes réduites en terme de résolution temporelle de détection à l'évaluation ($D_c < 1$), il n'est pas nécessaire d'annoter les événements sonores avec des frontières temporelles précises. Il est plus intéressant d'annoter faiblement les données car la meilleure performance est obtenue avec $D_c = 0,1$ à l'apprentissage, ce qui peut réduire de façon considérable l'effort d'annotation.

6.7 Conclusion

Dans ce chapitre, nous avons présenté une étude de l'impact des annotations faibles sur des modèles d'étiquetage d'événements sonores. Nous avons défini le problème des annotations faibles, proposé une façon de créer des jeux de données qui permettent d'isoler ce problème particulier et suggéré un scénario où les événements peuvent être en présence de bruit de fond dans des proportions variables. Nous avons d'abord montré que, lorsque nous sommes en présence d'annotations faibles, la fonction d'agrégation utilisée lors de l'évaluation est cruciale. Nous proposons d'utiliser la fonction d'agrégation L_p comme alternative aux fonctions d'agrégation max ou softmax puisqu'elle est différentiable et flexible. Nous avons aussi montré que l'agrégation est plus importante à l'évaluation qu'à l'apprentissage. En ce qui concerne la segmentation, nous avons montré qu'il est plus intéressant d'avoir des clips d'apprentissage trop longs (contenant du bruit de fond) que trop courts (tronquant les événements), quel que soit le scénario d'évaluation. Enfin, nous avons indiqué des pistes sur la granularité d'annotation nécessaire à l'apprentissage en fonction du scénario applicatif visé et de sa granularité temporelle.

7 Conclusion et perspectives

Dans ce chapitre nous résumons le travail réalisé dans le cadre de cette thèse et proposons des pistes pour continuer ce travail de recherche.

7.1 Conclusion

Dans la première partie de la thèse, nous avons décrit les recherches associées à la détection d'événements sonores en environnement domestique réalisées dans le cadre de la Tâche 4 du Challenge DCASE. Dans la deuxième partie, nous nous sommes concentrés sur deux problèmes spécifiques : l'apprentissage semi-supervisé et l'utilisation des annotations faibles.

Le chapitre 3 définit le problème, le corpus [DESED](#) et les résultats officiels de la Tâche 4 du Challenge DCASE, ainsi que son organisation à laquelle j'ai participé activement durant la thèse. Étant donné que nous voulons reproduire un scénario réaliste, le problème défini est complexe et pose de nombreux défis. Le coût des annotations est probablement le problème le plus important. Nous avons un budget d'annotation limité qui ne nous permet pas d'annoter assez de données fortement pour permettre de les utiliser à l'apprentissage. Un jeu de données avec des annotations hétérogènes (sans annotation, annotations faibles et annotations fortes) est donc créé. Les données d'apprentissage comportent des données non annotées ou faiblement annotées réalistes extraites de vidéos postées sur Youtube et des données synthétiques fortement annotées. Le jeu de données ainsi que la définition du problème ont évolué de façon continue entre 2018 et 2020 pour permettre de prendre en compte les difficultés rencontrées dans la résolution de la tâche et essayer de les traiter. Cette évolution est rendue en partie possible par l'analyse systématique des systèmes soumis. Une analyse des résultats officiels et des discussions sur les systèmes sont réalisées dans ce chapitre montrant une progression constante des performances des participants.

Le chapitre 4 décrit les systèmes de référence de la Tâche 4 du Challenge DCASE qui permettent de donner un point de comparaison chaque année aux participants et de refléter l'amélioration des performances année après année. Une analyse plus détaillée est réalisée sur le système de référence proposé en 2020 afin d'identifier les parties responsables de cette amélioration de performance. Ensuite, afin d'identifier les problèmes qui tendent à être résolus ou ceux qui posent toujours des difficultés aux participants à la tâche, nous définissons des scénarios d'évaluation spécifiques pour isoler certains problèmes rencontrés en environnement réel. Ces analyses nous ont permis de montrer

l'hétérogénéité des systèmes soumis, qui sont plus ou moins adaptés à certains problèmes spécifiques comme la robustesse à la durée et à la position temporelle des événements sonores, à l'intensité de l'événement sonore par rapport au bruit de fond ou bien à la réverbération de la pièce. Lors des différentes éditions, nous avons pu constater notamment une amélioration globale de la segmentation des événements (qui reste cependant un des enjeux majeurs de la tâche) et des difficultés de détection des événements lorsque le signal est dégradé dans une pièce avec une réverbération importante.

Le chapitre 5 présente le problème d'apprentissage de représentation de façon semi-supervisée. L'apprentissage de représentation est utile pour réduire la complexité des systèmes de classification utilisés sur ces représentations. Cependant, l'apprentissage d'une représentation de façon semi-supervisée n'est pas trivial et avait été peu étudié pour l'analyse de sons ambiants. Nous avons proposé de nous baser sur une approche existante d'apprentissage par triplets et proposé une nouvelle façon de tirer des exemples positifs de façon non supervisée basée sur la proximité des exemples dans l'espace d'entrée. Nous avons comparé l'apprentissage par triplets de façon supervisée, semi-supervisée et non supervisée et nous avons montré l'avantage que présente l'utilisation d'une approche semi-supervisée. Cependant, l'utilisation de la proximité dans l'espace d'entrée ne s'avère pas aussi efficace que l'utilisation de transformations des données proposée précédemment. Nous avons comparé l'apprentissage de représentation sur notre jeu de données (< 20 k exemples) contenant uniquement nos classes d'intérêt avec une représentation apprise de façon supervisée sur le gros corpus Audioset ne contenant qu'une faible partie avec nos exemples d'intérêt (excepté la voix qui est représentée dans un exemple sur 2, les autres classes d'événements sont très peu représentées). Nous avons montré que cette représentation générale est plus discriminante sur nos exemples d'intérêt que la représentation apprise par triplets. Cependant, la complexité du modèle proposé appris par triplets (plus faible) par rapport à celle du modèle général peut être en faveur de son utilisation dans certaines applications. Une de nos hypothèses concernant la performance relativement faible de l'apprentissage par triplets est que l'apprentissage par triplets avec des annotations faibles est trop difficile.

Le chapitre 6 analyse l'impact des annotations faibles dans un cadre supervisé. Tout d'abord nous analysons cet impact sur l'apprentissage de représentation en utilisant des méthodes basées sur la distance comme l'apprentissage par triplets et par réseau prototype. Nous avons montré que les annotations faibles influent de façon négative sur l'apprentissage de représentation effectué avec ces méthodes basées sur la distance par rapport à une approche basée directement sur la classification. Nous avons ensuite analysé le modèle de classification pour permettre de comprendre l'impact des annotations faibles pour ce type de modèle et définir les scénarios pour lesquels nous devons porter une attention particulière aux annotations faibles. Dans cette analyse, nous avons conçu des jeux de données permettant d'isoler certains aspects des annotations faibles. Nous avons montré l'importance de la fonction d'agrégation utilisée pour décider de la présence ou non d'un événement au niveau du clip. Nous avons montré que la présence de bruit

autour des événements a un impact négatif important à l'évaluation lorsque les conditions d'apprentissage ne correspondent pas au scénario d'évaluation. En revanche, les annotations faibles ont un impact faible à l'apprentissage et peuvent même être bénéfiques lorsque le scénario d'évaluation contient lui aussi du bruit autour des événements. Nous avons donc proposé des scénarios d'annotation possibles en fonction des informations connues sur le scénario d'évaluation.

En conclusion, dans ce manuscrit, nous nous sommes intéressés à la reconnaissance d'événements sonores en environnement domestique. Nous avons analysé en détail deux problèmes spécifiques : l'apprentissage semi-supervisé et l'apprentissage faiblement supervisé. Ces problèmes se retrouvent souvent dans les scénarios réels. L'analyse détaillée de certains de ces problèmes a été structurante pour la définition de la Tâche 4 du Challenge DCASE. Nous avons ainsi proposé d'analyser d'autres problèmes comme l'impact de dégradations du signal sonore ou des conditions acoustiques de la pièce grâce à l'analyse des systèmes soumis à la tâche. La définition de ce problème de reconnaissance d'événements sonores par apprentissage faiblement supervisé et semi-supervisé est un problème complexe. Nous avons affiné la définition de ce problème au cours de cette thèse et traité des problèmes particuliers, mais il reste des problèmes qui n'ont toujours pas été analysés en détail et de manière isolée. Nous avons défini un problème avec de nombreuses hypothèses sur le jeu de données, la collecte et l'annotation de celui-ci ainsi que sur les scénarios d'évaluation. Nous avons montré qu'il est difficile de définir précisément le problème réel complexe. Nous avons également montré l'importance d'analyser en détail et de manière systématique les différents aspects du problème pour bien le comprendre dans sa globalité. Cette analyse reste malheureusement partielle à l'heure actuelle mais représente une première avancée vers une meilleure compréhension du problème de la reconnaissance d'événements sonores en environnement réel. C'est une étape nécessaire à la résolution du problème. Nous avons constaté que cette analyse a inspiré la proposition de nouvelles solutions permettant la résolution de certains problèmes particuliers. Cela motive à continuer l'analyse détaillée du problème.

7.2 Poursuite des travaux

Les problèmes et analyses présentées dans cette thèse peuvent donner lieu à de nombreuses pistes de poursuite des travaux. Nous discutons ici des différentes pistes théoriques puis des pistes plus pratiques.

7.2.1 Perspectives théoriques

Représentation semi-supervisée À court terme, pour l'apprentissage de représentation semi-supervisé, il serait intéressant d'étendre l'étude que nous avons réalisée dans ce manuscrit à une étude avec un modèle appris par triplets en utilisant l'espace de représentation obtenu avec le modèle appris sur Audioset pour former des triplets. L'étude pourrait aussi être étendue à l'utilisation de nouveaux modèles comme l'apprentissage

par réseau prototype qui semble obtenir de meilleurs résultats et pour lequel on pourrait proposer une utilisation semi-supervisée. L'augmentation récente des performances des modèles de classification appris de façon semi-supervisée grâce à l'approche de professeur moyen pourrait être analysée afin de déterminer s'il est possible d'apprendre une représentation explicite avec ce modèle. Ensuite, il serait aussi intéressant d'analyser l'impact de la présence de multiples événements dans les clips audio lors de l'apprentissage de représentation [Cakir et al., 2015]. Finalement, il serait intéressant d'analyser les performances des représentations pour un scénario de détection d'événements sonores.

Annotations faibles L'analyse des annotations faibles peut être intéressante à faire dans le cadre d'un scénario de détection d'événements sonores. De plus, l'analyse sur le classifieur a été faite dans ce manuscrit avec des représentations d'entrée et un modèle précis, il serait donc intéressant d'étendre l'analyse pour comprendre si l'impact des annotations faibles est le même pour différents modèles ou différentes représentations d'entrée. La prochaine étape peut être d'analyser l'impact des annotations faibles en fonction de l'intensité du bruit de fond ou de l'apparition d'événements non-cibles. Finalement, en plus de la granularité temporelle des annotations nécessaires en fonction du scénario d'évaluation, il serait intéressant d'utiliser l'apprentissage actif pour réduire encore le budget consacré à l'annotation [Cohn et al., 1994].

Analyse systématique des systèmes soumis à la Tâche 4 du Challenge DCASE Lors de l'analyse des systèmes comme nous l'avons fait dans ce manuscrit, nous avons discuté des métriques utilisées, qui influencent l'analyse et les comparaisons entre systèmes. Ces analyses peuvent être faites en utilisant des scores permettant de refléter des scénarios d'évaluation différents, comme l'utilisation du PSDS avec des paramètres permettant de refléter des scénarios applicatifs réalistes. De plus, les systèmes soumis dépendent en réalité d'un nombre important de facteurs, et il est parfois difficile de déterminer si l'approche utilisée par un participant est meilleure que celle d'un autre participant en utilisant un seul modèle appris pour chaque système et sans prêter attention à ces différents facteurs de variation des performances (même en utilisant différentes métriques). Par exemple, il peut y avoir une variation importante des modèles soumis en fonction du tirage des données utilisées, de leur initialisation ou des hyper-paramètres utilisés dont nous ne mesurons pas l'impact [Bouthillier et al., 2021]. La définition d'une campagne d'évaluation rigoureuse permettant de prendre en compte certains de ces aspects est complexe et peut faire parties des pistes de travail à plus long terme.

Autres problèmes Il serait aussi intéressant de définir un scénario pour comprendre l'impact du déséquilibre des données à l'apprentissage et à l'évaluation. Ensuite, il serait intéressant de réaliser une analyse de l'impact de la polyphonie dans les clips audio. Ces analyses peuvent commencer par être faites grâce à l'isolation du problème en utilisant un jeu de données synthétiques qui serait possible avec DESED.

7.2.2 Perspectives pratiques

Amélioration du jeu de données DESED En ce qui concerne l'évolution du jeu de données DESED, à court terme il est possible d'utiliser les vidéos de Flickr pour augmenter le jeu de données avec des données non annotées plus diverses. De plus, la génération des données synthétiques d'apprentissage peut utiliser des méthodes de co-occurrence des événements plus avancées basées sur la présence des événements cibles et non-cibles dans les enregistrements réels pour permettre d'utiliser des données synthétiques plus réalistes.

Problèmes en environnement réel De manière générale, de multiples problèmes apparaissant en environnement réel pourraient être isolés et analysés comme l'impact de l'équilibre des classes d'événements cibles (en terme de nombre et de durée), de la polyphonie [Cakir et al., 2015] ou bien de la pièce dans laquelle les données sont collectées. Pour ce dernier point, chaque maison ou pièce étant différente, il serait intéressant d'analyser si nous pouvons adapter un système à une pièce ou une maison précise puisque dans un scénario réel, le dispositif utilisé sera sûrement dans une maison et bougera peu [Vafeiadis et al., 2020]. Finalement, dans cette thèse nous avons utilisé des données issues de plateformes web comme Youtube ou Vimeo, qui sont généralement enregistrées dans un but précis (montrer le miaulement de son chat ou présenter une recette de cuisine par exemple). Cela se traduit par le fait que les clips audio contiennent souvent de multiples événements sonores et le microphone peut se retrouver proche de notre événement d'intérêt. Il serait intéressant d'analyser la performance des systèmes sur des données enregistrées dans une maison de façon non contrôlée, de sorte que la densité d'événement est probablement plus faible et le microphone pas forcément à proximité de l'événement d'intérêt. Cela permettrait de déterminer un ordre de grandeur de performance des systèmes en environnement réel.

Applications Dans ce manuscrit, nous nous sommes intéressés à l'étiquetage et à la détection d'événements sonores en environnement domestique pour un nombre limité de classes qui ont été choisies en partie en fonction de la disponibilité des données sur le web. Il serait intéressant de déterminer un scénario réel d'assistance aux personnes en perte d'autonomie et les classes spécifiques nécessaires afin de déterminer quelles sont les classes disponibles sur le web et celles qui ne le sont pas.

Bibliographie

- Adavanne, S., Fayek, H., and Tourbabin, V. (2019a). Sound event classification and detection with weakly labeled data. In *2019 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 15–19.
- Adavanne, S., Politis, A., and Virtanen, T. (2019b). A multi-room reverberant dataset for sound event localization and detection. In *2019 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 10–14.
- Agrawal, D. M., Sailor, H. B., Soni, M. H., and Patil, H. A. (2017). Novel TEO-based Gammatone features for environmental sound classification. In *25th European Signal Processing Conference (EUSIPCO)*, pages 1809–1813.
- Ardouin, J., Charpentier, L., Lagrange, M., Fortin, N., Ecoti re, D., and Guillaume, G. (2018). An innovative low cost sensor for urban sound monitoring. In *47th International Congress and Exposition on Noise Control Engineering (Inter Noise)*, pages 2226–2237.
- Arora, V., Sun, M., and Wang, C. (2019). Deep embeddings for rare audio event detection with imbalanced data. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3297–3301.
- Avdeeva, A. and Agafonov, I. (2018). Sound event detection using weakly labeled dataset with convolutional recurrent neural network. Technical report, DCASE Challenge.
- Baillie, M. and Jose, J. M. (2003). Audio-based event detection for sports video. In *2nd International Conference on Image and Video Retrieval (CIVR)*, pages 300–309.
- Bello, J. P., Silva, C., Nov, O., DuBois, R. L., Arora, A., Salamon, J., Mydlarz, C., and Doraiswamy, H. (2019). SONYC : A system for the monitoring, analysis and mitigation of urban noise pollution. *Communications of the ACM*, 62 :68–77.
- Benetos, E., Lafay, G., Lagrange, M., and Plumbley, M. D. (2016). Detection of overlapping acoustic events using a temporally-constrained probabilistic model. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6450–6454.
- Bilen, C., Ferroni, G., Tuveri, F., Azcarreta, J., and Krstulovi c, S. (2020). A framework for the robust evaluation of sound event detection. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 61–65.

- Bisot, V., Essid, S., and Richard, G. (2017a). Overlapping sound event detection with supervised nonnegative matrix factorization. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35.
- Bisot, V., Serizel, R., Essid, S., and Richard, G. (2017b). Feature learning with matrix factorization applied to acoustic scene classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6) :1216–1229.
- Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Mohammadi Sepahvand, N., Raff, E., Madan, K., Voleti, V., et al. (2021). Accounting for variance in machine learning benchmarks. *Proceedings of Machine Learning and Systems*, 3.
- Bregman, A. S. (1990). *Auditory Scene Analysis : The Perceptual Organization of Sound*. MIT Press.
- Bromley, J., Guyon, I., LeCun, Y., Säcker, E., and Shah, R. (1993). Signature verification using a “siamese” time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4) :8.
- Brown, G. J. and Cooke, M. (1994). Computational auditory scene analysis. *Computer Speech & Language*, 8(4) :297–336.
- Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. (2003). Highlight sound effects detection in audio stream. In *2003 International Conference on Multimedia and Expo (ICME)*, volume 3, page 37.
- Cakir, E., Heittola, T., Huttunen, H., and Virtanen, T. (2015). Multi-label vs. combined single-label sound event detection with deep neural networks. In *23rd European signal processing conference (EUSIPCO)*, pages 2551–2555.
- Cakir, E., Parascandolo, G., Heittola, T., Huttunen, H., and Virtanen, T. (2017). Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6) :1291–1303.
- Cances, L., Pellegrini, T., and Guyot, P. (2018). Sound event detection from weak annotations : Weighted GRU versus multi-instance learning. Technical report, DCASE Challenge.
- Cances, L., Pellegrini, T., and Guyot, P. (2019). Multi task learning and post processing optimization for sound event detection. Technical report, DCASE Challenge.
- Carpentier, L., Vranken, E., Berckmans, D., Paeshuyse, J., and Norton, T. (2019). Development of sound-based poultry health monitoring tool for automated sneeze detection. *Computers and Electronics in Agriculture*, 162 :573–581.

- Cartwright, M., Cramer, J., Mendez Mendez, A. E., Wang, Y., Wu, H.-H., Lostanlen, V., Fuentes, M., Dove, G., Mydlarz, C., Salamon, J., Nov, O., and Bello, J. P. (2020). SONYC-UST-V2 : An urban sound tagging dataset with spatiotemporal context. In *2020 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 16–20.
- Cerutti, G., Prasad, R., Brutti, A., and Farella, E. (2019). Neural network distillation on IoT platforms for sound event detection. In *Interspeech*, pages 3609–3613.
- Chan, T. K. and Chin, C. S. (2020). A comprehensive review of polyphonic sound event detection. *IEEE Access*, 8 :103339–103373.
- Chan, T. K., Chin, C. S., and Li, Y. (2019). Non-negative matrix factorization-convolution neural network (NMF-CNN) for sound event detection. Technical report, DCASE Challenge.
- Chen, H., Xie, W., Vedaldi, A., and Zisserman, A. (2020). VGGSound : A large-scale audio-visual dataset. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725.
- Chen, J., Kam, A. H., Zhang, J., Liu, N., and Shue, L. (2005). Bathroom activity monitoring based on sound. In *3rd International Conference on Pervasive Computing (PERVASIVE)*, pages 47–61.
- Choe, H., Karisen, R., Gerhart, G., and Meitzler, T. (1996). Wavelet-based ground vehicle recognition using acoustic signals. In *SPIE Aerospace/Defense Sensing and Controls*.
- Choi, K., Fazekas, G., Cho, K., and Sandler, M. (2018). The effects of noisy labels on deep convolutional neural networks for music tagging. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2) :139–149.
- Chu, S., Narayanan, S., and Kuo, C.-J. (2009). Environmental sound recognition with time–frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6) :1142–1158.
- Chung, Y., Oh, S., Lee, J., Park, D., Chang, H.-H., and Kim, S. (2013). Automatic detection and recognition of pig wasting diseases using sound data in audio surveillance systems. *Sensors*, 13(10) :12929–12942.
- Clavel, C., Ehrette, T., and Richard, G. (2005). Events detection for an audio-based surveillance system. In *2005 IEEE International Conference on Multimedia and Expo*, pages 1306–1309.
- Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2) :201–221.
- Cornell, S., Pepe, G., Principi, E., Pariente, M., Olvera, M., Gabrielli, L., and Squartini, S. (2020). The UNIVPM-Inria systems for the DCASE 2020 Task 4. Technical report, DCASE Challenge.

- Cramer, J., Wu, H.-H., Salamon, J., and Bello, J. P. (2019). Look, listen, and learn more : Design choices for deep audio embeddings. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3852–3856.
- Dai, Z., Yang, Z., Yang, F., Cohen, W. W., and Salakhutdinov, R. R. (2017). Good semi-supervised learning that requires a bad GAN. In *31st Conference on Neural Information Processing Systems (NeurIPS)*, pages 6513–6523.
- Darna-Sequeiros, J. and Toledano, D. T. (2018). Audio event detection on Google’s Audio Set database : Preliminary results using different types of DNNs. In *IberSPEECH*, pages 64–67.
- Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4) :357–366.
- de Benito-Gorrón, D., Ramos, D., and Toledano, D. T. (2021). An analysis of sound event detection under acoustic degradation using multi-resolution systems. In *IberSPEECH*, pages 36–40.
- Debes, C., Merentitis, A., Sukhanov, S., Niessen, M., Frangiadakis, N., and Bauer, A. (2016). Monitoring activities of daily living in smart homes : Understanding human behavior. *IEEE Signal Processing Magazine*, 33(2) :81–94.
- Dekkers, G., Lauwereins, S., Thoen, B., Adhana, M. W., Brouckxon, H., Van den Bergh, B., van Waterschoot, T., Vanrumste, B., Verhelst, M., and Karsmakers, P. (2017). The SINS database for detection of daily activities in a home environment using an acoustic sensor network. In *2017 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 32–36.
- Delphin-Poulat, L., Nicol, R., Plapous, C., and Peron, K. (2020). Comparative assessment of data augmentation for semi-supervised polyphonic sound event detection. In *27th Conference of Open Innovations Association (FRUCT)*, pages 46–53.
- Delphin-Poulat, L. and Plapous, C. (2019). Mean teacher with data augmentation for DCASE 2019 Task 4. Technical report, DCASE Challenge.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT : Pre-training of deep bidirectional Transformers for language understanding. In *2019 Conference of the North American Chapter of the ACL : Human Language Technologies (NAACL-HLT)*.
- DiCiccio, T. J. and Efron, B. (1996). Bootstrap confidence intervals. *Statistical Science*, 11 :189–228.
- Dinkel, H., Qiand, Y., and Yu, K. (2018). A hybrid ASR model approach on weakly labeled scene classification. Technical report, DCASE Challenge.

- Dinkel, H., Wu, M., and Yu, K. (2021). Towards duration robust weakly supervised sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29 :887–900.
- Dorfer, M. and Widmer, G. (2018). Training general purpose audio tagging networks with noisy labels and iterative self-verification. Technical report, DCASE Challenge.
- Drossos, K., Adavanne, S., and Virtanen, T. (2017). Automated audio captioning with recurrent neural networks. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 374–378.
- Ducoffe, M. and Precioso, F. (2018). Adversarial active learning for deep networks : a margin based approach. *arXiv preprint arXiv :1802.09841*.
- Ebbers, J. and Haeb-Umbach, R. (2020). Convolutional recurrent neural networks for weakly labeled semi-supervised sound event detection in domestic environments. Technical report, DCASE Challenge.
- Ellis, D. P. W. (1996). *Prediction-driven computational auditory scene analysis*. PhD thesis, Columbia University.
- Ellis, D. P. W. (2001). Detecting alarm sounds. In *Recognition and Organization of Real-World Sounds : Workshop on Consistent and Reliable Acoustic Cues*, pages 59–62.
- Eronen, A. J., Peltonen, V. T., Tuomi, J. T., Klapuri, A. P., Fagerlund, S., Sorsa, T., Lorho, G., and Huopaniemi, J. (2006). Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1) :321–329.
- Favory, X., Fonseca, E., Font, F., and Serra, X. (2018). Facilitating the manual annotation of sounds when using large taxonomies. In *23rd Conference of Open Innovations Association (FRUCT)*, pages 447–451.
- Ferrari, M., Glotin, H., Marxer, R., and Asch, M. (2020). DOCC10 : Open access dataset of marine mammal transient studies and end-to-end CNN classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Ferroni, G., Turpault, N., Azcarreta, J., Tuveri, F., Serizel, R., Bilen, C., and Krstulović, S. (2020). Improving sound event detection metrics : Insights from DCASE 2020. *arXiv preprint arXiv :2010.13648*.
- Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., and Vento, M. (2015). Reliable detection of audio events in highly noisy environments. *Pattern Recognition Letters*, 65 :22–28.
- Fonseca, E., Favory, X., Pons, J., Font, F., and Serra, X. (2020). FSD50K : an open dataset of human-labeled sound events. *arXiv preprint arXiv :2010.00475*.

- Fonseca, E., Plakal, M., Ellis, D. P. W., Font, F., Favory, X., and Serra, X. (2019a). Learning sound event classifiers from Web audio with noisy labels. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25.
- Fonseca, E., Plakal, M., Font, F., Ellis, D. P. W., and Serra, X. (2019b). Audio tagging with noisy labels and minimal supervision. In *2019 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 69–73.
- Fonseca, E., Pons, J., Favory, X., Font, F., Bogdanov, D., Ferraro, A., Oramas, S., Porter, A., and Serra, X. (2017). Freesound datasets : a platform for the creation of open audio datasets. In *18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 486–493.
- Font, F., Roma, G., and Serra, X. (2013). Freesound technical demo. In *ACM International Conference on Multimedia (MM)*, pages 411–412.
- Foster, P., Sigtia, S., Krstulovic, S., Barker, J., and Plumbley, M. D. (2015). CHiME-home : A dataset for sound source recognition in a domestic environment. In *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5.
- Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep Bayesian active learning with image data. In *International Conference on Machine Learning (ICML)*, pages 1183–1192.
- Geiger, J. T., Schuller, B., and Rigoll, G. (2013). Large-scale audio feature extraction and SVM for acoustic scene classification. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–4.
- Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio Set : An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780.
- Gemmeke, J. F., Vuegen, L., Karsmakers, P., Vanrumste, B., and hamme, H. V. (2013). An exemplar-based NMF approach to audio event detection. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–4.
- Gharib, S., Drossos, K., Fagerlund, E., and Virtanen, T. (2019). VOICE : A sound event detection dataset for generalizable domain adaptation. *arXiv preprint arXiv :1911.07098*.
- Goldhor, R. S. (1993). Recognition of environmental sounds. In *1993 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 149–152.

- Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M. R., and Huang, D. (2018). CurriculumNet : Weakly supervised learning from large-scale Web images. In *15th European Conference on Computer Vision (ECCV)*, pages 139–154.
- Hao, J., Hou, Z., and Peng, W. (2020). Cross-domain sound event detection : from synthesized audio to real audio. Technical report, DCASE Challenge.
- Harb, R. and Pernkopf, F. (2018). Sound event detection using weakly labeled semi-supervised data with GCRNNs, VAT and self-adaptive label refinement. Technical report, DCASE Challenge.
- Heittola, T. and Klapuri, A. (2008). TUT acoustic event detection system 2007. In *International Evaluation Workshops CLEAR 2007 and RT 2007*, pages 364–370.
- Heittola, T., Mesaros, A., Eronen, A., and Virtanen, T. (2013). Context-dependent sound event detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2013(1) :1.
- Heittola, T., Mesaros, A., Virtanen, T., and Eronen, A. (2011). Sound event detection in multisource environments using source separation. In *1st International Workshop on Machine Listening in Multisource Environments (CHiME)*, pages 36–40.
- Henaff, M., Jarrett, K., Kavukcuoglu, K., and LeCun, Y. (2011). Unsupervised learning of sparse features for scalable audio classification. In *12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 681–686.
- Henze, D., Gorishti, K., Bruegge, B., and Simen, J. (2019). AudioForesight : A process model for audio predictive maintenance in industrial environments. In *18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 352–357.
- Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R., and Wilson, K. (2017). CNN architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8) :1735–1780.
- Hou, Y. and Li, S. (2018). Semi-supervised sound event detection with convolutional recurrent neural network using weakly labelled data. Technical report, DCASE Challenge.
- Huang, Q., Jansen, A., Zhang, L., Ellis, D. P. W., Saurous, R. A., and Anderson, J. (2020a). Large-scale weakly-supervised content embeddings for music recommendation and tagging. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8364–8368.

- Huang, Y., Lin, L., Ma, S., Wang, X., Liu, H., Qian, Y., Liu, M., and Ouch, K. (2020b). Guided multi-branch learning systems for DCASE 2020 Task 4. Technical report, DCASE Challenge.
- Hyeonggi, M., Joon, B., Bum-Jun, K., Shin-Hyuk, J., Youngho, J., Young-Cheol, P., and Sung-Wook, P. (2018). End-to-end CRNN architectures for weakly supervised sound event detection. Technical report, DCASE Challenge.
- Härmä, A., McKinney, M., and Skowronek, J. (2005). Automatic surveillance of the acoustic activity in our living environment. In *2005 IEEE International Conference on Multimedia and Expo (ICME)*, pages 634–637.
- Jansen, A., Plakal, M., Pandya, R., Ellis, D. P. W., Hershey, S., Liu, J., Moore, R. C., and Saurous, R. A. (2017). Unsupervised learning of semantic audio representations. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 126–130.
- Jati, A., Kumar, N., Chen, R., and Georgiou, P. (2019). Hierarchy-aware loss function on a tree structured label space for audio event detection. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6–10.
- JiaKai, L. (2018). Mean teacher convolution system for DCASE 2018 Task 4. Technical report, DCASE Challenge.
- Kavalerov, I., Wisdom, S., Erdogan, H., Patton, B., Wilson, K., Le Roux, J., and Hershey, J. R. (2019). Universal sound separation. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 175–179.
- Kendall, A. and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In *31st Conference on Neural Information Processing Systems (NIPS)*, pages 5580–5590.
- Kim, B. and Ghaffarzadegan, S. (2019). Self-supervised attention model for weakly labeled audio event classification. In *27th European Signal Processing Conference (EU-SIPCO)*, pages 1–5.
- Kim, J., Min, K., Jung, M., and Chi, S. (2020). Occupant behavior monitoring and emergency event detection in single-person households using deep learning-based sound recognition. *Building and Environment*, 181 :107092.
- Kim, T., Lee, J., and Nam, J. (2019). Comparison and analysis of SampleCNN architectures for audio classification. *IEEE Journal of Selected Topics in Signal Processing*, 13(2) :285–297.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations (ICLR)*.

- Kiyokawa, Y., Mishima, S., Toizumi, T., Sagi, K., Kondo, R., and Nomura, T. (2019). Sound event detection with ResNet and self-mask module for DCASE 2019 Task 4. Technical report, DCASE Challenge.
- Koh, C.-Y., Chen, Y.-S., Li, S.-E., Liu, Y.-W., Chien, J.-T., and Bai, M. R. (2020). Sound event detection by consistency training and pseudo-labeling with feature-pyramid convolutional recurrent neural networks. Technical report, DCASE Challenge.
- Koizumi, Y., Kawaguchi, Y., Imoto, K., Nakamura, T., Nikaido, Y., Tanabe, R., Purohit, H., Suefusa, K., Endo, T., Yasuda, M., and Harada, N. (2020). Description and discussion on DCASE2020 challenge Task2 : Unsupervised anomalous sound detection for machine condition monitoring. In *2020 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 81–85.
- Koizumi, Y., Saito, S., Uematsu, H., Harada, N., and Imoto, K. (2019). ToyADMOS : A dataset of miniature-machine operating sounds for anomalous sound detection. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 308–312.
- Komatsu, T., Toizumi, T., Kondo, R., and Senda, Y. (2016). Acoustic event detection method using semi-supervised non-negative matrix factorization with mixtures of local dictionaries. In *2018 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 45–49.
- Kong, Q., Turab, I., Yong, X., Wang, W., and Plumbley, M. D. (2018). DCASE 2018 Challenge baseline with convolutional neural networks. Technical report, DCASE Challenge.
- Kong, Q., Xu, Y., Wang, W., and Plumbley, M. D. (2020). Sound event detection of weakly labelled data with CNN-Transformer and automatic threshold optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28 :2450–2460.
- Kong, Q., Yu, C., Xu, Y., Iqbal, T., Wang, W., and Plumbley, M. D. (2019). Weakly labelled AudioSet tagging with attention neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(11) :1791–1802.
- Kothinti, S., Imoto, K., Chakrabarty, D., Sell, G., Watanabe, S., and Elhilali, M. (2018). Joint acoustic and class inference for weakly supervised sound event detection. Technical report, DCASE Challenge.
- Koutini, K., Eghbal-zadeh, H., and Widmer, G. (2018). Iterative knowledge distillation in R-CNNs for weakly-labeled semi-supervised sound event detection. Technical report, DCASE Challenge.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6) :84–90.

- Kroos, C., Bones, O., Cao, Y., Harris, L., Jackson, P. J. B., Davies, W. J., Wang, W., Cox, T. J., and Plumbley, M. D. (2019). Generalisation in environmental sound classification : The ‘Making Sense of Sounds’ data set and challenge. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8082–8086.
- Kumar, A. and Raj, B. (2016). Audio event detection using weakly labeled data. In *24th ACM International Conference on Multimedia (MM)*, pages 1038–1047.
- Kumar, A., Shah, A., Hauptmann, A., and Raj, B. (2019). Learning sound events from Webly labeled data. In *28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2772–2778.
- Laine, S. and Aila, T. (2016). Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv :1610.02242*.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324.
- Lemaitre, G. and Heller, L. (2013). Evidence for a basic level in a taxonomy of everyday action sounds. *Experimental Brain Research*, 226 :253–264.
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017a). Hyperband : A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(1) :6765–6816.
- Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. (2017b). Learning from noisy labels with distillation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1910–1918.
- Lim, W., Suh, S., and Jeong, Y. (2018). Weakly labeled semi-supervised sound event detection using CRNN with inception module. Technical report, DCASE Challenge.
- Lim, W., Suh, S., Park, S., and Jeong, Y. (2019). Sound event detection in domestic environments using ensemble of convolutional recurrent neural networks. Technical report, DCASE Challenge.
- Lin, L. and Wang, X. (2019). Guided learning convolution system for DCASE 2019 Task 4. Technical report, DCASE Challenge.
- Little, D. and Pardo, B. (2008). Learning musical instruments from mixtures of audio with weak labels. In *9th International Conference on Music Information Retrieval (ISMIR)*, pages 127–132.
- Liu, Y., Chen, H., and Zhang, P. (2020). Power pooling operators and confidence learning for semi-supervised sound event detection. *arXiv preprint arXiv :2005.11459*.
- Liu, Y., Yan, J., Song, Y., and Du, J. (2018). USTC-NELSLIP system for DCASE 2018 Challenge Task 4. Technical report, DCASE Challenge.

- Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S., and Bello, J. P. (2018). Birdvox-Full-Night : A dataset and benchmark for avian flight call detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 266–270.
- Lu, Z., Fu, Z., Xiang, T., Han, P., Wang, L., and Gao, X. (2017a). Learning from weak and noisy labels for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3) :486–500.
- Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017b). The expressive power of neural networks : A view from the width. In *31st International Conference on Neural Information Processing (NeurIPS)*, volume 30, pages 6231–6239.
- Mauch, M. and Ewert, S. (2013). The Audio Degradation Toolbox and its application to robustness evaluation. In *14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 83–88.
- McFee, B., Salamon, J., and Bello, J. P. (2018). Adaptive pooling operators for weakly labeled sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11) :2180–2193.
- McLoughlin, I., Zhang, H., Xie, Z., Song, Y., and Xiao, W. (2015). Robust sound event classification using deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3) :540–552.
- Mesaros, A., Diment, A., Elizalde, B., Heittola, T., Vincent, E., Raj, B., and Virtanen, T. (2019). Sound event detection in the DCASE 2017 Challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(6) :992–1006.
- Mesaros, A., Heittola, T., Benetos, E., Foster, P., Lagrange, M., Virtanen, T., and Plumbley, M. D. (2018a). Detection and classification of acoustic scenes and events : Outcome of the DCASE 2016 Challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(2) :379–393.
- Mesaros, A., Heittola, T., Diment, A., Elizalde, B., Shah, A., Vincent, E., Raj, B., and Virtanen, T. (2017a). DCASE 2017 Challenge setup : Tasks, datasets and baseline system. In *2017 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 85–92.
- Mesaros, A., Heittola, T., and Virtanen, T. (2016a). Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6) :162.
- Mesaros, A., Heittola, T., and Virtanen, T. (2016b). TUT database for acoustic scene classification and sound event detection. In *24th European Signal Processing Conference (EUSIPCO)*, pages 1128–1132.

- Mesaros, A., Heittola, T., and Virtanen, T. (2017b). Assessment of human and machine performance in acoustic scene classification : DCASE 2016 case study. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 319–323.
- Mesaros, A., Heittola, T., and Virtanen, T. (2018b). Acoustic scene classification : An overview of DCASE 2017 Challenge entries. In *16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 411–415.
- Mesaros, A., Heittola, T., and Virtanen, T. (2018c). A multi-device dataset for urban acoustic scene classification. In *2018 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 9–13.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *26th International Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119.
- Miyazaki, K., Komatsu, T., Hayashi, T., Watanabe, S., Toda, T., and Takeda, K. (2020). Convolution-augmented Transformer for semi-supervised sound event detection. Technical report, DCASE Challenge.
- Moreno, P. J. and Agarwal, S. (2003). An experimental study of semi-supervised EM algorithms in audio classification and speaker identification. In *Workshop on the Continuum from Labeled to Unlabeled Data*, page 10.
- Munich, M. E. (2004). Bayesian subspace methods for acoustic signature recognition of vehicles. In *12th European Signal Processing Conference (EUSIPCO)*, pages 2107–2110.
- Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. In *26th International Conference on Neural Information Processing Systems (NIPS)*, volume 26, pages 1196–1204.
- Navarro, J., Vidaña-Vila, E., Alsina-Pagès, R., and Hervás, M. (2018). Real-time distributed architecture for remote acoustic elderly monitoring in residential-scale ambient assisted living scenarios. *Sensors*, 18 :2492.
- Ono, Y., Onishi, Y., Koshinaka, T., Takata, S., and Hoshuyama, O. (2013). Anomaly detection of motors with feature emphasis using only normal sounds. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2800–2804.
- Pankajakshan, A., Bear, H. L., and Benetos, E. (2019). Polyphonic sound event and sound activity detection : A multi-task approach. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 323–327.

- Peng, Y.-T., Lin, C.-Y., Sun, M.-T., and Tsai, K.-C. (2009). Healthcare audio event classification using hidden Markov models and hierarchical hidden Markov models. In *2009 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1218–1221.
- Perez-Castanos, S., Naranjo-Alcazar, J., Zuccarello, P., and Cobos, M. (2020). Anomalous sound detection using unsupervised and semi-supervised autoencoders and gammatone audio representation. In *2020 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 145–149.
- Piczak, K. J. (2015). ESC : Dataset for environmental sound classification. In *23rd ACM International Conference on Multimedia (MM)*, pages 1015–1018.
- Politis, A., Adavanne, S., and Virtanen, T. (2020). A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection. In *2020 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 165–169.
- Pons, J., Serrà, J., and Serra, X. (2019). Training neural audio classifiers with few data. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16–20.
- Purohit, H., Tanabe, R., Ichige, T., Endo, T., Nikaido, Y., Suefusa, K., and Kawaguchi, Y. (2019). MIMII dataset : Sound dataset for malfunctioning industrial machine investigation and inspection. In *2019 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 209–213.
- Qian, K., Ren, Z., Pandit, V., Yang, Z., Zhang, Z., and Schuller, B. (2017). Wavelets revisited for the classification of acoustic scenes. In *2017 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 108–112.
- Radhakrishnan, R., Divakaran, A., and Smaragdis, A. (2005). Audio analysis for surveillance applications. In *2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 158–161.
- Raj, R., Waldekar, S., and Saha, G. (2018). Large-scale weakly labelled semi-supervised CQT based sound event detection in domestic environments. Technical report, DCASE Challenge.
- Ruge, L., Altakrouri, B., and Schrader, A. (2013). SoundOfTheCity — continuous noise monitoring for a healthy city. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 670–675.
- Saeed, A., Grangier, D., and Zeghidour, N. (2020). Contrastive learning of general-purpose audio representations. *arXiv preprint arXiv :2010.10915*.

- Salamon, J. and Bello, J. P. (2015a). Feature learning with deep scattering for urban sound analysis. In *23rd European Signal Processing Conference (EUSIPCO)*, pages 724–728.
- Salamon, J. and Bello, J. P. (2015b). Unsupervised feature learning for urban sound classification. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 171–175.
- Salamon, J., Jacoby, C., and Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *22nd ACM International Conference on Multimedia (MM)*, pages 1041–1044.
- Salamon, J., MacConnell, D., Cartwright, M., Li, P., and Bello, J. P. (2017). Scaper : A library for soundscape synthesis and augmentation. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 344–348.
- Sampan, S. (1998). *Neural Fuzzy Techniques in Vehicle Acoustic Signal Classification*. PhD thesis, Virginia Tech.
- Schroff, F., Kalenichenko, D., and Philbin, J. (2015). FaceNet : A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.
- Serizel, R., Bisot, V., Essid, S., and Richard, G. (2017). Acoustic features for environmental sound analysis. In *Computational Analysis of Sound Scenes and Events*, pages 71–101. Springer.
- Serizel, R. and Turpault, N. (2019). Sound event detection from partially annotated data : Trends and challenges. In *IcETTRAN conference*.
- Serizel, R., Turpault, N., Eghbal-Zadeh, H., and Parag Shah, A. (2018). Large-scale weakly labeled semi-supervised sound event detection in domestic environments. In *2018 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*.
- Serizel, R., Turpault, N., Shah, A., and Salamon, J. (2020). Sound event detection in synthetic domestic environments. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 86–90.
- Shah, A., Kumar, A., Hauptmann, A. G., and Raj, B. (2018). A closer look at weak label learning for audio events. *arXiv preprint arXiv :1804.09288*.
- Shi, B., Sun, M., Kao, C., Rozgic, V., Matsoukas, S., and Wang, C. (2019). Semi-supervised acoustic event detection based on tri-training. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 750–754.
- Shi, Z. (2019). HodgePodge : Sound event detection based on ensemble of semi-supervised learning methods. Technical report, DCASE Challenge.

- Shi, Z., Liu, L., and Liu, R. (2020). Hodge and Podge : Hybrid supervised sound event detection with multi-hot MixMatch and composition consistence training. In *28th European Signal Processing Conference (EUSIPCO)*, pages 1–5.
- Singh, J. and Joshi, R. (2019). Background sound classification in speech audio segments. In *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 1–6.
- Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical networks for few-shot learning. In *31st International Conference on Neural Information Processing Systems (NIPS)*, pages 4080–4090.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15 :1929–1958.
- Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J., Mostefa, D., and Soundararajan, P. (2006). The CLEAR 2006 Evaluation. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, page 46.
- Stiefelhagen, R., Bernardin, K., Bowers, R., Rose, R., Michel, M., and Garofolo, J. (2007). The CLEAR 2007 evaluation. In *International Evaluation Workshops CLEAR 2007 and RT 2007*, pages 3–34.
- Stowell, D., Giannoulis, D., Benetos, E., Lagrange, M., and Plumbley, M. D. (2015). Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10) :1733–1746.
- Stowell, D., Wood, M. D., Pamuła, H., Stylianou, Y., and Glotin, H. (2019). Automatic acoustic detection of birds through deep learning : The first Bird Audio Detection challenge. *Methods in Ecology and Evolution*, 10(3) :368–380.
- Su, T.-W., Liu, J.-Y., and Yang, Y.-H. (2017). Weakly-supervised audio event detection using event-specific Gaussian filters and fully convolutional networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 791–795.
- Subramanian, V., Benetos, E., and Sandler, M. B. (2019). Robustness of adversarial attacks in sound event classification. In *2019 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 239–243.
- Sudo, Y., Itoyama, K., Nishida, K., and Nakadai, K. (2020). Sound event aware environmental sound segmentation with Mask U-Net. *Advanced Robotics*, 34(20) :1280–1290.
- Sundaram, S. and Narayanan, S. (2006). Vector-based representation and clustering of audio using onomatopoeia words. In *AAAI 2006 Fall Symposium*.

- Säger, S., Borth, D., Elizalde, B., Schulze, C., Raj, B., Lane, I., and Dengel, A. (2016). AudioSentibank : Large-scale semantic ontology of acoustic concepts for audio content analysis. *arXiv preprint arXiv :1607.03766*.
- Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models : Weight-averaged consistency targets improve semi-supervised deep learning results. In *31st Conference on Neural Information Processing Systems (NIPS)*, pages 1195–1204.
- Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. (2016). YFCC100M : the new data in multimedia research. *Communications of the ACM*, 59(2) :64–73.
- Thulasidasan, S. and Bilmes, J. (2017). Acoustic classification using semi-supervised deep neural networks and stochastic entropy-regularization over nearest-neighbor graphs. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2731–2735.
- Tian, Y., Shi, J., Li, B., Duan, Z., and Xu, C. (2018). Audio-visual event localization in unconstrained videos. In *15th European Conference on Computer Vision (ECCV)*, pages 252–268.
- Turpault, N., Serizel, R., and Vincent, E. (2020a). Limitations of weak labels for embedding and tagging. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135.
- Turpault, N., Wisdom, S., Erdogan, H., Hershey, J., Serizel, R., Fonseca, E., Seetharaman, P., and Salamon, J. (2020b). Improving sound event detection in domestic environments using sound separation. In *2020 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 205–209.
- Tzinis, E., Wisdom, S., Hershey, J. R., Jansen, A., and Ellis, D. P. W. (2020). Improving universal sound separation using sound classification. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 96–100.
- Vafeiadis, A., Votis, K., Giakoumis, D., Tzovaras, D., Chen, L., and Hamzaoui, R. (2020). Audio content analysis for unobtrusive event detection in smart homes. *Engineering Applications of Artificial Intelligence*, 89 :103226.
- Valero, X. and Alias, F. (2012). Gammatone cepstral coefficients : Biologically inspired features for non-speech audio classification. *IEEE Transactions on Multimedia*, 14(6) :1684–1689.
- van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet : A generative model for raw audio. *arXiv preprint arXiv :1609.03499*.

- Ventura, R., Mallet, V., and Issarny, V. (2018). Assimilation of mobile phone measurements for noise mapping of a neighborhood. *Journal of the Acoustical Society of America*, 144(3) :1279–1292.
- Virtanen, T., Plumbley, M., and Ellis, D. (2017). *Computational Analysis of Sound Scenes and Events*. Springer.
- Wang, D., Xu, K., Zhu, B., Zhang, L., Peng, Y., and Wang, H. (2018). A CRNN-based system with mixup technique for large-scale weakly labeled sound event detection. Technical report, DCASE Challenge.
- Wang, H., Chong, D., Huang, D., and Zou, Y. (2019a). What affects the performance of convolutional neural networks for audio event classification. In *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 140–146.
- Wang, J. and Li, S. (2018). Self-attention mechanism based system for DCASE2018 Challenge Task1 and Task4. Technical report, DCASE Challenge.
- Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1386–1393.
- Wang, Y., Li, J., and Metze, F. (2019b). A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35.
- Weston, J., Ratle, F., Mobahi, H., and Collobert, R. (2018). Deep learning via semi-supervised embedding. In *25th International Conference on Machine Learning (ICML)*, pages 1168–1175.
- Wisdom, S., Erdogan, H., Ellis, D., Serizel, R., Turpault, N., Fonseca, E., Salamon, J., Seetharaman, P., and Hershey, J. (2020). What’s all the FUSS about Free Universal Sound Separation data? *arXiv preprint arXiv :2011.00803*.
- Woodard, J. P. (1992). Modeling and classification of natural sounds by product code hidden Markov models. *IEEE Transactions on Signal Processing*, 40(7) :1833–1835.
- Wu, Y. and Lee, T. (2019). Enhancing sound texture in CNN-based acoustic scene classification. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 815–819.
- Wysocki, L. E. and Ladich, F. (2005). Hearing in fishes under noise conditions. *Journal of the Association for Research in Otolaryngology*, 6(1) :28–36.
- Xu, Y., Kong, Q., Wang, W., and Plumbley, M. D. (2017). Large-scale weakly supervised audio classification using gated convolutional neural network. Technical report, DCASE Challenge.

- Xu, Y., Kong, Q., Wang, W., and Plumbley, M. D. (2018). Large-scale weakly supervised audio classification using gated convolutional neural network. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125.
- Yang, W., Liu, C., and Jiang, D. (2018). An unsupervised spatiotemporal graphical modeling approach for wind turbine condition monitoring. *Renewable Energy*, 127 :230–241.
- Yu, C., Barsim, K. S., Kong, Q., and Yang, B. (2018). Multi-level attention model for weakly supervised audio classification. *arXiv preprint arXiv :1803.02353*.
- Yu, D., Varadarajan, B., Deng, L., and Acero, A. (2010). Active learning and semi-supervised learning for speech recognition : A unified framework using the global entropy reduction maximization criterion. *Computer Speech & Language*, 24(3) :433–444.
- Zhang, Z. and Schuller, B. (2012). Semi-supervised learning helps in sound event classification. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 333–336.
- Zhao, S., Heittola, T., and Virtanen, T. (2020). Active learning for sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28 :2895–2905.
- Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National Science Review*, 5(1) :44–53.
- Zhuang, X., Zhou, X., Hasegawa-Johnson, M. A., and Huang, T. S. (2010). Real-world acoustic event detection. *Pattern Recognition Letters*, 31(12) :1543–1551.
- Zinemanas, P., Cancela, P., and Rocamora, M. (2019). MAVD : A dataset for sound event detection in urban environments. In *2019 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 263–267.