Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

Analysis and development of an end-to-end convolutional neural network for sounds classification through deep learning techniques.

RECEPTIONAL WORK that to obtain the **DEGREE** of
MAESTRO EN SISTEMAS COMPUTACIONALES

Presents: **CARLOS ALBERTO GALINDO MEZA**

Director: **DR. PAULO LOPEZ MEYER**

Tlaquepaque, Jalisco. November of 2021

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

Análisis y desarrollo de una red neuronal convolucional end-to-end para clasificación de sonidos a través de técnicas de aprendizaje profundo.

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: **CARLOS ALBERTO GALINDO MEZA**

Director: **DR. PAULO LOPEZ MEYER**

Tlaquepaque, Jalisco. noviembre de 2021.

ACKNOWLEDGEMENTS

I am truly grateful with Dr. Paulo Lopez Meyer whose conference inspired me to study a postgraduate degree oriented to artificial intelligence, and afterwards agreeing to be the director of this work. Thanks to his constant guidance and effort, this research is a reality.

I want to express my gratitude to Juan Antonio Del Hoyo Ontiveros and Jose Israel Torres Ortega whose support and research effort contributed directly to the object of study and data processing for the development of this work.

My thanks to Dr. Francisco Rangel who for many years has encouraged a huge number of engineers to continue improving themselves day to day, and who under his management received flexibility and support to study this master's degree.

I am grateful to Intel Corporation for the financial support required to complete my graduate studies.

I want to thank CONACYT for the support provided by awarding me the 501770 scholarship.

Finally, I am grateful to the ITESO teaching staff for the excellent work they do every day and their constant collaboration with the industry.

AGRADECIMIENTOS

Quiero agradecer infinitamente al Dr. Paulo Lopez Meyer cuya conferencia me inspiró a estudiar un posgrado orientado a la inteligencia artificial, y posteriormente aceptar ser director este trabajo. Gracias a su constante guía y esfuerzo esta investigación es una realidad.

Quiero expresar mi gratitud a Juan Antonio Del Hoyo Ontiveros y a Jose Israel Torres Ortega cuyo apoyo y esfuerzo en investigación aportaron directamente al objeto de estudio y procesamiento de datos para el desarrollo de este trabajo.

Gracias al Dr. Francisco Rangel quien por muchos años ha impulsado a una enorme cantidad de ingenieros a seguir superándose día a día, y quien bajo su gerencia recibí flexibilidad y apoyo para estudiar esta maestría.

Agradezco a Intel Corporation por el apoyo económico requerido para completar mis estudios de posgrado.

Quiero agradecer a CONACYT por el apoyo brindado al otorgarme la beca 501770.

Agradezco al personal docente del ITESO por el excelente trabajo que realizan día con día y su constante colaboración con la industria.

DEDICATION

I want to dedicate this work to my sister Karla Galindo, who has been a close witness of the effort and dedication that has been given to this document, always showing me flexibility and support. Wishing her the best of success in her start as an internal medicine resident.

To my mom, Ruth Meza, who every day has encouraged me to be a better student and human being. Her support and motivation in every aspect was essential to finish this thesis, this work is also hers.

To my dad, Carlos Galindo, whose tenacity has been a living example in my life of how perseverance is key to achieving any goal.

Finally, I want to respectfully dedicate this work to all the families who have suffered the loss of a close relative as victim of COVID-19 in the 2020 pandemic. Wishing that research and technology can always be used in favor of science and contribute to the society.

DEDICATORIA

Quiero dedicar este trabajo a mi hermana Karla Galindo, quien ha sido testigo cercano del esfuerzo y dedicación que se le ha dado a este escrito, siempre mostrándome flexibilidad y apoyo en todo momento. Deseándole el mayor de los éxitos en su inicio como residente de medicina interna.

A mi mamá, Ruth Meza, quien todos los días me ha alentado a ser mejor estudiante y ser humano. Su apoyo y motivación en todo aspecto fue indispensable para terminar esta tesis, este trabajo también es suyo.

A mi papá, Carlos Galindo, cuya tenacidad ha sido un vivo ejemplo en mi vida sobre como la constancia es clave para alcanzar cualquier objetivo.

Finalmente, quiero dedicar respetuosamente este trabajo a todas las familias que han sufrido la pérdida de algún ser cercano como víctima del COVID-19 en la pandemia del 2020. Deseando que la investigación y tecnología siempre pueda usarse en pro de la ciencia y aportar a la sociedad.

SUMMARY

The present work provides the analysis and continuous development of an artificial intelligence engine aimed to audio classification. Chapter 1 presents a background on the different audio-related tasks that research community has followed over the years, also states the core hypothesis of this work, and defines general and specific objectives to contribute to the enhancement of performance over an end-to-end audio embeddings generator. Chapter 2 presents state-of-the-art methods and published works that are mainly aimed to the development of audio classification and deep learning as disciplines with enormous potential to fulfill. Chapter 3 presents the conceptual framework in which this thesis is based on, split in two main sections: audio preprocessing and deep learning techniques. Each of these sections is divided among several subsections to represent audio classification process through deep neural networks. Chapter 4 provides a profound explanation of the audio embeddings generator named AemNet and its components, used as object of study which are further detailed in the following subsections. Initial experimentation was done over this approach and presented experimental results that suggested an improved performance by modifying stages of the neural network architecture. Chapter 5 is the first target application of our AemNet adaptation that was submitted to the DCASE 2021 challenge. The details on the challenge and results are described in this chapter sections, as well as the methodology followed to present our submission. Chapter 6 is the second target application and the first aimed to respirational sounds. The ICBHI challenge is explained in this chapter sections as well as the methodology and experiments performed to reach a robust classifier that distinguishes four different cough anomalies. A paper was created out of the proposed solution and presented into the IEEE LA-CCI 2021. Chapter 7 takes leverage on the several previous results to fulfill a modern approach such as COVID-19 detection, which data source collection and experimentation are described profoundly and experimental results suggest that a residual network adaptation named AemResNet, can comply to distinguish COVID-19 patients from cough and breath sounds. Finally, the conclusions of all this research and results evaluated in each target applications are discussed in chapter 8.

RESUMEN

El presente trabajo estudia el análisis y desarrollo continuo de un modelo de inteligencia artificial orientado a la clasificación de audio. El capítulo 1 presenta antecedentes sobre las diferentes tareas relacionadas a audio que la comunidad de investigación ha seguido a lo largo de los últimos años, también establece la hipótesis central de este trabajo y define objetivos generales y específicos para contribuir a la mejora del rendimiento sobre un generador de *embeddings* de audio de tipo *end-to-end*. El capítulo 2 presenta los métodos de vanguardia y trabajos publicados que se enfocan principalmente al desarrollo de la clasificación de audio y el aprendizaje profundo como disciplinas que aún tienen un gran potencial. El capítulo 3 presenta el marco conceptual en el que se basa esta tesis, dividido en dos secciones principales: preprocesamiento de audio y técnicas de aprendizaje profundo. Cada una de estas secciones se divide en varias subsecciones para representar el proceso de clasificación de audio a través de redes neuronales profundas. El capítulo 4 brinda una explicación profunda del generador de *embeddings* de audio llamado AemNet y sus componentes, utilizado como objeto de estudio, donde se detalla en las siguientes subsecciones. Se realizó una experimentación inicial sobre este enfoque y se presentaron resultados experimentales que sugirieron un mejor rendimiento mediante la modificación de las etapas de arquitectura de la red neuronal. El capítulo 5 es la primera aplicación objetivo de nuestra adaptación de AemNet que se presentó al desafío DCASE 2021. Los detalles sobre el desafío y los resultados se describen en las secciones de este capítulo, así como la metodología seguida para presentar nuestra propuesta. El capítulo 6 es la segunda aplicación objetivo y el primero en apuntar a los sonidos respiratorios. El desafío de ICBHI se explica en las secciones de este capítulo, así como la metodología y los experimentos realizados para llegar a un clasificador robusto que distingue cuatro anomalías de tos diferentes. Se creó un artículo a partir de la solución propuesta y se presentó en el IEEE LA-CCI 2021. El capítulo 7 aprovecha los diversos resultados anteriores para cumplir con un enfoque moderno como lo es la detección de COVID-19, cuya recopilación y experimentación de fuentes de datos se describen profundamente y los resultados experimentales sugieren que una adaptación de red residual denominada AemResNet, puede cumplir la función de distinguir a los pacientes con COVID-19 a partir de tos y sonidos respiratorios. Finalmente, las conclusiones de toda esta investigación y los resultados evaluados en cada una de las aplicaciones objetivo se discuten en el capítulo 8.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
AS	Average Score
ASEC	Audio Scene and Event Classification
ASR	Automatic Speech Recognition
AUTH	Aristotle University of Thessaloniki
CNN	Convolutional Neural Network
COPD	Chronic Obstructive Pulmonary Disease
COVID-19	Coronavirus
CQT	Constant Q Transform
DCASE	Detection and Classification of Acoustic Scenes and Events
DCT	Discrete Cosine Transform
DO	Dropout
DSP	Digital Signal Processor
DW	Depth Wise
DWSC	Depth Wise Separable Convolutions
e2e	End-To-End
ESSUA	School of Health Sciences from University of Aveiro
FIR	Finite Impulse Response
FP32	Single-precision floating-point format
GAM	Gammatone Filter
GMM	Gaussian Mixture Model
GPUs	Graphic Processing Units
HLF	High-level Features
HMM	Hidden Markov Modeling
HPSS	Harmonic Percussive Source Separation
HS	Harmonic Score
KNN	K-Nearest-Neighbor
LLF	Low-level Features
LR	Learning Rate
LT	Lottery ticket
MACs	Multiply-accumulate operations
mAP	Mean Average Precision
MFCC	Mel Frequency Cepstral Coefficients
ML	Machine Learning
NLP	Natural Language Processing
NN	Neural Network
ResNet	Residual Network
SARS-CoV2	Severe Acute respiratory Syndrome Coronavirus
SNR	Signal-to-Noise Ratio
STFT	Short-Time Fourier-Transform

SVM	Support Vector Machine
TDNN	Time-Domain Neural Network
VGG	Visual Geometry Group
WAV	Waveform Audio File Format
WM	Width Multiplier

1. INTRODUCTION

Summary: *This chapter presents a brief introduction to the core objectives and motivation of this work. Section 1.1 breaks down several audio related applications and how audio classification tasks have historically taken place through different mathematical and statistical algorithms. Section 1.2 describes the motivation of this work and explains the leverage of using end-to-end solutions for audio classification engines. Section 1.3 highlights the obstacles for the audio research community when dealing with machine learning or deep learning techniques when applied to classification problems. Section 1.4 presents the hypothesis of this work that states the possibility to increase the performance of an audio embeddings generator by modifying its main trainable feature extraction stages, aiming to new applications such as respiratory sounds. Section 1.5 presents the details on the general and specific objectives of this work which consist of objectively evaluating the performance of an enhanced audio embeddings generator through two main low-level-feature extractions, an adaptation of a residual network, the focal loss approach, and its application to different datasets.*

1.1. Background

Looking at the recent scientific literature, it can be observed that there has been a large amount of work focused on the development of systems that use artificial intelligence (AI). It is not deniable that image recognition has become the most popular area for AI and for the development of novel machine learning (ML) algorithms and has presented continuously state-of-the-art (SOTA) research peaks and findings. Because of this, audio-oriented tasks have mostly leveraged on the techniques developed by the computer vision community.

The use of artificial intelligence in audio related tasks finds its application in several domains:

- Automatic Speech Recognition (ASR)
- Natural Language Processing (NLP)
- Speaker Recognition
- Source Localization
- Speech Emotion Recognition
- Audio Scene and Event Classification (ASEC)

The technology presented in this work focuses on the audio classification tasks and applications. Historically, ASEC has been addressed through several algorithm approaches that comprise different audio signal processing and transformations through statistical stochastic processes. In [1], some audiovisual sports media was classified between its own speech, music and environmental sounds using Mel Frequency Cepstral Coefficients (MFCC) and a Gaussian Mixture Model (GMM) as classifier. Reference [2] is based on a K-nearest-neighbor (KNN) approach to perform a similar task of classifying environmental sounds from speech on audiovisual data. Also, hidden Markov modeling (HMM) has been a widely used approach for several years such as in [3] where is the main approach for audio event classification.

In the latest years, more efficient ASEC uses several ML techniques such as support vector machines (SVM), neural networks (NN) and more recently convolutional neural networks (CNNs). In the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge 2020, the top 50 submissions use actively deep learning techniques and a wide use of CNN topologies across different tasks. As expected, DCASE 2021 edition on Task 1A holds all its submissions with all CNN approaches [4]. Residual networks are the most common CNN architecture used in both editions, hence the motivation of this work on using this same network as key object of study.

On the other hand, ASEC end-to-end (e2e) approaches have shown a good performance by implementing a trainable feature transformation of audio data, instead of a fixed manual extraction through dedicated hardware or software. AclNet [5] is one of many previous e2e proposals aimed to ASEC, which creates the base model topology of AemNet [6], an audio embeddings generator and core subject of this research.

1.2. Justification

As mentioned above, computer vision together with NLP are the most popular AI research paths, leaving audio-oriented tasks with a long exploration on the way. From audio-oriented tasks, the most studied fields are human-speech recognition or ASR, digital signal processing and music tagging and generation [7]. But in our daily lives, there are several sounds that differ from speech and music that provide an important context for us humans to understand our world; to be precise, there is a wider number of acoustic sound types than the fields mentioned before. On recent years, there has been a huge effort by the scientific community to develop efficient ASEC models, mostly oriented to highly crowded streets label classification, human-like sounds, or in-house potential hazards (e.g., glass breaking, baby crying) [4], [8], [9].

ASEC has taken several turnarounds aiming to new investigation segments such as marine biology (whale-call classification for sub-populations and biological research) [10], industrial-specific (acoustic signatures of machines and equipment to monitor their operational robustness) [11] and healthcare: classification of respiratory sounds [12]–[16], assessment of depression and suicide risk using speech analysis [17]. Just the DCASE challenge is split in the following tasks: unsupervised anomalous sound detection for machines, sound event localization and detection with directional interference, sound event detection and separation in domestic environments, few-shot bioacoustics event detection, as well as automated audio captioning [4]. Also, current times that show a promising path to keep following such as the analysis of COVID-19 diagnosis and classification through an app or web-based audio [18].

The audio classification applications mentioned above leads us to ponder about a single audio embedding generator, i.e., a mechanism to condense an audio signal into a rich discriminative descriptive vector, that can be adapted into several audio classification tasks, and can be easily ported to different platforms, which is the main objective of [6]. While this is not the first of its class, it does create a path to keep on researching this approach by following a fine-tuning strategy that leads to competitive results on the ASEC application we are interested in.

Hence, the purpose of this work to make further investigation on developing AemNet and evaluate how ASEC is a need on modern society; its flexibility on several subjects and on-going work makes this a great opportunity to develop SOTA technologies aimed to modern day problems. In this thesis, we analyze the usage of deep learning techniques for the development of a flexible e2e convolutional neural network for applicability in both to ASEC and relevant modern biomedical problems.

1.3. Problem

There is an increasing amount of research based on music and speech-based applications. Whenever we are dealing with audio classification problems, there is a limited audio data for classification tasks (music, natural language processing, voice recognition and ASEC). This is one of several issues that ASEC deals with, hence the need to take the most advantage of large available datasets. Compared to computer vision, the number of papers and tasks performed by the research community is exceeded greatly against audio classification tasks, hence the need to use different techniques for deep learning, such as data augmentation, image-like representation of audio and models pretraining.

On the other hand, ASEC research community is mostly oriented to the preprocessing audio stage joined with a classifier stage, without considering the complexity of the model. SOTA in ASEC states that deep learning brings the best performance whenever a preprocessing stage is previously done, and the deeper and greater the classifier stage gets, the better results seem to be achieved. The concept that a deeper and more complex model brings better results outsides the fact that the ASEC applications are part of our day-to-day lives, and whenever they are used these are limited to the hardware they are programmed in. For instance, the UrbanSound8K [9] dataset was collected through different devices in several places of a city; if we were to classify the same classes with new urban sounds, most of the best-performance approaches are based on a raw audio handling transformation through a digital signal system and then put through a classifier stage in which the complexity may exceed most hardware used for said applications.

1.4. Hypothesis

An audio embeddings generator [6] can improve its performance through modifications on its low-level feature configurations and high-level features, changing its core topology from a VGG-ish to a ResNet18, achieving a competitive performance by aiming at healthcare-oriented applications such as respirational sounds. When the target application is restricted to a limited model complexity, this same audio embeddings generator can be optimized within its feature configurations through deep learning and optimization techniques to achieve a robust response to the target application.

1.5. Objectives

1.5.1 General Objective

To analyze AemNet and its core features to evaluate its robustness and improve its performance on new target applications aimed to ASEC.

1.5.2. Specific objectives

1. Understand which low-level-feature extraction configuration shows a better performance for AemNet.
2. Understand how the high-level-feature stage of AemNet can improve performance when compared to conventional deep layers and its convolution operation variants.
3. Understand how focal loss can support e2e CNN to handle unbalanced datasets.
4. Contribute to the ASEC research community by submitting an AemNet variant to the DCASE2021 challenge.
5. Understand the behavior of AemNet oriented to respiratory sounds on the ICBHI challenge and Cambridge University crowdsourced COVID-19 dataset.

1.6 Scientific, technological novel or contribution

This work presents a configurable e2e CNN based on AemNet [6] that addresses different ASEC tasks. Through different deep learning and optimization techniques, this resulted into several experimentations on AemNet which targeted three main applications: a low-memory size audio embeddings model that presents a competitive response on ASEC, a robust performance residual network CNN that classified different lung respirational sounds, and a solid contribution to the pandemic research community where we address to classify COVID-19 from non-COVID patients using breaths and coughs as input.

2. STATE OF THE ART OR TECHNIQUE

Summary: *In this chapter, we address the leading-edge audio classification techniques aimed to several applications. Section 2.1 brings an overall explanation of a common audio processing pipeline from a WAV file to the desired representation of the audio. The representation of the signal is key to what kind of approach is followed; both are explained in following sections. Section 2.2 brings the most recent conventional image-like representation of audio along the deep learning techniques performed, how robust is their performance, and the motivation of developing computational vision algorithms oriented to audio. On the other hand, section 2.3 explains how e2e have shown competitive results against image-like approaches sometimes outperforming ASEC applications, with all the benefits of keeping a simpler model that does not require any external hardware to preprocess raw time-domain signals.*

2.1 Audio classification overall process

Preprocessing data is an important stage of any image or sound classification task. We must understand what type of preprocessing is performed when a raw audio clip enters on any current SOTA audio classifier. SOTA methods for ASEC use deep learning, a variation of deep layer neural networks which takes advantage of a significant number of stacked neural layers to provide a better performance than conventional neural networks, this is explained profoundly on section 3.3.

Usually, the overall process of preprocessing audio signals is as follows:

1. WAV files are resampled to a preferred sample frequency of the user (16 KHz, 44.1 KHz are some of the most used sample frequencies chosen by the research community).
2. Audio processing techniques such as data augmentation, several types of filtering or random noise addition.
3. Transformation of raw time domain audio to a corresponding spectrogram or its equivalent 2-dimensional form (e.g., MFCC, Mel-spectrogram, filter banks).
4. With an image-like representation of the corresponding audio, the preprocessing is finished and ready to be the input of a standard classifier, SOTA points that most of these approaches are CNN-based.

2.2 Conventional audio classification approaches

Pham et al [19] explores three different approaches to form an ensemble of three systems and test each individually: log-Mel, Gammatone filter (GAM) and Constant Q Transform (CQT). According to [19], the best approach to continue was GAM, in which each audio sample is transformed into a Gammatone spectrogram with 128 GAM filters; this output creates an image-like representation vectors of size 128x128. After some mixup data augmentation [20] these vectors are entered into a VGG-similar network for feature embedding learning.

In [21], the DCASE 2017 dataset is the reference used for unconventional addition to commonly used spectrograms. A Harmonic Percussive Source Separation (HPSS) algorithm takes advantage of the binaural (2-channel) source audio to decompose these signals into harmonic and percussive components and creates a power spectrogram of their combination. The spectrogram operations use a short-time Fourier transform (STFT) that analyzes each harmonic or percussive component, based on median filters. Several input layers are created before a classifying stage such as the main input HPSS spectrogram of both channels, their addition, subtraction and average. Each of these, are entered into a multi-layer perceptron and a softmax layer to perform the classification.

Bai et al [22] uses Mel-frequency filter banks features from certain frame-length and its derivatives. This creates a 240-dimensional input to the proposed time-domain neural network (TDNN) that learns embedding features and is later connected to a classifier stage also based in CNN.

On the other hand, the use of MFCCs is also used in [23], [24], which pre-process audio to create a logarithmic power spectrum on a nonlinear Mel scale of frequency. The conversion of audio to an image comes at a cost of reducing information about the audio by using several transformation parameters. After this stage, [23] uses a series of 7 layers along max pooling operations perform the urban sounds

classification; whereas [24] has a similar CNN approach to classify rainfall intensity based on the timbre of several materials.

These recent works are just a handful of the latest approaches over the past couple of years regarding the SOTA of audio classification. We also observe that the audio-oriented tasks differ from speech recognition or music and have a wide variety of applications from urban sounds to rainfall classification, all of them using similar image-like approaches when processing raw audio samples. In conclusion, this widely used practice of treating an audio as an image is still a common practice to recent ASEC applications and is still being developed by researchers worldwide implementing several computer vision techniques.

2.3 E2E approaches

Approaches known as e2e do not rely on an image-like representation of the audio signal through spectrograms or filter banks, these rather use different types of preprocessing techniques applied to raw audio samples before entering deep learning architecture solutions. More recently, residual networks have been explored for audio-related tasks and have provided satisfactory results into well-known ASEC applications. In [25], six different residual block architectures were the object of experimentation over UrbanSound8K [9] and ESC-50 [8] datasets in which is observed how the performance is compared among different residual blocks configurations. It is concluded how the order of activation function layers and batch normalization operations inside the residual block has a direct impact on the performance of the CNN. Also, this e2e approach uses common audio techniques such as zero-padding and clipping of the raw time-domain samples as preprocessing stage and distinguishes between an audio scaling of maximum value and a zero mean standard normalization. On the other hand, [26] evaluates an e2e one-dimensional CNN which learns representation directly from the audio signal. Their approach deals with different sample lengths and evaluates the performance of GAM filter banks as a one-dimensional input to a multi-layer CNN topology that uses UrbanSound8K as reference dataset, outperforming several two-dimensional approaches with much less parameters. The motivation beneath Gammatone filter banks is to simulate the physiology behind peripheral auditory processing. This work also used different overlapping windows sample on the time-domain audio as means to reach data augmentation benefits maximizing the use of information. On [15], the squeeze-and-excitation (SE) and residual blocks are used into the model architecture to increase performance on their e2e approach. Raw waveforms are the input to the CNN with a small filter size of 2-3 samples in all layers, where the sub-sampling is done by max-pooling (excepting the first convolutional layer). The DCASE2017 dataset was used to evaluate the acoustic scene tagging performance of this model. This work also presents a deep analysis on several SE block parameters, residual blocks, and an ensemble of both, using music and keyword spotting audio datasets. Accuracy and F1-score were used as metrics, which also show a competitive performance against Mel spectrogram approaches on several of these tasks, sometimes outperforming them, giving a higher confidence to the e2e approach development oriented to ASEC applications. These previous works show the potential of audio-oriented research when it comes to developing e2e solutions, hence also one of the motivations of this work: to develop and improve a high-performance audio embeddings generator oriented to ASEC tasks.

3. CONCEPTUAL FRAMEWORK

***Summary**: In this chapter we address the basics of audio waveform processing explaining the concept of an audio waveform and its preprocessing aimed to audio classification tasks. Section 3.1 explains the main elements of a signal, the importance of a frequency domain transformation with the Fourier transform, and the various representation of audio. Section 3.2 describes the usual processing stages a time-domain signal goes through to finally create audio representations in the form of filter banks or MFCCs. Finally, Section 3.3 focuses on the deep learning concepts used in all this work, from defining why a neural network arrangement is called deep, through the CNN architectures used in this work and optimization techniques that improve the portability of an e2e audio embeddings generator.*

3.1. Audio and signal basics

To properly break down all elements regarding audio classification, we first need to talk about audio and how it is usually handled on classification applications. What us humans usually name as sound is nothing but air pressure changes that are propagated through a medium (usually air) and its frequency and intensity pressure variations can be represented as a one-dimensional signal over time [27].

In physics, a signal has several properties that can be measured and evaluated for different engineering purposes. Any signal wave contains a wavelength, amplitude, and a period.

The amplitude of audio signals is usually measured in represented in decibels (dB) which is a logarithmic unit of intensity or magnitude [28], whereas the period is the lapse in seconds (s) taken to the signal to complete one single cycle (whenever this signal is periodic). On the other hand, the frequency of an audio signal can be calculated as the inverse unit of the period and is defined as the number of cycles a signal is repeated per second, using Hertz (Hz) as its unit.

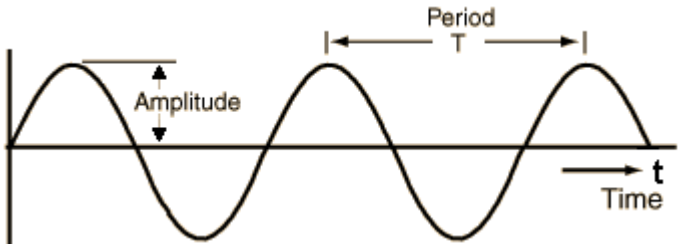


Figure 1. Main elements in a temporal signal representation [27].

The frequency of a sound wave is directly associated to human perception [28] or sound pitch. The audible range of frequencies for a human is 20 – 20 KHz.

Sound can be represented and plot in two different domains: analog and digital. Analogically, any signal can be measured and analyzed in an oscilloscope where we can observe the temporal behavior of the signal over time. We understand that this a continuous function which contains a real value in every point of time. Sound can also be represented digitally, usually an analog-digital-converter implemented in a digital device samples the sound at certain frequency rate and creates a fixed amplitude value per sample in each lapse, depending on the sample frequency. Both approaches of sound handling contain their own pros and cons.

On a deeper level, sound signals are not analyzed in a time-domain basis. Engineers constantly look on the intrinsic elements of the audio such as its frequency components. One of the most common mathematical transformation to convert a signal from the time domain to frequency is the Fourier transform. Fourier transform states that any signal can be decomposed as the addition of different sinusoidal signals of different frequencies and amplitudes. Equation 1 expresses the trigonometric Fourier transform for continuous periodic signals. Its goal is to solve coefficients for a_0, a_n, and b_n.

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\frac{n\pi x}{L} + b_n \sin\frac{n\pi x}{L} \right)$$

Equation 1. The trigonometric Fourier transform for continuous periodic signals.

Signal analysis is a common field in which the Fourier transform is used on its multiple forms (continuous or discrete) depending on the preprocessing analysis of interests.

Whichever method we use for signal analysis, we can represent any real-life sound in a graph which plots the magnitude of each frequency point, instead of a time-domain analysis. This visual representation is called a spectrum. Spectrums are an alternative way to represent the same signal, and can help to visualize the frequency components of a given signal and can be calculated by any of the techniques mentioned above or through a more sophisticated way using spectrum analyzers.

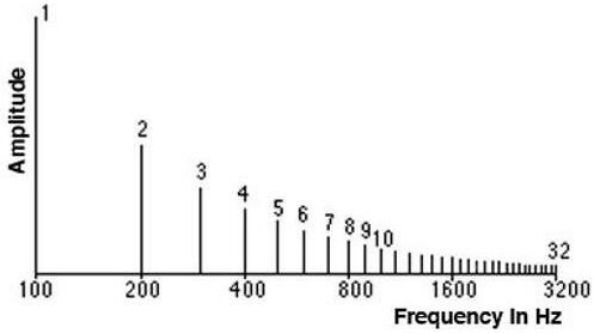


Figure 2. Frequency components of a time signal inside a lapse [27].

A spectrogram is a visual representation of the frequency spectrum over time. It displays time on the x-axis and frequency on the y-axis, hence the spectrum of any signal varies over time [27]. For frequency magnitude purposes, a different color is defined to plot the intensity of that spectrum at any given point. In other words, we are representing sound into an image.

There are several ways of generating a spectrogram but given the recent needs of audio analysis oriented to artificial intelligence, there exists software and programming libraries that already implement all mathematical operations needed to properly plot a spectrogram based on raw audio clips.

3.2. Audio pre-processing techniques

Several applications such as speech recognition or audio classification share the same processing techniques. Most of them involve computing filter banks and MFCCs as an input to the proposed machine learning or deep learning system. An overall sequence of computing filter banks and MFCCs is described in the following subsections.

3.2.1 Pre-emphasis

First step for most audio preprocessing is to apply a pre-emphasis filter on the signal so high frequencies can stand out. This fulfills to balance the frequency spectrum, avoid numerical problems during Fourier transform and may also improve the Signal-to-Noise Ratio (SNR) [29].

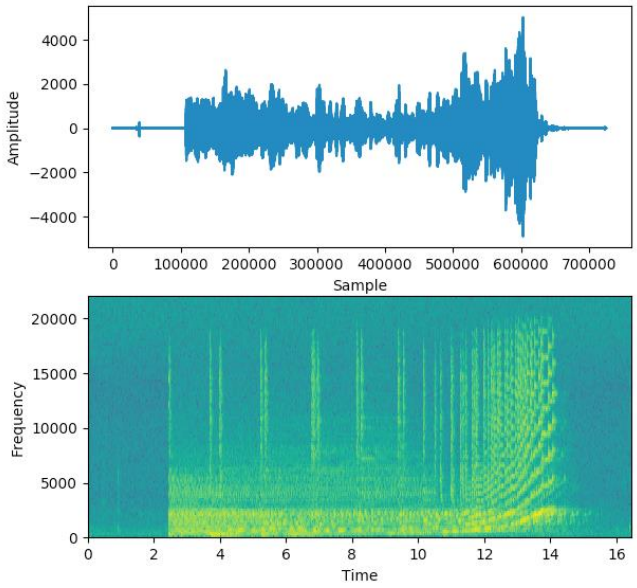


Figure 3. Temporal and spectral representations of an audio signal [27].

3.2.2 Framing

After pre-emphasis, the signal is split into several frames, this helps to obtain a good approximation of the frequency contours of the signal by joining adjacent frames. Normally the frame size varies from 20 ms to 40 ms with 50% ± 10% overlap between consecutive frames [29].

3.2.3 Window

After the framing process, a Hamming window function is applied to each frame. It is expected that the Hamming window stage counteracts the assumption that FFT data is infinite, also reducing spectral leakage. Equation 2 expresses the mathematics behind the Hamming window.

$$w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{N} - 1)$$

Equation 2. Hamming window function. Where $n \epsilon 0 < n \leq N - 1$.

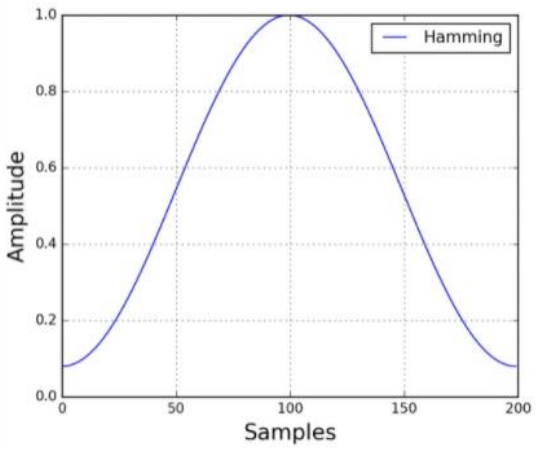


Figure 4. Hamming Window [29].

3.2.4 Mel scale

The Mel scale is a subjective scale of pitches judged by listeners to be equally distant from one to other. Figure 5 shows a plot of the Mel scale in which is observed that below 500 Hz the Mel and hertz scales coincide; above that, larger intervals are judged by listeners to produce equal pitch increments. The reference point between both scales is defined by equating a 1000 Hz tone with a pitch of 1000 Mels [30].

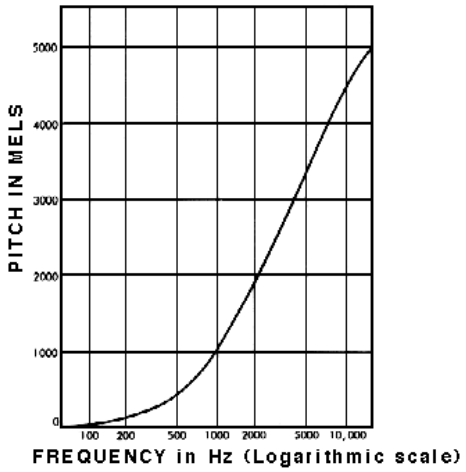


Figure 5. Mel scale plot against Hertz scale [30].

We can convert between Hertz (f) and Mel (m) using the following equations:

$$m = 2595 \log_{10}(1 + \frac{f}{700})$$

Equation 3. Conversion from frequency (Hz) to Mels [30].

$$f = 700\ (10^{\frac{m}{2595}} - 1)$$

Equation 4. Conversion from Mels to frequency (Hz) [30].

3.2.5 Fourier transform and power spectrum

An N-point FFT, also called Short-Time Fourier-Transform (STFT), is applied on each frame to calculate the frequency spectrum where N is typically 256 or 512. Afterwards, the power spectrum of the output FFT is calculated using Equation 5.

$$p = \frac{|FFT(x_i)|^2}{N}$$

Equation 5. Calculation of the power spectrum from the FFT output of the signal. Where x_i is the i^{th} frame of signal x [29].

3.2.6 Finite Impulse Response (FIR) decimation filter bank.

The final step to computing filter banks is applying triangular filters, typically 40 filters, on a Mel-scale to the power spectrum to extract frequency bands. Figure 6 represents the filter bank shape as triangular, having a response of 1 at the center frequency and decrease linearly towards 0, until it reaches the center

frequencies of the two adjacent filters where the response is 0. Figure 7 illustrates the output after applying the filter bank to the power spectrum of the signal.

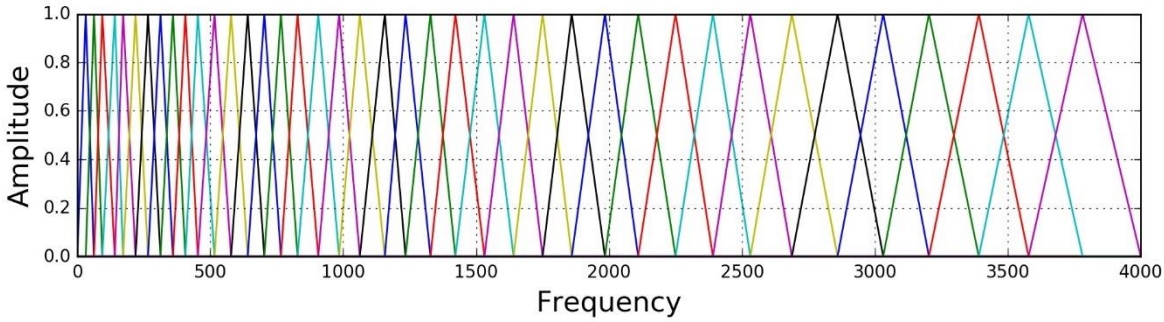


Figure 6. Filter bank representative shape response [29].

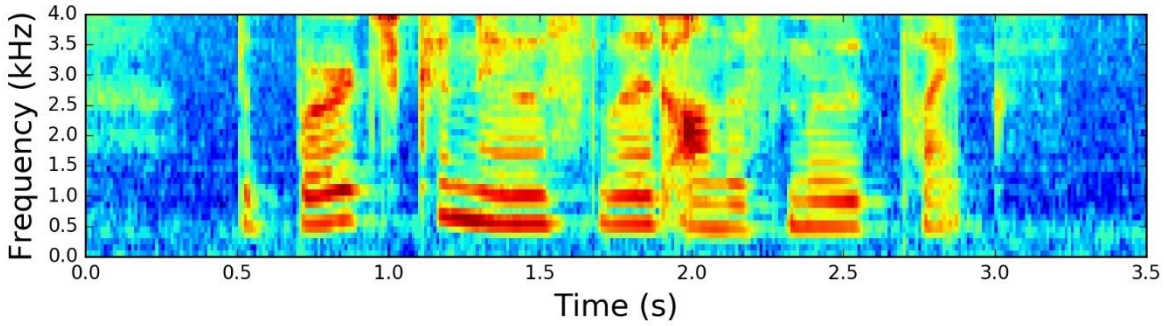


Figure 7. Filter bank representation of an audio signal [29].

3.2.7 MFCC

Discrete Cosine Transform (DCT) can be applied to decorrelate the filter bank coefficients and yield a compressed representation of the filter banks. Also, applying a sinusoidal filter to the MFCC to de-emphasize higher MFCCs has claimed to improve performance [29].

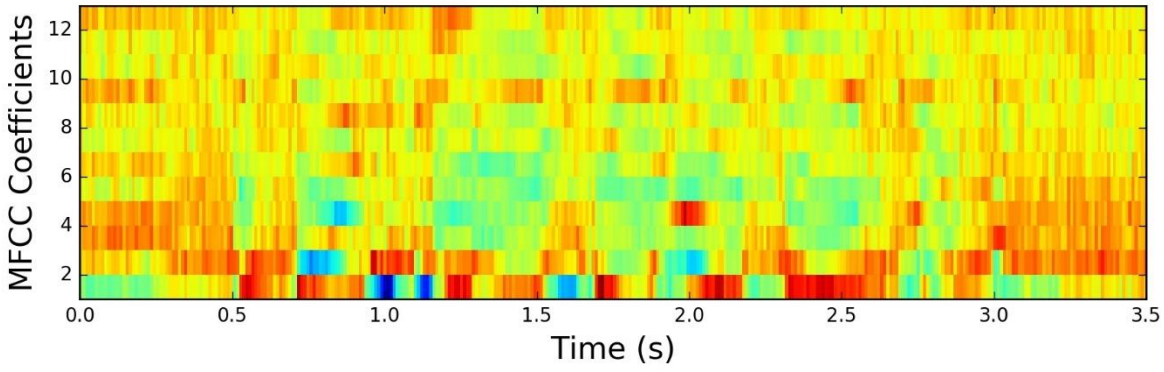


Figure 8. The DCT applied to the filter bank creates a MFCC representation [29].

To balance the spectrum and improve the SNR, we can subtract the mean of each coefficient from all frames through a mean normalization filter.

3.3. Deep learning

Deep learning is a subset of artificial neural networks which consist of linking multi-layer structures of CNNs creating deeper model algorithms. Andrew Ng describes the idea of deep learning as a brain simulation that hopes to make learning algorithms much better and easier to use and making revolutionary advances in AI [31]. In [32], deep learning is described as computational models that are composed of multiple processing layers to learn data representations through multiple abstraction levels.

3.3.1. CNN architectures

Visual Geometry Group (VGG): Consists of a simple architecture, using only blocks composed of an incremental number of convolutional layers with 3x3 size filters [33]. It also includes max-pooling blocks that are spread between convolutional layers, followed by a classification block. It usually comes with a number on its name (e.g., VGG-16, VGG-19) that refers to the number of weighted layers each network contains.

Residual Network (ResNet): One of the issues seen commonly in deep learning is the "vanishing gradient", due to the great depth of the network, the gradients from where the loss function is calculated easily trend to zero. This phenomenon creates an effect in which the weights are never updating their values, therefore, no learning is done [34]. With ResNet, the gradients can skip connections backwards from later layers to initial filters.

4. UNDERSTANDING AEMNET

***Summary**: The purpose of this chapter is to have a proper understanding of what is an audio embeddings generator, it describes its main components through several sections. Section 4.1 describes AemNet as a general-purpose e2e audio embedding generator that can be adapted to various ASEC applications; learning directly from raw audio samples to create an embeddings output tensor which is later input to a classifier. Also, it defines each concept of AemNet which are: e2e, audio and embeddings. Section 4.2 describes the purpose of an e2e model over other approaches, the importance of a low-complexity model for real-life scenarios and how can an e2e model achieve comparable results against other works. Section 4.3 analyzes the AemNet as the main core of this work describing its three main blocks: 1) low-level-features (LLF) block and its two main configurations, which create an image-like tensor representation and can later be the input to the CNN of our preference, 2) high-level-features (HLF) block which follows a similar structure to a common computational vision CNN, its features such as the existence of a width-multiplier and the different types of convolutional operations and finally 3) a classifier layer which varies depending on the convolutional operation of the HLF and outputs a softmax response based on the embeddings created by the two previous blocks. As well, section 4.4 describes the datasets used to evaluate the robustness of the AemNet, the purpose of each dataset and an overall description of them. Section 4.5 describes the experiments performed over the high-level -features block and the different approaches to load a pretrain model into the AemNet, presents results based on the three datasets described in the previous section and concludes a final model to keep on using for this work. Finally, section 4.6 explains the experimentation done in the low-level-features block of two different configurations, shows a decision on which one presents a better performance and the discussion around it.*

4.1. Audio embeddings generator disambiguation

AemNet [6] is a general-purpose e2e audio embedding generator that can be adapted to various ASEC applications; it learns to extract the required information from raw audio samples and the inner convolutional layers compute the most efficient weights to create an embeddings tensor, which becomes the input to a classifier layer that uses an activation function to match an audio input to a label correctly.

The definition of AemNet can also be interpreted as the combination of the concepts explained next:

End-to-end: E2E learning is a deep learning topic which takes advantage of deep neural networks, and in the past few years, has been a popular technique used for several on-field applications. E2E is understood as an integral solution system in which everything is calculated inside the CNN; neural network weights are learned by raw input and eventually an output is provided. In the ASEC context, its main purpose is to handle audio signals of different lengths directly from the input and achieve an efficient classification [26].

Audio: In ASEC, all scene and event surroundings that can be taken through a microphone through different devices. e.g., street traffic sound, breaking glass, bus driving by, or respiratory sounds.

Embedding: It refers to the mapping of a categorical variable into continuous values. Its purpose is mainly to cluster a variable and represent it as much as possible in a vectorized way [35]. This representation of audio signals in its most efficient way helps a CNN to be competitive in different applications.

For instance, if we were to represent a book through embeddings, we can define that in its more general way it can be out of two ways: science or arts. One other feature of books is the year they were published, hence: old or recent. Just taking these two features into account, the representation of a book on CNNs can easily be represented as a "science-recent" book, or in other words: [1,0] (science) and [0,1] (recent). With this, maybe we can add a different book topic that will generate its own embedding representation.

Pytorch [36] was the framework of choice for our AemNet base code. The rest of the following experimentations described in later sections of AemNet and its adaptations leverage most of the code already created.

4.2. Purpose of an E2E model

Over the years, the advances of improving accuracy trend to make deeper and more complicated networks to achieve higher accuracy [37], whereas this is not always the most efficient way to get it. Concepts such as latency, speed and size of a model are usually dismissed when creating a CNN by means of achieving its best performance, forgetting that the complexity of such models is quite important when they are needed on a production level. One of the main objectives of using an E2E system on ASEC applications is its deployment on a wide use of hardware systems that use low-power digital signal processors (DSPs) or neural-net accelerators.

The hardware in which such models can be stored can be custom build for specific applications and may have several types of accelerators and components within the same chip [38]. Applications in embedded

systems such as autonomous driving must consider all these aspects on complexity, speed, and peripherals on hardware, to provide an efficient on-time response to avoid collisions on the vehicle.

At the end, a CNN may always seem extremely fast and efficient when running on the graphic processing units (GPUs) where it is trained and validated, quite far from the destination circuit in which this can be implemented.

In conclusion, and E2E model avoids the use of the type of hardware described above such as DSPs for image pre-processing on spectral features by implementing its own spectral representation out of convolutional layers and comprises the advantage of data-driven learning to fulfill the ASEC task, considering the total complexity size of the model, while achieving competitive results [5], [6], [37] against other computational vision CNNs oriented to different audio datasets.

4.3. Analysis and breakdown of AemNet

As mentioned before, the development of this E2E system is the main purpose of this work. AemNet comprises three main blocks: low-level features (LLF), high-level features (HLF), and a classifier.

The combination of these blocks is represented in Figure 9 as the AemNet structure, creating an e2e CNN that accepts raw audio samples, and outputs a classification based on the ASEC application target. All layer weights in the LLF and HLF can be loaded from a pretrained model or learned on a data-driven manner, creating a series of embeddings which finalizes in a classifier with an output specific to the number of classes to categorize. Each block is explained in the following subsections.

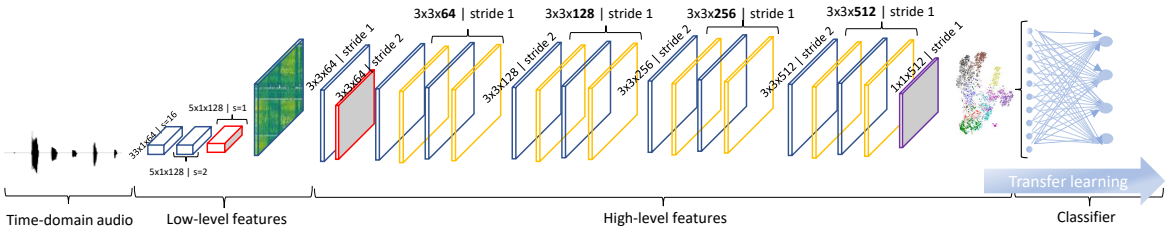


Figure 9. AemNet structure showing its core stages: LLF, HLF and Classifier blocks.

4.3.1. Low-level features (LLF)

The LLF block is the composition of two one-dimensional CNN layers that substitute the spectral feature pre-processing stage of common ASEC CNN approaches and its performance is equivalent to a FIR decimation filter bank. As a time-domain audio signal enters to the first layer, the LLF creates an output of 128 channels at a frame rate of 10ms after an added max-pool layer [6].

For instance, if the input to the LLF is a raw audio sample of 10 seconds, this will produce a tensor of size [128,1,1000], in which 128 is the number of channels, 1 is the dimension of the audio input and 1000 is the number of frames in 10 seconds (10 ms per second, 1000 for 10 seconds). The tunable hyper-parameters of this stage are the stride number and kernel size of convolutional layer 1 and 2 (S1, K1, S2, K2 respectively). These values will be later carried on for experimentation in the following sections.

The LLF block by itself creates its own image-like representation in tensors, which can later be the input to the computational vision CNN of our preference, demonstrating that each block can work independently.

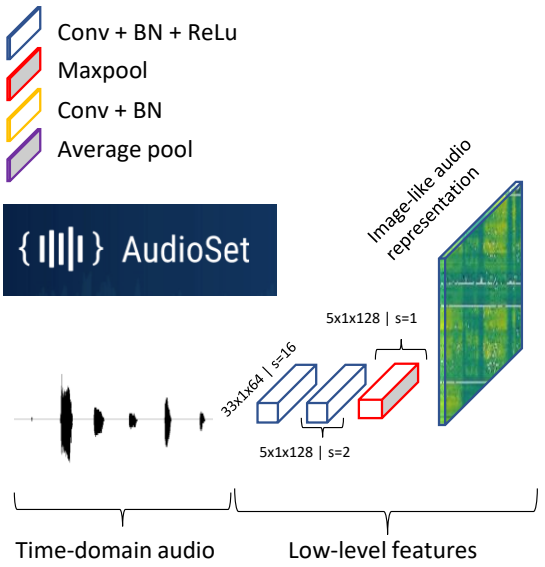


Figure 10. LLF stage. 2 linear CNNs act as alternative for image-like representations.

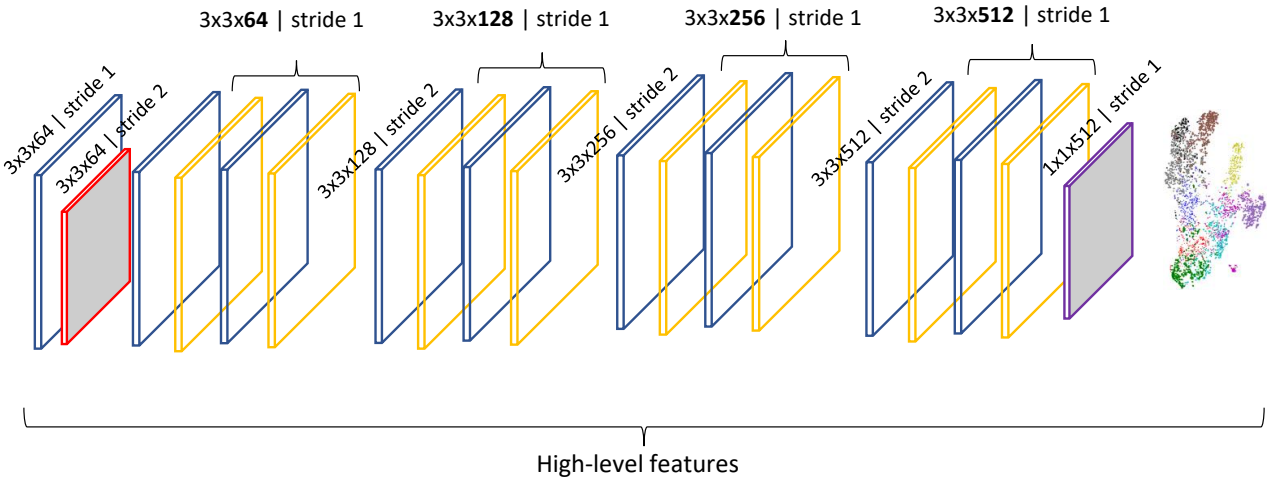


Figure 11. HLF block. This block can store any CNN architecture according to the target application.

4.3.2. High-level features (HLF)

The structure of the HLF block follows a similar structure to a common computational vision CNN. After a wide research on both [5], [6], it was found that VGG-like architecture achieves a good performance for ASEC applications. VGG topology is previously explained in section 3. The default HLF block of the

audio embeddings generator consists of a modified VGG shown in Figure 11, in which the depth and channel width are adjusted. An important feature of this block is the existence of a width-multiplier (WM) parameter that controls the network width size of the system based on [37]. When VGG-ish HLF is chosen, the WM parameter linearly scales the number of output channels from the third to eleventh CNN layer, its purpose is to thin a network uniformly at each layer [37].

For a given multiplier α, the number of input channels M becomes αM and the number of output channels N becomes αN.

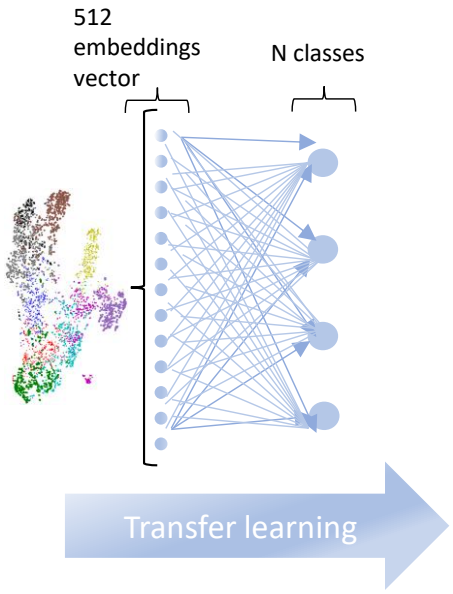


Figure 12. Classifier stage composed by a fully connected layer, a dropout operation and a softmax N-class NN that outputs a classification output.

The HLF block also embraces the concept of Depth Wise Separable Convolutions (DWSC / DW) used on MobileNet [37] and applied to AemNet. As previously explained, the DW approach has the effect of reducing computation and model size on an impactful way against a standard convolution.

Because of this, the use of this factorization operation on the CNN layers denoted the use of AemNet-DW.

AemNet-DW implements two additional modifications in its LLF block: K1 and K2 were redefined to 33 and 10 respectively and S1 and S2 to 16 and 2, to speed up the training time. As well, the last convolutional and average pool layers were reconfigured to produce a vector of 512 elements which represent the embeddings; this topic is explained extensively in the next section.

It is worth saying that this block acts as independently as the previous block, which expects an image-like audio representation tensor to train through its layer, hence this HLF can be replaced with any other computational vision CNN. The experimentation of using different HLF blocks is explained in following sections.

4.3.3 Classifier

The last layers of the VGG architecture were modified to implement a classifier layer group that outputs the number of classes needed to label. This last layer group is the last connected module of the E2E topology and can also be modified depending on the HLF configuration we are using.

There exist two main classifier configurations created for AemNet which are described next:

AclNet-Classifier: Used in VGG-ish architecture using standard convolution operations. Consists of a convolutional layer with a dropout of 0.2 that outputs a vector of $1 \; x \; C$; where C is the number of classes. Followed by a global average pool layer which provides the mean value of each channel to finally provide a decisive label to classify. When the standard LLF and HLF blocks are joined with AclNet-Classifier, the architecture is referred to as Aemnet.

Dataset	Num of clips	Clip duration (seconds)	Folds
DCASE2013 [4]	200	30	5
UrbanSound8K [9]	8732	≤4	10
ESC50 [8]	2000	5	5

Table 1. Datasets comparison used in the analysis of AemNet.

AclNetX-Classifier: Used in VGG-ish with DWSC operations, consists of a dropout operation to reduce overfitting and a fully connected layer with linear activation function. Unlike the previous architecture, if this classifier is used along the standard LLF and HLF blocks, the architecture is referred to as Aemnet-DW.
A softmax layer is used as activation function at the output of the AemNet to present normalized output values.

4.4. Description of datasets evaluated.

To give a proper assessment of the AemNet against other E2E solutions, three publicly available datasets commonly used to benchmark ASEC models were used: DCASE2013 [4], ESC-50 [8] and Urban Sounds8K [9].

4.4.1. DCASE 2013

Stowell et al [39] aimed to frame a general-purpose machine listening tasks to benchmark the SOTA and reinforce the research community on the ASEC domain. Because of this, a research challenge was organized under the auspices of the IEEE Audio and Acoustic Signal Processing Technical Committee: the DCASE challenge. This addressed two main objectives: recognizing the general environment or scene and detecting and classifying events within them [39].

On the DCASE 2013 edition, two main tasks were released regarding the dataset collected in [39]. Task 1 described a scene classification problem of identifying and classifying acoustic scenes and soundscapes.

The dataset stores 30-second audio files (WAV, stereo, 44.1 kHz, 16-bit) using binaural headphones around London at various times in 2012 [4]. Locations were selected to represent instances of the following 10 classes: busy street, quiet street, park, open-air market, bus, subway-train, restaurant, shop/supermarket, office, and subway station.

The submitted algorithms were evaluated with a 5-fold stratified cross validation. The metrics to evaluate were the accuracy, standard deviation, and a confusion matrix for each report. It is fair to mention that on this 2013 edition, the best performance models used a MFCC approach along a SVM to classify this task, showing how the ASEC approaches have changed on recent years moving to CNN-oriented audio applications.

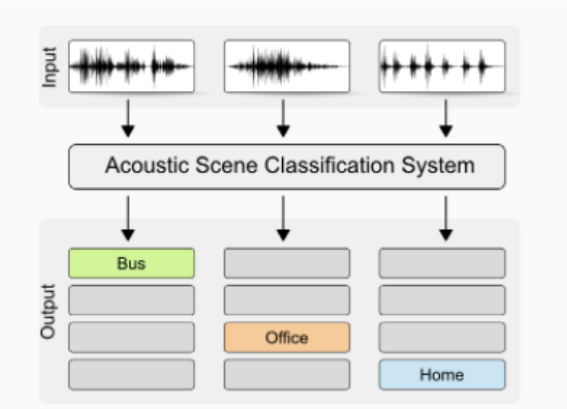


Figure 13. Overview of an ASEC system [4].

4.4.2. ESC-50

Piczak et all [8] facilitates open research with the Environmental Sound Classification (ESC) dataset by contributing with a public available dataset of environmental recordings and presenting a comparison between a machine learning approach versus a human accuracy estimate.

The audio samples were reconverted to a unified format (44.1 kHz, single channel) manually extracted from public recordings gathered by the Freesound.org project [40]. The dataset was arranged into 5 cross-validation folds, ensuring that clips from the same initial source are always contained in a single fold.

The ESC-50 dataset consists of 2000 labeled environmental sounds, 40 clips per class, keeping an equal class balance among them. These are grouped in 5 major categories, each category stores 10 classes. Table 2 stores the 5 domain main classes and the 10 classes under each.

Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

Table 2. ESC-50 disambiguation of 50 classes [41].

4.4.3. UrbanSound8K

Salomon et all [42] wanted to address one of the main challenges to urban sound research when it comes to labeled audio data, hence the creation of UrbanSound, a dataset of 10 low-level urban sound classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. Excepting "children playing" and "gunshot", all other classes were selected as they are addressed as urban noise complaints. The collection resulted in a total of 3075 labeled occurrences, bringing a total of 18.5 hours of labeled audio segments.

Since sound source identification is also an element of research, [42] created an additional subset of 4-seconds audio snippets named UrbanSound8K. It sets a maximum duration limit of 4 seconds, and segments longer than 4s are sliced using a sliding window with a hop size of 2 s. A limit of 1000 slices per class is set, to promote class distribution balance, resulting in 8732 labeled slices (8.75hours), hence the 8K suffix.

All urban sounds are in WAV format. The sampling rate, bit depth, and number of channels are the same as the original file uploaded to Freesound and may vary from file to file [9].

One strong recommendation that this dataset states is to use the predefined 10-fold cross validation splits, to be completely comparable to previous results in literature, which is the same approach we use to evaluate the robustness of AemNet and its respective experiments.

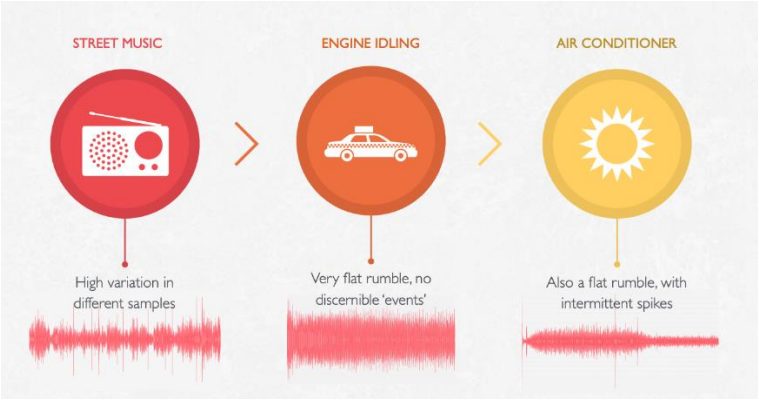


Figure 14. Examples of urban sounds that compose the UrbanSound dataset [43].

4.5 High-level features experimentation and analysis

Based on AemNet sand its multiple configurations described in the previous sections, we defined a series of experiments to evaluate the robustness of deep learning techniques such as depth-wise convolution and transfer learning over three different HLF topologies: a) VGG-ish topology with standard convolution, b) VGG-ish with depth-wise-convolution and finally c) ResNet18. To achieve the best performance for each topology, pretrained models were loaded and used for all topologies.

For pre-training purposes, the raw audio from dataset was downsampled to 16 KHz. Random audio clips were selected and used in the mini batches in the training stage, whereas complete standardized audio clips were used for validation inference.

For topologies A) and B) we use the knowledge distillation pretrained model, whereas topology C) used a transfer learning approach both from a ResNet18 topology pretrained with AudioSet. Notice how these two approaches are different from each other. Details on each pretraining are described below and illustrated in Figure 15.

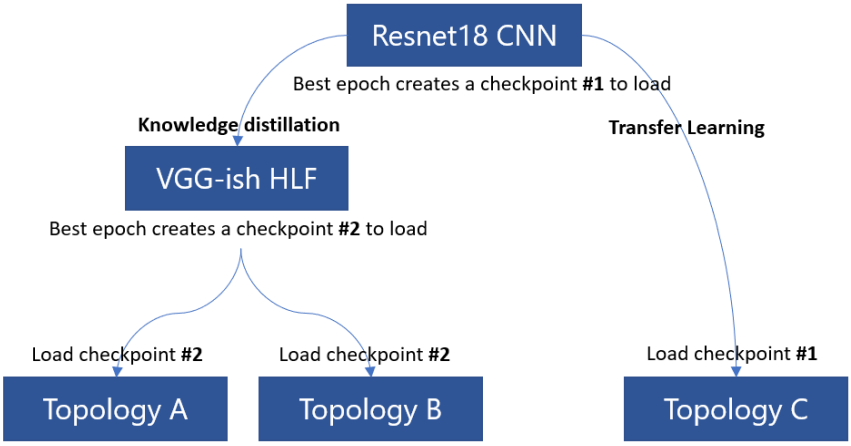


Figure 15. Representation of each checkpoint source. Used transfer learning and knowledge distillation.

First, we need to define what a pretrained model is in this context: whenever we are training our CNN model, we save our model weights and optimizer whenever it gets a better metric, normally we save by best accuracy, but this metric may vary depending on the experiment and application. In this case, we are saving the best epoch of the entire training in a compressed model named "checkpoint.pt" (pt extension refers to a PyTorch model save). After the training and validation of the model are done, this checkpoint #1 includes all calculated weights of the topology layers at the best epoch of the training.

Naturally, if the reference is a ResNet18 CNN, all calculated weights would match only with the same exact topology. Instead, we can use the knowledge distillation technique to match as close as possible a different CNN topology to the pretrained one, hence creating checkpoint #2.

Checkpoint #1 loads an AemResNet audio embedding generator previously described. This model includes a ResNet18 topology in which Adam optimizer with a learning rate (LR) of 5×10^{-4} was used, weight decay of 1×10^{-8} , and a mini-batch size of 512 over 80 epochs. Cosine aligned learning rate schedule was used. This model led to 11,744,143 number of parameters, with a mean average precision (mAP) of 0.3690.

On the other hand, checkpoint #2 is the result of a previously trained knowledge distillation model of the ResNet18 configuration to match a VGG-ish HLF configuration. The VGG-ish CNN trained with an Adam optimizer with a learning rate of 5×10^{-4}, weight decay of 1×10^{-11}, and a mini-batch size of 256 over 60 epochs. Cosine aligned learning rate schedule was used. This model led to 2,967,130 number of parameters, with a mAP of 0.3486.

Using checkpoints #1 and #2 as described in Figure 15, we used accuracy as the metric to evaluate the performance of three different configurations of AemNet. Let us remember that AemNet is based on a VGG-ish HLF with a standard convolutional layer operation, Aemnet-DW is based on the same VGG-ish HLF with depth-wise convolutional layer operation and its own classifier, whereas AemResNet is based on a HLF-based ResNet18 architecture. We used DCASE2013 and ESC-50 datasets to evaluate these three approaches both in a scratch training against a loaded pretrained model performance.

Mean accuracy 5-fold cross-validation						
	Aemnet		Aemnet-DW		AemResnet	
Dataset	Scratch	Pretrained	Scratch	Pretrained	Scratch	Pretrained
DCASE2013	66.67%	75.33%	60.67%	87.00%	68.67%	87.33%
ESC-50	77.50%	83.10%	72.45%	92.90%	78.65%	94.05%

Table 3. Comparison of mean accuracy per AemNet approach.

These two datasets are divided into 5-fold-like cross validation with an 80/20 split ratio, using 5 different random seeds. Results in Table 3 show the average of those 5-fold performance divided by a pretrained model and a performance of the topology from scratch training (no checkpoint loaded).

Results show a clear advantage of using a pretrained model in our CNN for all cases. This is expected since transfer learning has been a technique in several works [44]. Addressing the results of these experiments based on Table 3, we can conclude the following:

AemResNet shows the best performance against both AemNet and AemNet-DW. The performance difference of the two VGG-ish models against ResNet18 can be explained due to the architecture topology properties [45] also reaffirmed by [46] on an evaluation of different computational vision CNNs of ResNet18 against different topologies. A pretrained model shows a greater performance against a model trained from scratch in all approaches. In cases such as AemNet-DW with DCASE2013 improvement is around ~27% in accuracy which shows the great potential of having a pretrained model for future references.

On and overall analysis, the best performance model was an AemResNet pretrained model approach for both datasets. As this is an e2e audio embedding generator we also want to look at the model complexity, normally computed by number of parameters or multiply-accumulate operations (MACs), which are closely related to the topology architecture defined in the HLF of each model. For these approaches, we used Pytorch framework to calculate both MACs and number of parameters for each model which resulted in Table 4.

e2e CNN	WM	Params
AemNet	2.0	14,158,514
AemNet-DW	2.0	2,722,429
AemResNet	NA	11,499,442

Table 4. Model complexity per AemNet architecture.

From Table 4 we can see how WM variable impacts only AemNet and AemNet-DW. WM indicates the multiplying factor that linearly scales the number of output channels for [5], [6] and is directly reflected in the number of MACs. On the other hand, AemResNet which constitutes a different ResNet18 HLF approach does not have a WM.

We can conclude from Table 3 that even though these 3 models are all based on audio embedding generator, we have a great difference among their respective HLF architectures and can be evaluated for different experimentation purposes, e.g., AemNet-DW can be used for minor complexity tasks with certain limitation of parameters for production/hardware purposes (fitting an entire model into a limited capacity drive). Also, AemResNet approach can take advantage of both AemNet and ResNet18 properties to end with a high-performance model with a lower complexity than AemNet, analyzed and developed for the first time in this work. AemNet is the object of study applied to different tasks and approaches described in following chapters.

4.6 Low-level features experimentation and analysis

Under this same research, we also experimented with the LLF segment of the audio embeddings generator, focused more on the kernel size and stride applied at the beginning of the AemNet topology. The LLF details are described in section 4.5.3.

We explored two main types of LLF configurations:

Configuration 1: A kernel size of 33 x 33 convolutional layer is chosen along a stride of 16.

Configuration 2: A kernel size of 9 x 9 convolutional layer is chosen along a stride of 4.

Dataset	Mean accuracy 5-folds	AemNet 1.0			AemNet-DW 1.0		
		Accuracy	F1-score	Time	Accuracy	F1-score	Time
DCASE 2013	LLF config 1 (s1-16, k1-33)	67.00%	66.07%	30 min	55.00%	53.29%	33 min
DCASE 2013	LLF config 2 (s1-4, k1-9)	67.00%	65.67%	31 min	58.33%	56.15%	33 min
ESC-50	LLF config 1 (s1-16, k1-33)	76.00%	75.31%	87 min	70.10%	68.73%	125 min
ESC-50	LLF config 2 (s1-4, k1-9)	79.70%	79.13%	105 min	71.05%	69.71%	143 min

Table 5. Results comparison of LLF configs #1 and #2 based on two datasets. Time is measured in minutes.

The experiments performed involved two different topologies: AemNet and AemNet-DW. In this case, each model had a WM of 1.0. We evaluated the performance of both LLF configurations by its accuracy, F1-score (based on precision and recall), and execution time in minutes.

Each topology was trained without a pretrained model, having both CNNs training from scratch helps to evaluate in a cleaner way the performance of both LLF configurations. Both AemNet and AemNet-DW used an Adam optimizer with a learning rate of 1×10^{-3}, weight decay of 2×10^{-4}, mixup data warmup of 0.05 and alpha of 0.1 and a batch size of 128 along 600 epochs.

It is very clear how configuration #2 achieves a better performance in accuracy and F1-score (average of precision and recall) compared to configuration #1, at the cost of longer training time, more clearly seen in ESC-50 database with a ~20-minute difference between configurations in both topologies. We also notice how DCASE2013 database show very few improvements in execution time, this can be explained

to the number of data samples in each dataset, which may be an indication that the execution time would increase on a significant number of samples.

In conclusion, this evaluation shows that the LLF configuration #2 has a better performance and should be the main configuration for the following tasks experimentation. Also, notice how execution time is not a main factor on deciding between both configurations, since we are training on a discrete environment which can later be implemented in hardware, we are trying to provide the best audio classification performance, regardless of how long it takes.

5. DCASE 2021

Summary: *In this chapter, we explore the implementation of AemNet into a target application based on ASEC such as the DCASE2021 challenge. Section 5.1 provides an introduction and motivation of AemNet and its variants to participate into the DCASE2021 challenge. Section 5.2 provides the main objective of the DCASE challenges and its objectives to the research community. It also provides an insight to the dataset of Task 1A and details on its labels and cross-validation splits. Section 5.3 describes the methodology followed to train and validate AemNet, also explaining the process of two main optimization techniques to decrease the memory size of our approach by maintaining a competitive performance above the baseline. Finally, section 5.5 presents the results of our experiments by comparing AemNet metrics performance on each optimization stage. The submission released to the DCASE2021 challenge was ranked as the 24th best out of 30 submissions sorted by accuracy.*

5.1 Introduction

For the 2021 Detection and Classification of Acoustic Scenes and Events challenge (DCASE2021), acoustic data were provided to solve different acoustic related tasks. We aimed our proposal to solve Task 1A, which refers to building a classifying system for urban environment scenes under a certain memory size restriction. We are using AemNet-DW as a base model to implement optimization techniques with the purpose of lowering the memory size of the CNN.

5.2 DCASE 2021 Task 1A description

The DCASE2021 data was provided to solve different acoustic related tasks. Historically, the members of the organizing committee challenge participants to present their work on different types of tasks. Each task can have subtasks such as the one we participated on: Task 1, subtask A (Task 1A).

AemNet and AclNet are quite familiar with the DCASE challenge [47], so we decided to continue the

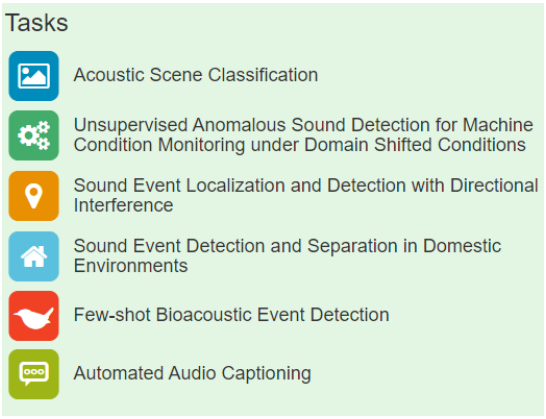


Figure 16. DCASE challenge tasks categories [4].

work on the 2021 edition with the very well-known Task 1A. Its goal is to classify a test recording urban sound into one of 10 predefined classes corresponding to environment scenes. Task 1A consists of classifying data from multiple devices (real and simulated) across several different devices on low-complexity models [4]. We focused our solution to comply with the system complexity requirements stated by the Task 1A.

Model complexity limit is 128 KB for non-zero parameters, meaning parameters data type have a direct impact e.g. 32768 parameters using float32 = 128 KB. There is no hard recommendation on which method to minimize the model size.

This challenge's dataset consists of 10-second audio recordings obtained in 10 different acoustic scenes from 12 major European cities, grouped in three major classes: airport, bus, metro, metro station, park, public square, shopping mall, street pedestrian, street traffic, and tram [48]. This acoustic dataset comprises audio signals at 44.1 kHz of sampling rate in 24-bit resolution.

The challenge suggests the usage of a 1-fold arrangement for development as part of this task, with 70% for training and 30% for testing.

5.3 Methodology description and techniques used

Following the guidelines provided by the challenge in Task 1A, we experimented with one low-memory implementation pipeline of an audio CNN architecture through two optimization techniques: pruning of models using the lottery ticket hypothesis approach, followed by a FP32 to INT8 quantization. These techniques to lower the complexity of deep learning models are explained in section 3.3.

We based our topology on the settings corresponding to the AemNet audio embeddings generator work described in [6], using a width multiplier of 0.5, and conventional depth-wise convolution layers. This AemNet adaptation, which we refer to as AemNet-DW, was pre-trained with Google AudioSet [49] to generate a vector of 512 audio embeddings that are sent to a fully connected layer classifier built with ReLU activation functions in a transfer learning manner.

Our preprocessing stage consisted of taking raw audio data from the Task 1A dataset downsampled to 16 kHz and fed to the pretrained e2e CNN, where the generated embeddings were used to train the classifier.

To increase the robustness of the training process, we also used different audio data augmentation techniques commonly used in audio processing, such as random noise addition, random cropping of 1-second of the audio signal, and random gain variation, together with the widely used mixup data augmentation technique [20]. During the training, acoustic data were randomly selected to form mini-batches of training clips. At testing time, we run the inference over each complete audio file.

5.4 Experiments performed

A search of the optimal parameters was performed by experimenting with several hyperparameters such as learning rate, learning rate decay, weight decay and dropout rate. We also explored the flexibility of the audio embeddings to dynamically adapt to the application by allowing to adjust its weights during training as a percentage of the classifier layer learning rate.

5.4.1 Pruning phase

Using AemNet-DW as base model, we ended up with a CNN formed by 319,093 parameters. This base model is initially trained and afterwards pruned by 60% through the lottery ticket hypothesis [50] to have a final model of 127,637 parameters. We also experimented removing 60% of the parameters by a typical pruning scheme, i.e., remove the parameters that are contributing less to the model's classification behavior.

The lottery ticket hypothesis comes into place when, after identifying the post-pruning weights, a new training process is carried out with the original randomly initialized weights values assigned at the initial pre-training stage.

5.4.2 INT8 quantization

The resulting pruned e2e CNN constitutes a single-precision floating-point format (FP32) base model with 127,637 trainable parameters, which yields into 498.58 KB of memory size, clearly above the 128 KB restriction in the challenge. To decrease the memory size of this model, we applied a straight FP32-to-

INT8 training-aware quantization based on the methodology described in [51], using the available tool accessible in [52], that results in an optimized 124.64 KB e2e audio classification CNN model.

5.5 Results and discussion

Table 6 shows the performance initially obtained by AemNet-DW before any optimization means to lower its complexity. It is observed that this proposal exceeds greatly the DCASE2021 baseline defined on Task 1A challenge but fails the memory KB restriction. Hence, the following results of AemNet-DW after the lottery ticket (LT) 60% pruning show a reduction near to 2.5X of memory size by giving up nearly 2.7% of accuracy. Still after pruning, the 498.58KB is still above the maximum restriction the Task 1A states. Over the same AemNet-DW LT the quantization we experimented with, shows a memory reduction of 10X against the original proposal, trading off the accuracy loss of 4.99%.

In conclusion, by exploring transfer learning, pruning, and quantization to execute neural networks model optimization, we were able to successfully construct an e2e audio classification deep learning-based model that achieves 56.50% accuracy performance on the DCASE2021 testing dataset, with 124.64 KB of memory size.

Model	Accuracy	Params	Memory KB	Memory reduction	Format
DCASE2021 Baseline	46.40%	-	90.00	-	-
AemNet-DW	61.49%	319,093	1246.45	0	FP32
AemNet-DW LT	58.73%	127,637	498.58	2.5X	FP32
AemNet-DW INT8	56.50%	127,637	124.64	10X	INT8

Table 6. Experimental testing results obtained for our DCASE2021 submission.

6. ICBHI 2017 CHALLENGE

Summary: *In this chapter we present the use of an e2e deep learning based pre-trained audio embeddings generator and apply it to the purpose of classification of respiration sounds. With this approach, there is no need to pre-compute spectral representations, e.g., MFCC or filter banks, since the classification model uses raw audio as the input. We make use of the audio embeddings generator described in chapter IV, which classifies the type of respiration sound as defined in the IEEE International Conference on Biomedical and Health Informatics (ICBHI) Scientific Challenge released in 2017. Section 6.1 gives a brief introduction of respiration sound classification background. Section 6.2 presents the ICBHI Scientific Challenge description, purpose and metrics used to evaluate results. Section 6.3 provides an in-depth explanation of the ICBHI dataset composition and digital format. Section 6.4 breaks down into two subsections the pretraining stage of the audio embeddings generator and the experiments performed to tune the model to reach an optimal classification based on the F1-score obtained. Transfer learning was used to train an audio classifier for sounds of respiratory cycles as defined in the ICBHI 2017 challenge. Finally, section 6.5 shows that this e2e model represents a viable alternative to common spectral-based classifiers, that are able to achieve a SOTA performance. Overall, this chapter describes the paper publication named "Classification of Respiration Sounds Using Deep Pre-Trained Audio Embeddings" delivered on the IEEE Latin American Conference on Computational Intelligence on November 2nd, 2021.*

6.1 Introduction

The direct analysis of respiration sounds by health professionals provide significant insight to build a clinical assessment of different health conditions, e.g., pneumonia, bronchitis, asthma, etc. The typical method carried out for this purpose is for a patient to attend the doctor's office, where an auscultation takes place over the chest and the back of the patient looking for characteristic sounds in the respiration through a stethoscope. This method is only effective when the patient attends physically to the evaluation, with the major constraint that is not subject to objective monitoring over extended periods of time.

Different automatic and novel methods have been proposed for sound of respiration cycle classification, i.e., to identify normal from abnormal respiration that can be associated to different medical conditions. The recent methods reported in the literature typically follow an implementation based on deep learning technologies. In [12] a recurrent neural network is used for lung sound identification. An ensemble of two large deep learning models is used in [13] to enhance performance in the prediction of respiratory anomalies. A deep learning architecture was also used in [14] to detect a possible lung disease with the classification of respiratory anomalies. A VGG16 convolutional neural network was proposed in [15] for automatic classification of respiratory sounds. Another example in the use of deep learning is the ensemble of convolutional neural networks proposed in [16] for lung sound classification.

This work proposes the use of an e2e deep learning-based model to identify normal respiration cycles from those showing presence of wheezes, crackles, or both.

6.2 The ICBHI Scientific Challenge

The ICBHI 2017 Scientific Challenge dataset [53] is an ensemble of 920 recordings from 126 subjects, resulting in 6,898 respiration cycles over 5.5 hours; collected independently by two research teams in two different countries over several years. These two research teams were the School of Health Sciences, University of Aveiro (ESSUA) research team and the Aristotle University of Thessaloniki (AUTH). The ESSUA research team recordings were collected at Respiratory Research and Rehabilitation Laboratory (Lab3R), ESSUA and at Hospital Infante D. Pedro, Aveiro. These respiratory sounds followed the computerized respiratory sounds analysis guidelines for short-term acquisitions, collecting sounds from seven chest locations: trachea; left and right anterior, posterior, and lateral. These sounds were collected in clinical and non-clinical settings [53]. On the other hand, the AUTH research team acquired respiratory sounds at the Papanikolaou General Hospital, Thessaloniki and at the General Hospital of Imathia (Health Unit of Naousa), Greece. In this team's research, sounds were collected sequentially from six chest locations, on adult and elderly patients that had chronic obstructive pulmonary disease (COPD) with comorbidities (e.g., heart failure, diabetes, hypertension). On the ESSUA collected data, two respiratory physiotherapists and one medical doctor annotated the sound files in terms of presence/absence of adventitious sounds and identification of breathing phases [53]. For the AUTH acquired data, two specialized pulmonologists and one cardiologist performed sound annotations. The sounds discriminated where the following: normal (respiratory sound), fine crackles, coarse crackles, wheezing, speech, cough, and artifact [53].

The first edition of the ICBHI Scientific Challenge consisted in classify, for each respiratory cycle of a short recording (10-90s), whether the respiratory cycle contains crackles, wheezes, or crackles and wheezes.

As evaluation metrics, there were two performance measures: average score (AS) and harmonic score (HS). AS is the average of sensitivity and specificity, while HS is the harmonic mean of both. Each metric calculation is described in Equations 6 and 7.

$$AS = \frac{sensitivity + specificity}{2}$$

Equation 6. Average score calculation.

$$HS = \frac{(2 \times sensitivity \times specificity)}{(sensitivity + specificity)}$$

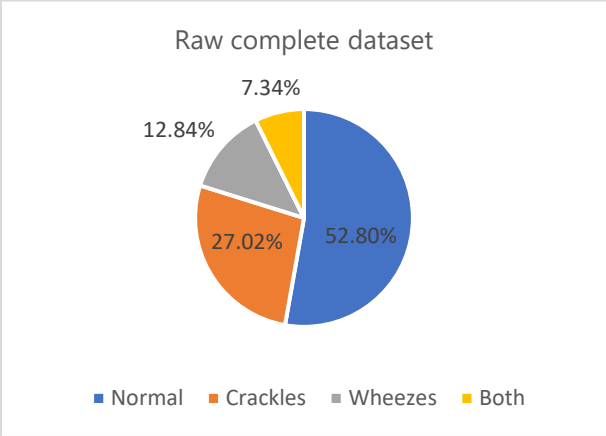
Equation 7. Harmonic score calculation.

6.3 Data exploration and processing

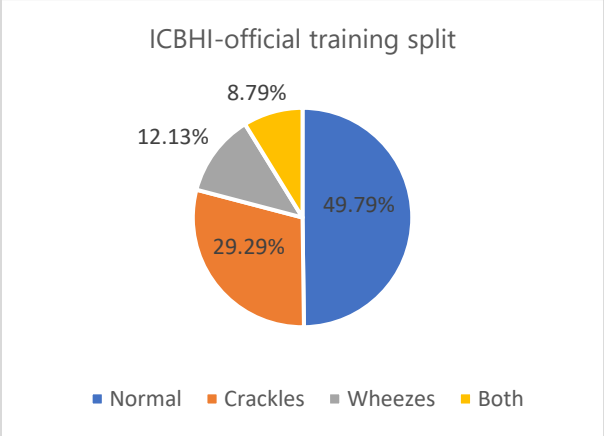
The ICBHI 2017 challenge dataset contains 920 WAV audio files: these vary from 10-90 seconds sampled at different frequencies. 90 samples at 4 kHz, 6 at 10 kHz, and 825 at 44.1 kHz. In total, the number of patients is 126: 77 adults and 49 children [53]. For experimentation in this work, audio data were processed to be single channel with 16 kHz sampling rate, a 16-bit resolution, and standardized in amplitude.

The training and validation official split defined by the ICBHI 2017 challenge was strictly followed for cross-validation analysis of results and viability, and direct comparison with other published works, where 60% of the data is used for training, and 40% for validation (60/40 split) to develop our first set of cross-validation folds. Additionally, as observed in other published works, we developed a second set of 5 custom folds, which were defined randomly with a split 80% of the data for training, and 20% for validation (80/20 split).

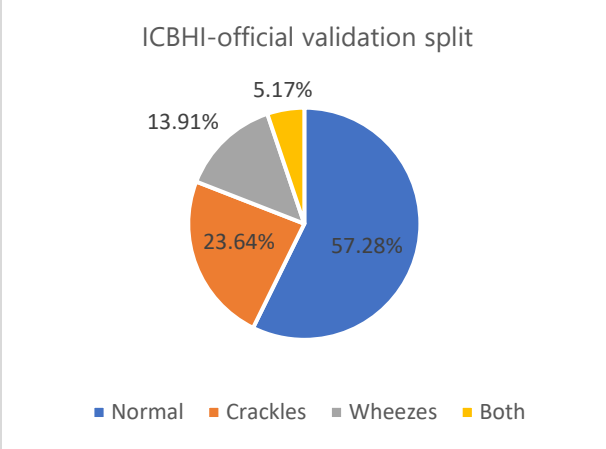
To have a broader understanding on the dataset we are using, we created several pie charts to visualize the components of each training and validations splits on a per-class basis. The analysis on the data is shown on figures 17 and 18.



(a)



(b)



(c)

Figure 17. Pie charts of ICBHI Scientific Challenge dataset. A) Distribution of complete dataset. B) Training split distribution. C) Validation split distribution.

From Figure 17(a) we observe how the 4 classes are clearly unbalanced, more than 50% of the dataset corresponds to a single class ("normal"), whereas "both" (wheezes and crackles detection) corresponds barely to 7% of the total samples collection.

The same class balance is kept on the official split proposed by the ICBHI challenge [54] in which 60% of the complete dataset is used for training and 40% for validation.

From the training samples in Figure 17(b), almost 50% correspond to the "normal" class, while the rest of the classes keep a ±2% from the original class balance.

Validation samples in Figure 17(c) show an increase of "normal" class up to 57%, reducing "crackles" and "both" classes; "wheezes" class is 2% increased in this new data split.

The purpose of this analysis is to understand the visible class unbalance problem which might directly affect the development of the deep learning approach to solve a 4-label classification task. To know the precise percentage on a per-class basis results very helpful when implementing class imbalance mitigation techniques such as the implementation of different loss functions that counter the weight of low percentage classes.

On the other hand, Figure 18 shows the same class-balance analysis made for the 80/20 custom split. The class-balance in both training and validation datasets is kept mostly uniform in all 5 cross-validation folds.

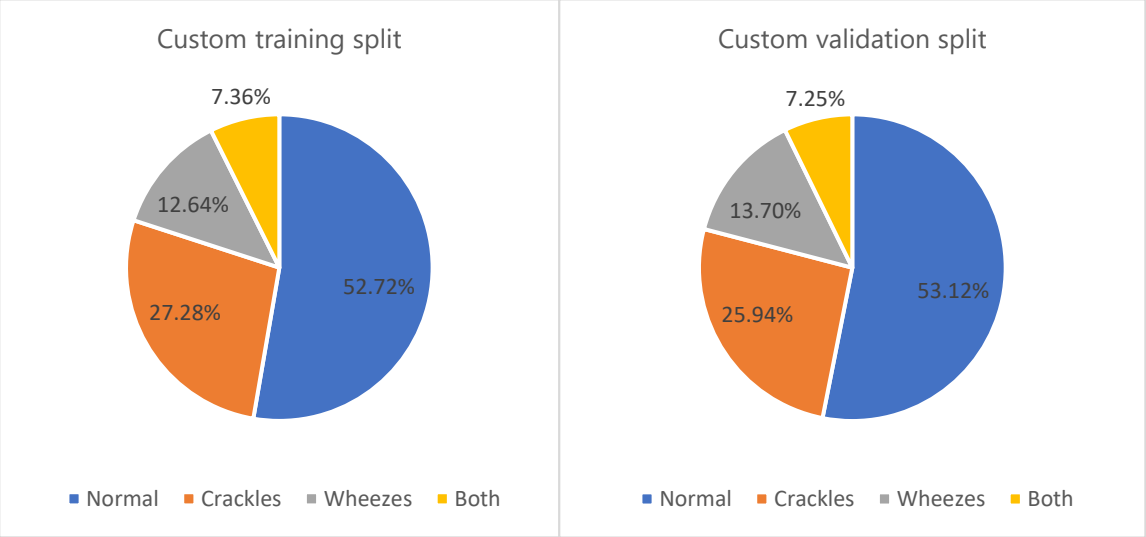


Figure 18. Distribution of the custom-made folds for the ICBHI dataset (80/20).

6.4 Audio embedding generator explanation

The audio embeddings generator described in Section 4 is the baseline reference into the approach followed in this new section, where a pre-trained model efficiently generated high quality and robust audio embeddings and was successfully implemented in target applications over different benchmark datasets. The proposed e2e CNN architecture discussed in this work comprises three main blocks: the LLF block, the HLF block that constitute the audio embeddings generator, and a final classification block trained with the embeddings generated by the previous two blocks.

The time-domain waveform is input to the LLF block, that produces an output of 128 channels at frame rate of 10ms after an added max-pool layer, from a 16 kHz raw audio input. We observed heuristically that 128 channels provide a robust performance on audio classification, while a 10 ms window is used to have a good time resolution of the signal. On the sampling frequency 16 kHz is the chosen trade-off value between a good quality audio sample and a low complexity model. For 1 second of audio input, the LLF produces an output tensor of dimension [128, 1, 100], these convolutional layers act as a trainable equivalent of a spectral filter bank feature extraction.

The HLF block follows a CNN topology like the ones commonly used for image classification. For this work, we experimented with a ResNet topology of 18 layers.

The final block of the e2e CNN acts as a classifier and comprises a dropout (DO) layer to reduce overfitting and one fully connected layer with linear activation functions. The input to this classification block is the

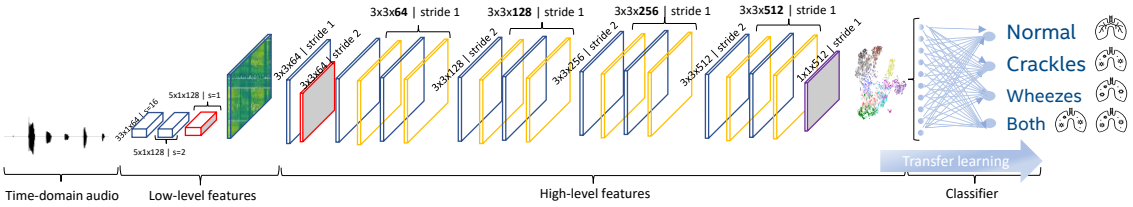


Figure 19. AemResNet proposed to solve the ICBHI Scientific Challenge.

output of the convolutional layer averaged pooled to produce a vector of fixed size representing 512 audio embeddings that represent the input to the classifier.

6.4.1 Pretraining stage

The LLF and the HLF blocks that constitute the audio embedding generator are pre-trained using AudioSet, a large dataset of manually annotated audio events released by Google [49].

Before using this embedding generator model for a specific classification application, the final classification block is removed, i.e., the fully connected layer, resulting in a 512-dimensional audio embeddings representation as the output. Data augmentation techniques typically used in audio processing were also used: random noise addition, random cropping of a segment of the sample audio signal, and random gain variation. Additionally, the widely used mixup data augmentation technique [20] was used. During the training process, random audio clips were selected and used in mini batches.

Adam optimizer with a learning rate of 5×10^{-4} was used, with a weight decay of 1×10^{-8} , and a mini-batch size of 512 over 80 epochs. Cosine aligned learning rate schedule was used. This audio embedding was trained using the available unbalanced set and validated with the evaluation set for the 527 classes. All experimentation was executed using the Pytorch framework [36]. This audio embedding generator model resulted in 11,744,143 number of trainable parameters, with a mAP of 0.3690.

6.4.2 Experimentation stage

The training strategy followed for this application is similar to the pre-training of the audio embedding generator: Adam optimizer was used with a learning rate of 1×10^{-3}, weight decay of 2×10^{-4}, mini-batch size of 64 over 350 epochs, cosine aligned learning rate schedule, and warm up of 35 epochs before mixup. It is important to notice that the ICBHI 2017 dataset presents a highly imbalanced number of samples per type of respiration cycles, i.e., there is a significantly larger number of normal respiration samples compared to the other 3 not normal types of respiration. Due to this issue, a focal loss approach was used in the loss function [55], resulting in a more efficient training process. We explored the flexibility of the audio embeddings generator to dynamically adapt to respiratory classification by allowing the adjustment of its weights during training.

For both the official and custom split experiments, we focused on the validation dataset class ratio, since the deep learning model outcome was based on the cross-validation best model after the training stage. Given that premise, we calculated the weights based on the percentage of each class, as shown in Table 7. These newly formed weights will be the input to the loss function, which was explicitly chosen to handle the class imbalance present in the ICBHI dataset.

Class	Validation split %	Calculated weight (1 - validation %)
Normal	0.5728	**0.4272**
Crackles	0.2364	**0.7636**
Wheezes	0.1391	**0.8609**
Both	0.0517	**0.9483**

Table 7. Weight calculation per class on the ICBHI dataset.

The focal loss function described in Section X, was the loss function approach to mitigate the class imbalance problem in the ICBHI dataset. As described before, the components of gamma and alpha values correspond to an exponential value of loss and a weighted vector value based on each class ratio. The alpha value is then a vector formed by each class weights as described in Equation 8. Where the suffix "w" corresponds to the calculated weight of each class stated in Table 7.

$$\alpha = [Normal_w + Crackles_w + Wheezes_w + Both_w]$$

Equation 8. Alpha vector calculation based on each weighted class.

As for the gamma (γ) value chosen for this task, we performed several experiments to find the best value according to the alpha (α) vector already defined. Several experimentations based primarily on seeking the best F1-score pointed that $\gamma = 0.5$ was the optimal value for both official and custom splits. Some other experimental values were $\gamma \epsilon [0,2]$ based on the works described on [55] in which class imbalance is also mitigated by finding the optimal alpha and gamma values.

Different learning rate values for the LLF and the HLF were utilized as a percentage of the fully connected layer learning rate; experiments to find the optimal percentage were evaluated sweeping through different values from 10% to 100% in increments of 10%.

We have observed experimentally this behavior in other applications and hypothesized that there has to be an optimal learning rate percentage for the embeddings model, i.e. a smaller learning rate than the one used in the last fully connected layer; this is consistent with the differential learning rates observed in other works [56], [57], where it is explained that for pre-trained models there is not a need to change significantly

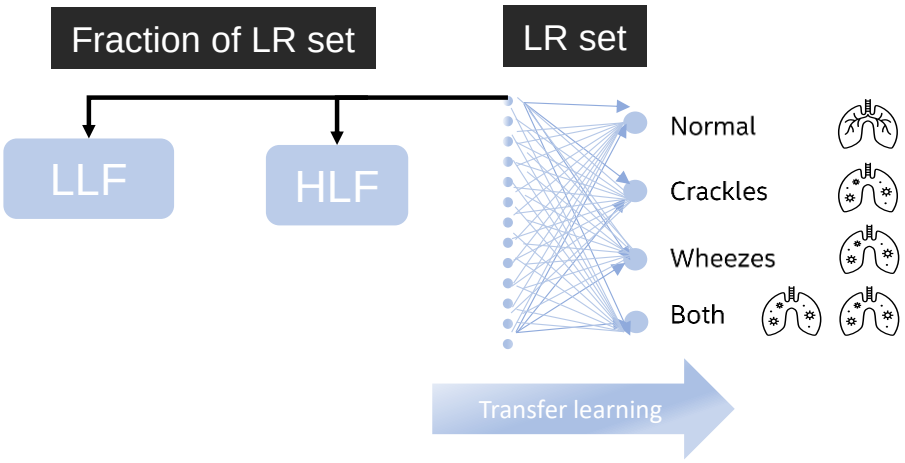


Figure 20. Usage of a learning rate fraction for LLF and HLF blocks on the ICBHI classification.

the front-end layers at the same rate as the last layers, since these have already learned to do a good generalization over rough initial features.

The model obtained after training the respiration sound classifier using the pre-trained audio embedding generator resulted in 11,475,844, which is 2.3% less parameters against the pretrained model described above, due to a smaller classifier block with only 4 outputs. Additionally, the number of MACs results in 1.84×10^9.

For the assessment of the performance of the proposed e2e CNN model, Sensitivity, Specificity, and Score metrics were followed as explicitly stated for the ICBHI 2017 challenge.

6.5 Results and discussion

All experimental results over the official and custom validation splits are displayed in this section. Table 8 shows the impact over the final score validation metric when updating the weights of the audio embeddings generator as a percentage of the learning rate used by the classification block. This shows the importance of the pretraining stage, updating flexibility during training applied to the pre-trained audio embeddings during the transfer learning process, to have a more robust model.

The last row in Table 8 also shows the performance of an e2e CNN models trained without the use of any pre-trained embeddings; over the official split the score results are 0.535 Vs. 0.561, and over the custom split the score results are 0.684 Vs. 0.772 for scratch training and transfer learning training, respectively. These results evaluate the metrics for the pre-trained deep audio embeddings implemented to the task of respiratory sound classification.

	Official split			**Custom split**		
LR %	***Sensitivity***	***Specificity***	***Score***	***Sensitivity***	***Specificity***	***Score***
10%	0.289	0.799	0.544	**0.678**	**0.865**	**0.772**
20%	0.311	0.767	0.539	0.644	0.874	0.759
30%	0.298	0.786	0.542	0.645	0.868	0.757
40%	0.289	0.804	0.547	0.641	0.845	0.743
50%	0.396	0.706	0.551	0.647	0.850	0.749
60%	**0.251**	**0.870**	**0.561**	0.622	0.859	0.740
70%	0.382	0.710	0.546	0.606	0.863	0.735
80%	0.342	0.768	0.555	0.603	0.865	0.734
90%	0.284	0.790	0.537	0.596	0.865	0.731
100%	0.387	0.729	0.558	0.594	0.855	0.725
No TL	0.304	0.765	0.535	0.540	0.828	0.684

Table 8. Experimental results obtained over the official and custom split on the ICBHI Scientific Challenge.

	Official split			Custom split		
Model	***Sensitivity***	***Specificity***	***Score***	***Sensitivity***	***Specificity***	***Score***
BLSTMDAE[12]	-	-	-	0.720	0.920	0.760
VGG16[15]	0.280	0.810	0.540	-	-	-
C-DNN[14]	0.260	0.680	0.470	0.680	0.900	0.790
SE-8Cycle[16]	-	-	-	0.694	0.873	0.784
Ensemble[13]	-	-	-	0.730	0.860	0.800
Our proposal	**0.251**	**0.870**	**0.561**	**0.678**	**0.865**	**0.772**

Table 9. Comparison of our proposed AemNet adaptation with other SOTA methods.

Table 9 shows a comparison of our e2e CNN with the SOTA results reported in recent published works [12]–[16]. Not all these works present their results on both the official and a custom split but are considered here to present a broader scope on how the proposed e2e CNN model compares to these. In these results it can be observed that generally, the official split tends to have smaller score values than the custom splits; this can be explained by the fact that the official split presents a more challenging problem with its 60/40 data distribution for training and validation. In this case, the e2e CNN used in this work achieved the highest score of 0.561 when compared to [14], [15], which is to the best of our knowledge above the SOTA reported in the existing literature. Comparison over the custom split is not as straight forward as with the official split, since the use of random splits and number of folds for cross-validation will differ for each work but gives a clear idea that the e2e CNN is performing competitively.

A factor that might impact the difference in score results on the custom splits could be explained by the difference in the deep learning model complexity, e.g., 11.4 million parameters of the e2e CNN Vs. 39 million parameters in [16]. Ongoing research for this type of acoustic applications consist of the optimization of the HLF block; the ResNet network used here contributes significantly to the 11.4 million parameters of the final model; we are addressing the issue of reducing the number of the parameters in the model by exploring different topologies that could reduce this number, while maintaining the same or better performance.

The experimental results presented in this work prove that an e2e deep learning approach can be successfully used to classify different sounds of respiration cycles such as normal, wheezes, crackles, or both. The main characteristic of this approach is that it avoids the need of additional pre-processing steps for feature extraction, thus facilitating its portability into an e2e inference engine. With pre-trained deep audio embeddings, a respiration sounds classifier model was built through transfer learning that achieved a SOTA Score of 0.561 over the official split defined for the ICBHI 2017 challenge dataset. Additionally, this model achieved a competitive 0.772 score over custom defined 5-fold random splits.

7. COVID-19 CLASSIFICATION

Summary: *Due to the COVID-19 worldwide pandemic situation, automatic audio classification research has been of interest for analysis of respiratory sounds. Several deep learning approaches have shown promising performance for distinguishing COVID-19 in respiratory cycles. We explored the usage of transfer learning from a pre-trained e2e deep-learning based audio embeddings generator named AemResNet, applied to the classification of respiration and coughing sounds into healthy or COVID-19. Section 7.1 provides an introductive background of the COVID-19 pandemic and an overview of the respiratory classification tasks around it. Section 7.2 describes the composition of Cambridge Crowdsourced dataset on respiratory sounds collected to aid diagnosis of COVID-19. Section 7.3 describes our work focus split into 3 experimental tasks: 1) detection of COVID-19 from a combination of breath and cough sounds, 2) detection of COVID-19 from breath sounds only, and 3) detection of COVID-19 from cough sounds only. Section 7.4 describes how the experimental results obtained over this respiratory dataset show that a pre-trained audio embedding generator achieves competitive performance compared to the recent published SOTA. Finally, section 7.5 shows the results obtained in this research regarding Aemnet aimed to COVID-19 classification based on respiratory sounds. This chapter describes the publication named "Detection of COVID-19 in Respiratory Sounds using End-to-End Deep Audio Embeddings" presented on the Call for Papers contest celebrated in the 4th International Student Conference in Latin America presented through the IEEE EMBS Chapter in Guadalajara, on November 6th, 2021. In this contest, this paper was awarded 1st place under the "Graduate" category.*

7.1 Introduction

Coronavirus (COVID-19) is an infectious disease caused by the severe acute respiratory syndrome coronavirus (SARS-CoV-2) virus [58] first detected in Wuhan, China in 2019. On March 2020th, COVID-19 was declared a pandemic by the World Health Organization (WHO). Most people experience moderate respiratory symptoms such as: coughing, fever, and shortness of breath. The first time this novel virus was detected was within a cluster of patients with pneumonia of unknown cause. According to the WHO, 15% of overall COVID-19 patients present a severe pneumonia [58], which is auscultated by a physician listening respiratory sounds through breath and cough. The main purpose of recording respiratory sounds is to find a weakness of hypoventilation which can lead to diagnose the patient illness.

Nowadays, there are several methods proposed to distinguish the respiratory cycles, e.g., identifying a shortness of breath mostly related to pneumonia. The implementation of the most recent approaches on respiratory sound classification includes a recurrent neural network used for lung sound classification in [12], two deep learning ensemble model aimed to predict respiratory anomalies is proposed in [13], a deep learning architecture to detect possible lung disease in presented in [14] by classifying respiratory anomalies.

COVID-19 aimed works have taken part on the research community. The work reported in [18] shows the efforts on the creation of an Android application aimed to collect different sounds from patients such as breath, cough, and speech; with this, they have created a dataset containing more than 459 samples from 378 patients through a crowdsourced methodology, named Cambridge Crowdsourced dataset. In this work, some ML techniques such as SVMs were used as the classifier for COVID-19 detection. In [59], the composition of residual network blocks is used to classify COVID-19 based on audio spectrograms and motivates to a comprehensive follow-up research. On [60], respiratory audio recordings are treated as a visual representation through two different spectrogram configurations and as raw audio, each of these samples are inputted into a CNN layer and the output is concatenated and ensembled to classify COVID-19. Overall, it can be observed how deep learning is currently leading the SOTA when it comes to audio classification for COVID-19.

7.2 Cambridge Crowdsourced dataset

The University of Cambridge launched an application in Android and on a website [61] in which participants are asked to fill demographics general information and symptoms check. The dataset comprises 459 cough and breath samples from 378 users from Web and Android applications until May 2020. These data were annotated by experts and the audio samples were carefully checked to guarantee the quality of the data that contains only cough and breathing.

The nature of this dataset is entirely crowdsourced, that means that the ground truth is what the users state in terms of symptoms and COVID-19 testing status. Also, the source of this datasets encourages to overcome challenges such as different phones and microphones in very different environments.

The data collection of this dataset is presented in [18], using a web-based app and an Android app. In both, the user is asked to input their age and gender as well as a brief medical history. Users then input symptoms and record respiratory sounds: they are asked to cough three times, to breathe deeply through their mouth

three to five times and to read a short sentence appearing on the screen three times. Finally, users are asked if they have been tested for COVID-19, and a location sample is gathered with permission [18].

Helped by a large media campaign orchestrated by the University, we were able to crowdsource data from many users. As of 22 May 2020, our dataset is composed of 4352 unique users collected from the web app and 2261 unique users collected from the Android app, comprising 4352 and 5634 samples respectively. Of these, 235 declared having tested positive for COVID-19: 64 in the web form and 171 in the Android app.

By May 2020, the Cambridge Crowdsourced dataset was composed of 4352 unique users collected from the web app and 2261 unique users from the Android app. The analysis for further feature extraction and classification in [18] was focused on a curated set of collected data and restricted to use coughs and breathing only.

The way the dataset was categorized is as follows:

a) **Non-COVID**: those users with a clean medical history, who had never smoked, had not tested positive for COVID- 19, and did not report any symptoms. 298 samples.
b) **Non-COVID with cough**: users who meet the same criteria as the *non-COVID* users but declared a cough as symptom; these provided 32 samples.
c) **Asthma with cough**: the users who have asthma, had not tested positive for COVID-19, and had a cough; these gave us 20 samples.

7.3 Tasks baseline description

To identify which audio modality (cough or breathing) contributes more to the classification performance, we repeated our experiments with three different audio inputs: only cough, only breathing, and combined.

In [18] the audio modality was also explored to identify which one contributes more to the classification performance and created tasks with three different audio inputs: cough, breath, and both. The binary classification tasks are described as follows:

• **Task 1**: Cough and breath sounds are used to distinguish users who have declared they tested positive for COVID-positive from users who have not declared a positive test for COVID-19, have a clean medical history, never smoked, have no symptom and were in countries where COVID-19 was not prevalent at the time. This task compared 66 users (282 samples or 32% of the audio samples) against 220 users (596 samples or 68% of the audio samples), respectively.

• **Task 2**: Cough sounds are used to distinguish users who have declared they tested positive for COVID-19 and have declared a cough as a symptom from users who have declared not to have tested positive for COVID-19, have a clean medical history, never smoked, were in countries where at the time COVID-19 was not prevalent and have a cough as a symptom. This task compared 23 users (54 samples which represented 63% of the audio samples) against 29 users (32 samples representing 37% of the audio samples), respectively.

• **Task 3**: Breath sounds are used to distinguish users who have declared they tested positive for COVID-19 and have declared a cough as a symptom (COVID-positive with cough), from users who have not declared to have tested positive for COVID-19, are from countries where at the time COVID-19 was not prevalent, have reported asthma in their medical history and have a cough as a symptom. This task compared 23 user (54 samples which represented 73% of the audio samples) and 18 users (20 samples representing 27% of the audio samples), respectively.

The training and test sets were created making sure that samples from the same user appear only in the training or test split.

7.4 Experimentation

The training strategy followed for this application is similar to the pre-training of the audio embedding generator: Adam optimizer was used with a learning rate of 1×10^{-3} for Task 1 and Task 2, Task 3 used 1×10^{-6}, weight decay of 1×10^{-8}, mini batch size of 32 over 400 epochs, cosine aligned learning rate schedule, and warm up of 20 epochs before mixup.

Due to the restrictions on the training and test sets, both splits present a highly imbalanced number of samples per condition, i.e., there is a significantly larger number of healthy breath and cough samples compared to the COVID-19 ones (approximately 73% against 27%, respectively.). To mitigate this issue, a focal loss approach was used in the loss function [55] for all our experiments, resulting in a more efficient training process. We followed the same calculation of weights through the same approach as in Section 6.4.2 as described in Table 10.

Task	Class	Validation split %	Calculated weight (1 - validation %)
Task 1	COVID-19	0.32	**0.68**
	Healthy	0.68	**0.32**
Task 2	COVID-19	0.63	**0.37**
	Healthy	0.37	**0.63**
Task 3	COVID-19	0.73	**0.27**
	Healthy	0.27	**0.73**

Table 10. Percentage distribution per class on the Cambridge Crowdsourced dataset.

An exhaustive search was executed to find optimal learning rate and dropout values hyperparameters in the classifier block; learning rates from 1×10^{-3} to 1×10^{-6} and dropout values from 0.1 to 0.9 were explored.

As described in section 6.5, we also explored the flexibility of the audio embeddings generator to dynamically adapt to the current target application by allowing to adjust its weights during training, just as represented in Figure 21.

Different learning rate values for the LLF and the HLF were utilized as a percentage of the fully connected layer learning rate; to fine-tune the model, this percentage was swept through different values from 10% to 100% in increments of 10%. The optimal values found for AemResNet across the three different tasks can be found in Table 11.

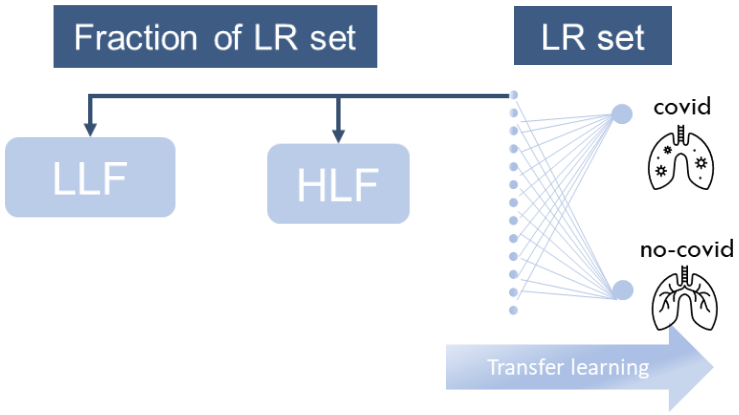


Figure 21. Usage of a learning rate fraction for LLF and HLF blocks on the COVID-19 classification.

Task	Learning rate	Learning Rate %	Dropout
Task 1	$1x10^{-3}$	80%	0.2
Task 2	$1x10^{-3}$	60%	0.2
Task 3	$1x10^{-6}$	90%	0.9

Table 11. Optimal hyper-parameters found for AemResNet per task.

Since there is no suggested official data split available for training/validation of the developed classification models, we randomly defined a set of 5 custom folds with an 80% split data for training, and 20% for validation (80/20 split). In all 5 folds, the proportion of available healthy and COVID-19 samples in maintained in both the training and validation split.

For a quantitative assessment of the performance of the proposed AemResNet model, Precision, Recall, and F1-score metrics were used for better understanding of our proposed implementation. These metrics are defined by:

$$Precision = \frac{TP}{TP + FP}$$

Equation 9. Precision computation based on confusion matrix.

$$Recall = \frac{TP}{TP + FN}$$

Equation 10. Recall computation based on confusion matrix.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Equation 11. F1-score computation used in this work.

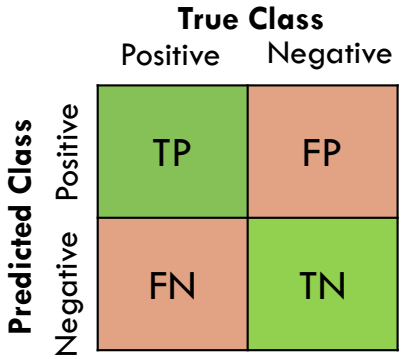


Figure 22. Confusion matrix on a binary classification application.

In Equations 9 and 10, the TP represents the true positives or the number of correctly classified breath and/or cough sounds into healthy or covid, FP represents an incorrect classification, and FN represents a miss classification. Finally, the F1-score is computed as in Equation 11 to have a single metric that represents the performance of our model.

7.5 Results and discussion

All experimental results obtained with AemResNet are described in Table 12. Our approach is averaged over 5-fold random 80/20 splits and is trained and validated for the three tasks. This table also shows how the performance obtained by the AemResNet compares to results reported in recent published works that benchmark over the same dataset. Although these works present their results based on different metrics, we tried to consolidate and compare the performance of our approach as much as possible.

We computed the F1-score from the SVM system in [18] based on the reported Precision and Recall and using Equation 11. From this, it can be observed that AemResNet presents a slightly better F1-score of around 3.0% for Task 1, but this difference is more significant for Task 2 (almost 12.0%), and for Task 3 (> 17.0%). This suggest that AemResNet can generalize better for COVID-19 detection if only one type of respiratory sounds is considered, i.e., cough or breath sounds in separate models.

In this context, the Recall metric represents how accurate are the models at correctly classifying healthy and COVID-19 sounds. We found that AemResNet yields better positive classification accuracy in Task 2 (>16.5%) and Task 3 (>5.6%). However, this was not the case for Task 1, where AemResNet results in <4.6% recall. Lastly, we compared our F1-score results to the ones reported in [62], where AemResNet felt short to the 1D CNN used in their work, particularly for Task 1 (~17.5%). A major difference here could be the use of efficient data augmentation procedures, which would suggest that handling of more data would be expected to be beneficial. We believe we could adopt this type of data augmentation to increase the robustness of our own e2e model and is part of our ongoing future research.

Overall, the results obtained by AemResNet suggest that the use of the pre-trained deep audio embeddings applied to the task of COVID-19 detection is a robust, convenient, and competitive approach by achieving a F1-score of 0.7332 for cough and breath sounds combined, 0.8773 for cough sounds, and 0.8654 for breath sounds, over the 2020 Cambridge Crowdsourced dataset. The highlighted AemResNet solution is the proposed approach in this work.

		Task 1			Task 2			Task 3		
Model	Folds	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
AemResNet	**5**	**0.6975**	**0.7235**	**0.7332**	**0.8697**	**0.8850**	**0.8773**	**0.9040**	**0.8300**	**0.8654**
Cambridge[18]	10	0.7200	0.6900	0.7047[b]	0.8000	0.7200	0.7579[b]	0.6900	0.6900	0.6900[b]
Ensemble[16]	3	-	0.7020	-	-	-	-	-	-	-
1D CNN[26]	-	-	-	0.9078	-	-	0.8926	-	-	0.8913
CI-ResNet[62]	10	-	0.7700	-	-	0.5350	-	-	0.7740	-

Table 12. Experimental validation results obtained as the 5-fold average of AemResNet compared to other published works. [b] F1-score was computed with Equation 11.

8. CONCLUSIONS

8.1 Conclusions

The analysis and development of AemNet was the main object of research through all this work. This e2e proposal differs from current SOTA audio classification works which depend on dedicated software or hardware to pre-process audio samples and transform them to spectral representations, which have shown the best performance when inputted to computational vision architectures, these exceed greatly the storage capacities of current neural network accelerators. Most of deep learning approaches do not limit their model complexity and require additional pre-processing steps for feature extraction at the front-end, compared to e2e approaches like AemNet and its respective enhancements like AemResNet, which facilitate portability into an inference engine.

Deep learning techniques such as transfer learning, data augmentation, pruning, quantization and learning rate percentage training were key to fulfill and adapt our proposal to the three main applications this work targeted which were:

1. A low-complexity model for acoustic scene classification on the DCASE 2021 Task 1A challenge,
2. A ResNet18 HLF usage of the AemNet aimed to classify respiratory sounds and indicate if cough sounds included crackles, wheezes, both or are classified as normal.
3. ResNet18 HLF usage of the AemNet renamed AemResNet aimed to distinguish breath and cough sounds from COVID-19 patients from non-COVID patients through three different tasks.

The general objective of this work was achieved, by evaluating the performance when changing the core elements of AemNet (LLF and HLF) and reconfirmed the robust generalization of audio classification by targeting different audio applications such as the DCASE2021 challenge and respiratory sounds classification.

For the specific objectives, we were able to conclude that LLF1 configuration is a better approach against LLF2 since the stride values and kernel values sweep deeper into the image-like representation of the audio. A ResNet18 improves AemNet performance against a VGG-ish approach in which a residual net is preferred when classifying audio. The AemNet adaptation to AemResNet is evaluated with three different datasets showing up to 12% accuracy increase against AemNet even with 2.6 million parameters less.

Also, we were able to understand how a focal loss approach works to mitigate the class imbalance presented mainly in the respiratory sound datasets (ICBHI Scientific Challenge and Cambridge Crowdsourced) and leveraged the enhancements this loss contributes against a cross-entropy loss.

Finally, experimental results show that AemResNet on respiratory sounds presents a high performance in two different biomedical applications: distinguish normal coughs against those with anomalies (wheezes, crackles, and both) as well as classifying COVID-19 patients through breath and cough samples. The research and experimentation of both these applications resulted in two papers: “Classification of Respiration Sounds Using Deep Pre-trained Audio Embeddings”[63] presented in IEEE Latin American Conference on Computational Intelligence (LA-CCI 2021) and “Detection of COVID-19 in Respiratory Sounds using End-to-End Deep Audio Embeddings” [64] in the 4th International Student Conference in Latin America Call for Papers contest presented through the IEEE EMBS Chapter in Guadalajara in was awarded the first place under the Graduate category.

8.2 Future work

As future work, the AemResNet performance can be analyzed further by evaluating the e2e model against a spectral representation form with a ResNet18 CNN. This comparison could provide a significant measurable variable to understand the tradeoff and impact in performance of an e2e model against a conventional audio classification approach.

Also, AemResNet can take advantage on larger datasets aimed to other biomedical applications, in which the e2e portability to inference engines can be a leverage for production.

There are still certain limits for AemResNet, the model complexity is still composed by a high number of parameters and as future work we might research other techniques to reduce the model complexity. On a short-term basis, a contribution would be to test the low-complexity techniques described in section 5 directly to AemResNet and measure its performance on lower memory footprints.

The low-complexity AemNet performance for the DCASE2021 challenge can certainly be improved, we believe we can explore other deep learning techniques like the ones described in other DCASE2021 submissions to fulfill the needs of the task 1 announced in the site.

Finally, I encourage the research community to continue the research of AemResNet and apply it to modern databases, so its performance can be evaluated and if needed, optimized.

REFERENCES

[1] L. Jiqing, D. Yuan, H. Jun, Z. Xianyu, and W. Haila, "Sports audio classification based on MFCC and GMM," in *2009 2nd IEEE International Conference on Broadband Network Multimedia Technology*, Oct. 2009, pp. 482–485. doi: 10.1109/ICBNMT.2009.5348520.

[2] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 7, pp. 504–516, Oct. 2002, doi: 10.1109/TSA.2002.804546.

[3] Y.-T. Peng, C.-Y. Lin, M.-T. Sun, and K.-C. Tsai, "Healthcare audio event classification using Hidden Markov Models and Hierarchical Hidden Markov Models," in *2009 IEEE International Conference on Multimedia and Expo*, Jun. 2009, pp. 1218–1221. doi: 10.1109/ICME.2009.5202720.

[4] "DCASE." http://dcase.community/ (accessed Jul. 31, 2021).

[5] J. J. Huang and J. J. A. Leanos, "AclNet: efficient end-to-end audio classification CNN," *ArXiv181106669 Cs Stat*, Nov. 2018, Accessed: Aug. 17, 2021. [Online]. Available: http://arxiv.org/abs/1811.06669

[6] P. Lopez-Meyer, "EFFICIENT END-TO-END AUDIO EMBEDDINGS GENERATION FOR AUDIO CLASSIFICATION ON TARGET APPLICATIONS," p. 5.

[7] G. Mendels, "How to apply machine learning and deep learning methods to audio analysis," *Medium*, Nov. 18, 2019. https://towardsdatascience.com/how-to-apply-machine-learning-and-deep-learning-methods-to-audio-analysis-615e286fcbbc (accessed Jun. 27, 2021).

[8] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, Brisbane Australia, Oct. 2015, pp. 1015–1018. doi: 10.1145/2733373.2806390.

[9] "UrbanSound8K," *Urban Sound Datasets*. https://urbansounddataset.weebly.com/urbansound8k.html (accessed Aug. 22, 2021).

[10] L. Zhang and D. Wang, "Large-Scale Whale-Call Classification by Transfer Learning on Multi-Scale Waveforms and Time-Frequency Features," p. 11, 2019.

[11] "Audio AI: Learning to understand sounds," *Bosch Global*. https://www.bosch.com/research/know-how/success-stories/audio-ai/ (accessed Jun. 27, 2021).

[12] A. Manzoor, Q. Pan, H. J. Khan, S. Siddeeq, H. M. A. Bhatti, and M. A. Wedagu, "Analysis and Detection of Lung Sounds Anomalies Based on NMA-RNN," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec. 2020, pp. 2498–2504. doi: 10.1109/BIBM49941.2020.9313197.

[13] L. Pham, I. McLoughlin, H. Phan, M. Tran, T. Nguyen, and R. Palaniappan, "Robust Deep Learning Framework For Predicting Respiratory Anomalies and Diseases," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, Jul. 2020, pp. 164–167. doi: 10.1109/EMBC44109.2020.9175704.

[14] L. D. Pham, H. Phan, R. Palaniappan, A. Mertins, and I. Mcloughlin, "CNN-MoE based framework for classification of respiratory anomalies and lung disease detection," *IEEE J. Biomed. Health Inform.*, pp. 1–1, 2021, doi: 10.1109/JBHI.2021.3064237.

[15] K. Minami, H. Lu, H. Kim, S. Mabu, Y. Hirano, and S. Kido, "Automatic Classification of Large-Scale Respiratory Sound Dataset Based on Convolutional Neural Network," in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, Oct. 2019, pp. 804–807. doi: 10.23919/ICCAS47443.2019.8971689.

[16] T. Nguyen and F. Pernkopf, "Lung Sound Classification Using Snapshot Ensemble of Convolutional Neural Networks," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, Jul. 2020, pp. 760–763. doi: 10.1109/EMBC44109.2020.9176076.

[17] N. Cummins, S. Scherer, J. Krajewski, S. Schnieder, J. Epps, and T. F. Quatieri, "A review of depression and suicide risk assessment using speech analysis," *Speech Commun.*, vol. 71, pp. 10–49, Jul. 2015, doi: 10.1016/j.specom.2015.03.004.

[18] C. Brown *et al.*, “Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data,” *San Diego*, p. 11.
[19] L. Pham, I. McLoughlin, H. Phan, R. Palaniappan, and A. Mertins, “Deep Feature Embedding and Hierarchical Classification for Audio Scene Classification,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2020, pp. 1–7. doi: 10.1109/IJCNN48605.2020.9206866.
[20] “Mixup data augmentation,” *Deep Learning Course Forums*, Sep. 24, 2018. https://forums.fast.ai/t/mixup-data-augmentation/22764 (accessed Dec. 05, 2020).
[21] F. Arabnezhad and B. Nasersharif, “Acoustic Scene Classification using Binaural Representation and Classifier Combination,” in *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, Oct. 2019, pp. 351–355. doi: 10.1109/ICCKE48569.2019.8964809.
[22] H. Bai, H. Chen, and Y. Yan, “Audio Scene Classification with Discriminatively-Trained Segment-Level Features,” in *2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, Jul. 2019, pp. 354–359. doi: 10.1109/ICMEW.2019.00067.
[23] M. Massoudi, S. Verma, and R. Jain, “Urban Sound Classification using CNN,” in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Jan. 2021, pp. 583–589. doi: 10.1109/ICICT50816.2021.9358621.
[24] R. Avanzato, F. Beritelli, F. Di Franco, and V. F. Puglisi, “A Convolutional Neural Networks Approach to Audio Classification for Rainfall Estimation,” in *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Sep. 2019, vol. 1, pp. 285–289. doi: 10.1109/IDAACS.2019.8924399.
[25] J. Naranjo-Alcazar, S. Perez-Castanos, I. Martín-Morató, P. Zuccarello, F. J. Ferri, and M. Cobos, “A Comparative Analysis of Residual Block Alternatives for End-to-End Audio Classification,” *IEEE Access*, vol. 8, pp. 188875–188882, 2020, doi: 10.1109/ACCESS.2020.3031685.
[26] S. Abdoli, P. Cardinal, and A. L. Koerich, “End-to-End Environmental Sound Classification using a 1D Convolutional Neural Network,” *ArXiv190408990 Cs Stat*, Apr. 2019, Accessed: Aug. 17, 2021. [Online]. Available: http://arxiv.org/abs/1904.08990
[27] K. Doshi, “Audio Deep Learning Made Simple (Part 1): State-of-the-Art Techniques,” *Medium*, May 21, 2021. https://towardsdatascience.com/audio-deep-learning-made-simple-part-1-state-of-the-art-techniques-da1d3dff2504 (accessed Aug. 11, 2021).
[28] “Waves and Wavelengths | Introduction to Psychology.” https://courses.lumenlearning.com/atd-bhcc-intropsych/chapter/waves-and-wavelengths/ (accessed Aug. 11, 2021).
[29] H. Fayek, “Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What’s In-Between,” *Haytham Fayek*, Apr. 21, 2016. https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html (accessed Nov. 10, 2021).
[30] “Mel.” https://www.sfu.ca/sonic-studio-webdav/handbook/Mel.html (accessed Nov. 11, 2021).
[31] Xin Huang, *Andrew Ng: Deep Learning, Self-Taught Learning and Unsupervised Feature Learning*, (May 14, 2013). Accessed: Nov. 11, 2021. [Online]. Available: https://www.youtube.com/watch?v=n1ViNeWhC24
[32] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
[33] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ArXiv14091556 Cs*, Apr. 2015, Accessed: Nov. 11, 2021. [Online]. Available: http://arxiv.org/abs/1409.1556
[34] P. Ruiz, “Understanding and visualizing ResNets,” *Medium*, Apr. 23, 2019. https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8 (accessed Nov. 12, 2021).
[35] W. Koehrsen, “Neural Network Embeddings Explained,” *Medium*, Oct. 02, 2018. https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526 (accessed Aug. 17, 2021).

[36] A. Paszke, S. Gross, and F. Massa, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024--8035.
[37] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *ArXiv170404861 Cs*, Apr. 2017, Accessed: Aug. 17, 2021. [Online]. Available: http://arxiv.org/abs/1704.04861
[38] “What is a neural network accelerator?,” *FierceElectronics*. https://www.fierceelectronics.com/electronics/what-a-neural-network-accelerator (accessed Aug. 17, 2021).
[39] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, “Detection and Classification of Acoustic Scenes and Events,” *IEEE Trans. Multimed.*, vol. 17, no. 10, pp. 1733–1746, Oct. 2015, doi: 10.1109/TMM.2015.2428998.
[40] “Freesound - Freesound.” https://freesound.org/ (accessed Aug. 21, 2021).
[41] K. J. Piczak, *karolpiczak/ESC-50*. 2022. Accessed: Jan. 27, 2022. [Online]. Available: https://github.com/karolpiczak/ESC-50
[42] J. Salamon, C. Jacoby, and J. P. Bello, “A Dataset and Taxonomy for Urban Sound Research,” in *Proceedings of the 22nd ACM international conference on Multimedia*, Orlando Florida USA, Nov. 2014, pp. 1041–1044. doi: 10.1145/2647868.2655045.
[43] M. Kelechava, “Urban Sound Classification with Librosa — tricky cross-validation,” *Medium*, Oct. 15, 2020. https://towardsdatascience.com/urban-sound-classification-with-librosa-nuanced-cross-validation-5b5eb3d9ee30 (accessed Jan. 27, 2022).
[44] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” *ArXiv14111792 Cs*, Nov. 2014, Accessed: Sep. 18, 2021. [Online]. Available: http://arxiv.org/abs/1411.1792
[45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *ArXiv151203385 Cs*, Dec. 2015, Accessed: Apr. 18, 2021. [Online]. Available: http://arxiv.org/abs/1512.03385
[46] S. Hershey *et al.*, “CNN Architectures for Large-Scale Audio Classification,” *ArXiv160909430 Cs Stat*, Jan. 2017, Accessed: Nov. 12, 2021. [Online]. Available: http://arxiv.org/abs/1609.09430
[47] J. Huang, H. Lu, P. Lopez Meyer, H. Cordourier, and J. Del Hoyo Ontiveros, “Acoustic Scene Classification Using Deep Learning-based Ensemble Averaging,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019, pp. 94–98. doi: 10.33682/8rd2-g787.
[48] T. Heittola, A. Mesaros, and T. Virtanen, “Acoustic scene classification in DCASE 2020 Challenge: generalization across devices and low complexity solutions,” *ArXiv200514623 Eess*, Nov. 2020, Accessed: Nov. 12, 2021. [Online]. Available: http://arxiv.org/abs/2005.14623
[49] J. F. Gemmeke *et al.*, “Audio Set: An ontology and human-labeled dataset for audio events,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 776–780. doi: 10.1109/ICASSP.2017.7952261.
[50] J. Frankle and M. Carbin, “The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks,” *ArXiv180303635 Cs*, Mar. 2019, Accessed: Nov. 12, 2021. [Online]. Available: http://arxiv.org/abs/1803.03635
[51] A. Kozlov, I. Lazarevich, V. Shamporov, N. Lyalyushkin, and Y. Gorbachev, “Neural Network Compression Framework for fast model inference,” *ArXiv200208679 Cs Eess*, Dec. 2020, Accessed: Nov. 12, 2021. [Online]. Available: http://arxiv.org/abs/2002.08679
[52] A. Kozlov, *Neural network compression framework for pytorch (nncf)*. 2020. [Online]. Available: https://https://github. com/openvinotoolkit/nncf pytorch
[53] B. M. Rocha *et al.*, “An open access database for the evaluation of respiratory sound classification algorithms,” *Physiol. Meas.*, vol. 40, no. 3, p. 035001, Mar. 2019, doi: 10.1088/1361-6579/ab03ea.
[54] “ICBHI 2017 Challenge | ICBHI Challenge.” https://bhichallenge.med.auth.gr/ (accessed Jul. 05, 2021).

[55] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” *ArXiv170802002 Cs*, Feb. 2018, Accessed: Apr. 18, 2021. [Online]. Available: http://arxiv.org/abs/1708.02002
[56] J. Howard and S. Ruder, “Universal Language Model Fine-tuning for Text Classification,” *ArXiv180106146 Cs Stat*, May 2018, Accessed: Nov. 12, 2021. [Online]. Available: http://arxiv.org/abs/1801.06146
[57] S. Ivanov, “Differential Learning Rates,” *Medium*, Nov. 20, 2017. https://blog.slavv.com/differential-learning-rates-59eff5209a4f (accessed Nov. 12, 2021).
[58] “Coronavirus.” https://www.who.int/westernpacific/health-topics/coronavirus (accessed Sep. 18, 2021).
[59] H. Coppock, A. Gaskell, P. Tzirakis, A. Baird, L. Jones, and B. Schuller, “End-to-end convolutional neural network enables COVID-19 detection from breath and cough audio: a pilot study,” *BMJ Innov.*, vol. 7, no. 2, pp. 356–362, Apr. 2021, doi: 10.1136/bmjinnov-2021-000668.
[60] M. A. Nessiem, M. M. Mohamed, H. Coppock, A. Gaskell, and B. W. Schuller, “Detecting COVID-19 from Breathing and Coughing Sounds using Deep Neural Networks,” in *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*, Jun. 2021, pp. 183–188. doi: 10.1109/CBMS52027.2021.00069.
[61] “New app collects the sounds of COVID-19,” *University of Cambridge*, Apr. 06, 2020. https://www.cam.ac.uk/research/news/new-app-collects-the-sounds-of-covid-19 (accessed Sep. 18, 2021).
[62] K. K. Lella, A. Pja, and Department of Computer Applications, NIT Tiruchirappalli, Tamil Nadu, India, “Automatic COVID-19 disease diagnosis using 1D convolutional neural network and augmentation with human respiratory sound based on parameters: cough, breath, and voice,” *AIMS Public Health*, vol. 8, no. 2, pp. 240–264, 2021, doi: 10.3934/publichealth.2021019.
[63] C. A. Galindo-Meza, J. A. del Hoyo Ontiveros, and P. Lopez-Meyer, “Classification of Respiration Sounds Using Deep Pre-trained Audio Embeddings,” in *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Temuco, Chile, Nov. 2021, pp. 1–5. doi: 10.1109/LA-CCI48322.2021.9769831.
[64] C. A. Galindo-Meza, J. A. del Hoyo Ontiveros, J. I. Torres Ortega, and P. Lopez-Meyer, “Detection of COVID-19 in Respiratory Sounds using End-to-End Deep Audio Embeddings,” *2021 Int. Stud. Compet.*, vol. page 101, p. 4. https://somib.org.mx/wp-content/uploads/2022/02/PROCEEDINGS-ISC2021-1.pdf (accessed Jun. 1, 2022).

APPENDIX A

Classification of Respiration Sounds Using Deep Pre-trained Audio Embeddings

Carlos A. Galindo Meza
Dept. of Electronics, Systems and Informatics, ITESO
Tlaquepaque, Jalisco, Mexico
ms729621@iteso.mx

Juan A. del Hoyo Ontiveros
Intel Labs
Intel Corporation
Guadalajara, Jalisco, Mexico
juan.antonio.del.hoyo.ontiveros@intel.com

Paulo Lopez-Meyer
Intel Labs
Intel Corporation
Guadalajara, Jalisco, Mexico
paulo.lopez.meyer@intel.com

***Abstract*—In this work we present the use of an end-to-end deep learning based pre-trained audio embeddings generator, and apply it to the purpose of classification of respiration sounds. With this approach, there is no need to pre-compute spectral representations, e.g. MFCC or filterbanks, since the classification model uses raw audio as the input. Transfer learning was used to train an audio classifier for sounds of respiratory cycles as defined in the ICBHI 2017 challenge. The results on this dataset show that this end-to-end model represents a viable alternative to more common spectral-based classifiers, while achieving state-of-the-art performance.**

***Index Terms*—audio classification, deep audio embeddings, deep learning, transfer learning, respiration sound classification.**

I. Introduction

The direct analysis of respiration sounds by health professionals provide significant insight in order to build a clinical assessment of different health conditions, e.g. pneumonia, bronchitis, asthma, etc. The typical method carried out for this purpose is for a patient to attend the doctor's office, where an auscultation takes place over the chest and the back of the patient looking for characteristic sounds in the respiration through a stethoscope. This method is only effective when the patient attends physically to the evaluation, with the major constraint that is not subject to objective monitoring over extended periods of time.

Different automatic and novel methods have been proposed for sound of respiration cycle classification, i.e. to identify normal from abnormal respiration that can be associated to different medical conditions. The recent methods reported in the literature typically follow an implementation based on deep learning technologies. In [1] a recurrent neural network is used for lung sound identification. An ensemble of two large deep learning models is used in [2] to enhance performance in the prediction of respiratory anomalies. A deep learning architecture was also used in [3] to detect a possible lung disease with the classification of respiratory anomalies. A VGG16 convolutional neural network was proposed in [4] for automatic classification of respiratory sounds. Another example in the use of deep learning is the ensemble of convolutional neural networks proposed in [5] for lung sound classification.

This work proposes the use of an end-to-end (e2e) deep learning based model to identify normal respiration cycles from those showing presence of wheezes, crackles, or both. Section II describes the technology used for deep learning audio classification; Section III presents details on the methodology and the experimentation followed; Section IV presents the experimental results obtained and a discussion around them; and finally, Section V presents the conclusions drawn from this work.

II. End-to-End Audio Embeddings generation

This work presents the use of a deep learning pre-trained audio embeddings generator constructed with e2e convolutional neural network (CNN): this means that the model takes a raw time-domain waveform as input instead of typically used spectral representations. This approach results attractive when considering deployment of the inference model in dedicated hardware, since its input is based on raw audio without any additional elaborated pre-processing. The use of this type of deep e2e audio embeddings technology has been described in detail in [6], where pre-trained models efficiently generated high quality and robust audio embeddings that were successfully implemented in target applications through transfer learning [7] over different benchmark datasets. The proposed e2e CNN architecture discussed in this work comprises three main blocks as seen in Fig. 1: the low-level feature block (LLF), and the high-level feature block (HLF) that constitute the audio embeddings generator, and a final classification block trained with the embeddings generated by the previous two blocks.

The LLF block is engineered to extract meaningful and discriminating features, and replaces the typically used spectral feature extraction in audio classification. Details of this block are shown in Table I, where there are two 1-dimensional strided convolutional layers (Conv), each followed by a batch normalization (BN) layer and a ReLU activation function. The time-domain waveform is input to the LLF block produces an output of 128 channels at frame rate of 10ms after an added max-pool layer, from a 16 kHz raw audio input. We observed heuristically that 128 channels provide a robust performance on audio classification, while a 10 ms window is used to have a good time resolution of the signal. On the sampling frequency,

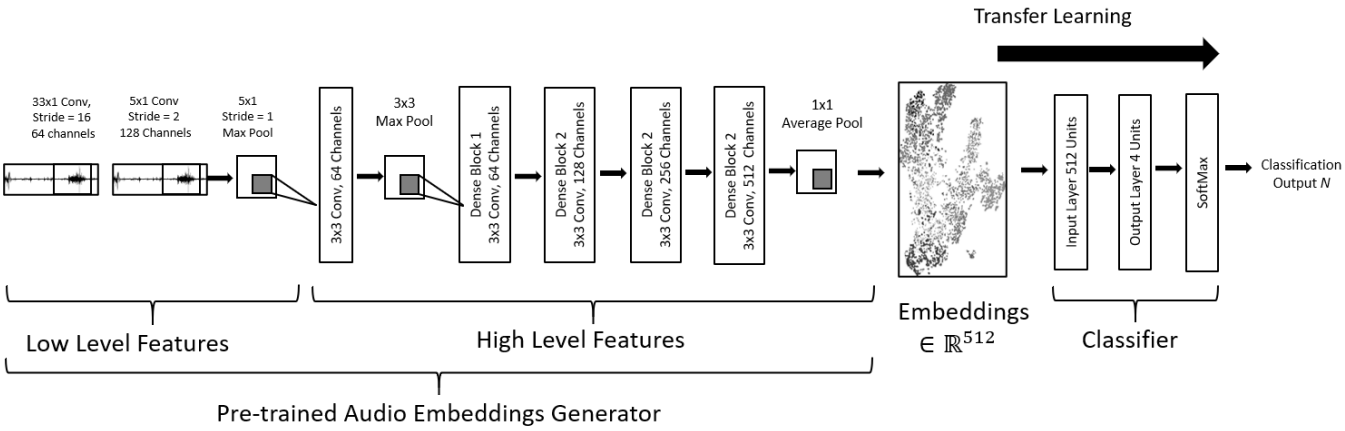


Fig. 1. E2E CNN architecture. The LLF and the HLF blocks are pre-trained with a large dataset to generate audio embeddings, and the classifier layer is trained by means of transfer learning.

16 kHz is the chosen trade-off value between a good quality audio sample and a low complexity model. For 1 second of audio input, the LLF produces an output tensor of dimension [128, 1, 100]; these convolutional layers act as a trainable equivalent of a spectral filterbank feature extraction.

The output of the LLF block results in an image-like tensor that represents the input to the HLF block. The HLF block follows a CNN topology similar to the ones commonly used for image classification. For this work, we experimented with a ResNet topology of 18 layers [8], detailed in Table II.

The final block of the e2e CNN acts as a classifier, and comprises a dropout (DO) layer to reduce overfitting and one fully connected layer with linear activation functions. The input to this classification block is the output of the convolutional layer averaged pooled to produce a vector of fixed size representing 512 audio embeddings that represent the input to the classifier. The use of this average pool layer allows arbitrary lengths of audio inputs without the need to modify the parameters of the network model. A softmax layer is used at the output to present normalized output values.

TABLE II
HIGH LEVEL FEATURES AND CLASSIFICATION BLOCKS ARCHITECTURE

Layer	Stride	Out Channels	Kernel Size
Conv3+BN+ReLU	1,1	64	3x3
Maxpool2	2,2	64	3x3
Conv4+BN+ReLU	1,1	64	3x3
Conv5+BN	1,1	64	3x3
Conv6+BN+ReLU	1,1	64	3x3
Conv7+BN	1,1	64	3x3
Conv8+BN+ReLU	2,2	128	3x3
Conv9+BN	1,1	128	3x3
Conv10+BN+ReLu	1,1	128	3x3
Conv11+BN	1,1	128	3x3
Conv12+BN+ReLU	2,2	256	3x3
Conv13+BN	1,1	256	3x3
Conv14+BN+ReLu	1,1	256	3x3
Conv15+BN	1,1	256	3x3
Conv16+BN+ReLU	2,2	512	3x3
Conv17+BN	1,1	512	3x3
Conv18+BN+ReLu	1,1	512	3x3
Conv19+BN	1,1	512	3x3
Avpool1	1	512	1x1
DO+Linear+Softmax	-	*N**	-

* Number of outputs defined by the number of audio classes.

III. METHODOLOGY AND EXPERIMENTS

The LLF and HLF blocks described in the previous section were pre-trained on a large set of audio data. The resulting pre-trained model is then subsequently fine-tuned over a given audio classification application e.g. classification of respiratory cycles sounds, by means of transfer learning.

A. Pre-Training of the LLF and the HLF

The LLF and the HLF blocks that constitute the audio embedding generator are pre-trained using AudioSet, a large dataset of manually annotated audio events released by Google [9]. This dataset contains 2.1 million samples corresponding to 5.8 thousand hours of recordings, representing 527 different audio classes. Before using this embedding generator model for a specific classification application, the final classification block is removed, i.e. the fully connected layer, resulting in a 512-dimensional audio embeddings representation as the output.

TABLE I
LOW LEVEL FEATURES ARCHITECTURE

Layer	Stride	Out Channels	Kernel Size
Conv1+BN+ReLU	16	64	33x1
Conv2+BN+ReLU	2	128	5x1
Maxpool1	1	128	5x1

For the pre-training of audio embedding generator on Audioset, the dataset's single channel raw audio downsampled to 16 kHz was used, with additional standardization in amplitude (subtracting the mean and dividing by the standard deviation of the audio signal). Data augmentation techniques typically used in audio processing were also used: random noise addition, random cropping of a segment of the sample audio signal, and random gain variation. Additionally, the widely used mixup data augmentation technique [10] was used. During the training process, random audio clips were selected and used in mini-batches. During validation, complete standardized audio

segments were used for inference.

Adam optimizer with a learning rate of 5e-4 was used, with a weight decay of 1e-8, and a mini-batch size of 512 over 80 epochs. Cosine aligned learning rate schedule was used. This audio embedding was trained using the available unbalanced set and validated with the evaluation set for the 527 classes. All experimentation was executed using the Pytorch framework [11]. These audio embedding generator model resulted in 11,744,143 number of trainable parameters, with a mean average precision (mAP) of 0.3690.

B. End-to-End CNN for classification of respiration sounds

The pre-trained audio embedding generator was used to train a respiration sound classifier using the commonly used transfer learning technique. For this purpose, the ICBHI 2017 challenge dataset [12] was used. This dataset is an ensemble of 920 recordings from 126 subjects, resulting in 6,898 respiration cycles over 5.5 hours. These data were annotated by experts to include normal, crackles, wheezes, and a combination of wheezes and crackles respiration sounds. For experimentation in this work, audio data were processed to be single channel with 16kHz sampling rate, a 16 bit resolution, and standardized in amplitude. Initially, the challenge defined 4 different tasks; this work focuses only in the development of a classifier for the classification of 4 different respiration sounds as noted above (normal, crackles, wheezes, both).

The training strategy followed for this application is similar to the pre-training of the audio embedding generator: Adam optimizer was used with a learning rate of 1e-3, weight decay of 2e-4, mini-batch size of 64 over 350 epochs, cosine aligned learning rate schedule, and warm up of 35 epochs before mixup. It is important to notice that the ICBHI 2017 dataset presents a highly imbalanced number of samples per type of respiration cycles, i.e. there is a significantly larger number of normal respiration samples compared to the other 3 not normal types of respiration. Due to this issue, a focal loss approach was used in the loss function [13], resulting in a more efficient training process.

We explored the flexibility of the audio embeddings generator to dynamically adapt to the respiration classification application by allowing to adjust its weights during training. Different learning rate values for the LLF and the HLF were utilized as a percentage of the fully connected layer learning rate; experiments to find the optimal percentage were evaluated sweeping through different values from 10% to 100% in increments of 10%.

The training and validation official split defined by the ICBHI 2017 challenge was strictly followed for cross-validation analysis of results and viability, and direct comparison with other published works, where 60% of the data is used for training, and 40% for validation (60/40 split). Additionally, as observed in other published works, a set of 5 custom folds were defined randomly with a split 80% of the data for training, and 20% for validation (80/20 split).

The model obtained after training the respiration sound classifier using the pre-trained audio embedding generator resulted in 11,475,844, which is 2.3% less parameters due to a smaller classifier block with only 4 outputs. Additionally, the number of multiply-accumulate operations (MACs) results in 1.84×10^9.

TABLE III
EXPERIMENTAL RESULTS OBTAINED OVER THE OFFICIAL AND CUSTOM SPLIT OF THE ICBHI 2017 CHALLENGE DATASET

	Official Split			**Custom Split**		
LR% *	**Sens**	**Spec**	**Score**	**Sens**	**Spec**	**Score**
10%	0.289	0.799	0.544	**0.678**	**0.865**	**0.772**
20%	0.311	0.767	0.539	0.644	0.874	0.759
30%	0.298	0.786	0.542	0.645	0.868	0.757
40%	0.289	0.804	0.547	0.641	0.845	0.743
50%	0.396	0.706	0.551	0.647	0.850	0.749
60%	**0.251**	**0.870**	**0.561**	0.622	0.859	0.740
70%	0.382	0.710	0.546	0.606	0.863	0.735
80%	0.342	0.768	0.555	0.603	0.865	0.734
90%	0.284	0.790	0.537	0.596	0.865	0.731
100%	0.387	0.729	0.558	0.594	0.855	0.725
No TL**	0.304	0.765	0.535	0.540	0.828	0.684

* Learning rate % used by the embedding model
**No transfer learning used, model trained from scratch

For the assessment of the performance of the proposed e2e CNN model, Sensitivity, Specificity, and Score metrics were followed as explicitly stated for the ICBHI 2017 challenge. These metrics are defined by:

$$Sensitivity = \frac{TPc + TPw + TPb}{Nc + Nw + Nb}, \tag{1}$$

$$Specificity = \frac{TPn}{Nn}, \tag{2}$$

In Equations (1) and (2), the *TP* represents the true positives or the number of correctly classified respiration cycles, and *N* represents the total number existing samples; these counts are defined for each one of the respiration sound classes: *n* for normal, *c* for crackles, *w* for wheezes, and *b* for both crackles and wheezes. Finally, the computation of the Score is obtained as the average between the Sensitivity and Specificity. The experimental results obtained based on the metrics defined above are presented in the following section.

IV. RESULTS AND DISCUSSION

All experimental results over the official and custom validation splits are displayed in this section. Table III shows the impact over the final score validation metric when updating the weights of the audio embeddings generator as a percentage of

TABLE IV
COMPARISON OF THE PROPOSED E2E CNN WITH OTHER SOTA METHODS

	Official Split			**Custom Split**		
Model	**Sens**	**Spec**	**Score**	**Sens**	**Spec**	**Score**
BLSTMDAE [1]	*	*	*	0.720	0.920	0.760
SP+SE VGG16 [4]	0.280	0.810	0.540	*	*	*
C-DNN [3]	0.260	0.680	0.470	0.680	0.900	0.790
SE-8Cycle [5]	*	*	*	0.694	0.873	0.784
Ensemble [2]	*	*	*	**0.730**	**0.860**	**0.800**
e2e CNN *	**0.251**	**0.870**	**0.561**	0.678	0.865	0.772

* Approach proposed in this work.

the learning rate used by the classification block. This shows the importance of considering this updating flexibility during training applied to the pre-trained audio embeddings during the transfer learning process, in order to have a more robust model.

We have observed experimentally this behavior in other applications and hypothesized that there has to be an optimal learning rate percentage for the embeddings model, i.e. a smaller learning rate than the one used in the last fully connected layer; this is consistent with the differential learning rates observed in other works [14], [15], where it is explained that for pre-trained models there is not a need to change significantly the front-end layers at the same rate as the last layers, since these have already learned to do a good generalization over rough initial features. The results in Table III support this idea.

In this work we used transfer learning as an attempt to increase the respiration sound classification performance. The last row in Table III also shows the performance of an e2e CNN models trained without the use of any pre-trained embeddings; over the official split the score results are 0.535 Vs. 0.561, and over the custom split the score results are 0.684 Vs. 0.772 for scratch training and transfer learning training, respectively. These results show the convenience and benefit of the use of the pre-trained deep audio embeddings implemented to the task of respiration sound classification application.

Table IV shows a comparison of our e2e CNN with state-of-the-art (SOTA) results reported in recent published works [1]–[5]. Not all these works present their results on both the official and a custom split, but are considered here to present a broader scope on how the proposed e2e CNN model compares to these. In these results it can be observed that generally, the official split tends to have smaller score values than the custom splits; this can be explained by the fact that the official split presents a more challenging problem with its 60/40 data distribution for training and validation. In this case, the e2e CNN used in this work achieved the highest score of 0.561 when compared to [3], [4], which is to the best of our knowledge above the SOTA reported in the existing literature. Comparison over the custom split is not as straight forward as with the official split, since the use of random splits and number of folds for cross-validation will differ for each work, but gives a clear idea that the e2e CNN is performing competitively.

A factor that might impact the difference in score results on the custom splits could be explained by the difference in the deep learning model complexity, e.g. 11.4 million parameters of the e2e CNN Vs. 39 million parameters in [5]. Ongoing research for this type of acoustic applications consist on the optimization of the HLF block; the ResNet network used here contributes significantly to the 11.4 million parameters of the final model; we are addressing the issue of reducing the number of the parameters in the model by exploring different topologies that could reduce this number, while maintaining the same or better performance.

V. Conclusion

The experimental results presented in this work prove that an e2e deep learning approach can be successfully used to classify different sounds of respiration cycles such as normal, whezzes, crackles or both. The main characteristic of this approach is that it avoids the need of additional pre-processing steps for feature extraction, thus facilitating its portability into an e2e inference engine. Through the use of pre-trained deep audio embeddings, a respiration sounds classifier model was build through transfer learning that achieved a SOTA Score of 0.561 over the official split defined for the ICBHI 2017 challenge dataset. Additionally, this model achieved a competitive 0.772 score over custom defined 5-fold random splits.

Acknowledgment

We appreciate the help provided by Israel Torres-Ortega in the handling and structuring of the data for easy of use.

References

[1] A. Manzoor, Q. Pan, H. J. Khan, S. Siddeeq, H. M. A. Bhatti, and M. A. Wedagu, "Analysis and Detection of Lung Sounds Anomalies Based on NMA-RNN," in 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Dec. 2020, pp. 2498–2504.

[2] L. Pham, I. McLoughlin, H. Phan, M. Tran, T. Nguyen, and R. Palaniappan, "Robust Deep Learning Framework For Predicting Respiratory Anomalies and Diseases," in 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), Jul. 2020, pp. 164–167.

[3] L. D. Pham, H. Phan, R. Palaniappan, A. Mertins, and I. Mcloughlin, "CNN-MoE based framework for classification of respiratory anomalies and lung disease detection," IEEE J. Biomed. Health Inform., pp. 1–1, 2021.

[4] K. Minami, H. Lu, H. Kim, S. Mabu, Y. Hirano, and S. Kido, "Automatic Classification of Large-Scale Respiratory Sound Dataset Based on Convolutional Neural Network," in 2019 19th International Conference on Control, Automation and Systems (ICCAS), Oct. 2019, pp. 804–807.

[5] T. Nguyen and F. Pernkopf, "Lung Sound Classification Using Snapshot Ensemble of Convolutional Neural Networks," in 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), Jul. 2020, pp. 760–763.

[6] P. Lopez-Meyer, J. A. del Hoyo Ontiveros, H. Lu, G. Stemmer, "Efficient Ent-to-End Audio Embedding Generation for Audio Classification on Target Applications", IEEE International Conference on Acoustics, Speech, and Signal Processing 2021. Accepted for presentation June 2021.

[7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 3320–3328. Curran Associates, Inc., 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," ArXiv151203385 Cs, Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.03385.

[9] Jort F. Gemmeke Daniel P. W. Ellis Dylan Freedman Aren Jansen Wade Lawrence R. Channing Moore Manoj Plakal Marvin Ritter, "Audio Set: An ontology and human-labeled dataset for audio events", Proc. IEEE ICASSP 2017, New Orleans, LA.

[10] H. Zhang, M. Isse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," CoRR, vol. abs/1710.09412, 2017.

[11] A. Paszke et. al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[12] J-B. M. Rocha et al., " Respiratory Sound Database for the Development of Automated Classification", Precision Medicine Powered by pHealth and Connected Health. ICBHI 2017. IFMBE Proceedings, vol 66. Springer, Singapore.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," ArXiv170802002 Cs, Feb. 2018.
[14] J. Howard, H. Ruder, "Universal Language Model Fine-tuning for Text Classification", Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia, 2018.
[15] S. Ivanov, "Differential Learning Rates". 2017. [Online]. Available: https://blog.slavv.com/differential-learning-rates-59eff5209a4f.

APPENDIX B

Detection of COVID-19 in Respiratory Sounds using End-to-End Deep Audio Embeddings

1st Carlos A. Galindo-Meza
DESI
Instituto Tecnologico de Estudios Superiores de Occidente
Tlaquepaque, Jal., Mexico
ms729621@iteso.mx

2nd Juan A. del Hoyo Ontiveros
Electrónica e Informática
Universidad Autónoma de Guadalajara
Zapopan, Jal., Mexico
juan.hoyo@edu.uag.mx

3rd Jose I. Torres Ortega
Programa de IOT
Universidad Alinnco
Santiago de Querétaro, Qro., Mexico
israel.torres@alinnco.edu.mx

4th Paulo Lopez-Meyer
Intel Labs
Intel Corporation
Zapopan, Jal., Mexico
paulo.lopez.meyer@intel.com

Abstract*— Due to the COVID-19 worldwide pandemic situation, automatic audio classification research has been of interest for analysis of respiratory sounds. Several deep learning approaches have shown promising performance for distinguishing COVID-19 in respiratory cycles. In this work we explored the usage of transfer learning from a pre-trained end-to-end deep-learning based audio embeddings generator named AemResNet, applied to the classification of respiration and coughing sounds into healthy or COVID-19. We experimented with the publicly available large-scale Cambridge Crowdsourced dataset of respiratory sounds collected to aid diagnosis of COVID-19. Our presented work focuses into 3 experimental tasks: 1) detection of COVID-19 from a combination of breath and cough sounds, 2) detection of COVID-19 from breath sounds only, and 3) detection of COVID-19 from cough sounds only. The experimental results obtained over this respiratory dataset show that a pre-trained audio embedding generator achieves competitive performance compared to the recent published state-of-the-art.

Keywords—audio classification, cough sounds, COVID-19 detection, deep learning, respiratory sounds, transfer learning.

I. INTRODUCTION

Coronavirus (COVID-19) is an infectious disease caused by the severe acute respiratory syndrome coronavirus (SARS-CoV-2) virus [1] first detected in Wuhan, China in 2019. On March 2020th, COVID-19 was declared a pandemic by the World Health Organization (WHO). Most people experience moderate respiratory symptoms such as: coughing, fever, and shortness of breath. The first time this novel virus was detected was within a cluster of patients with pneumonia of unknown cause. According to the WHO, 15% of overall COVID-19 patients present a severe pneumonia [1], which is auscultated by a physician listening respiratory sounds through breath and cough. The main purpose of recording respiratory sounds is to find a weakness of hypoventilation which can lead to diagnose the patient illness.

Nowadays, there are several methods proposed to distinguish the respiratory cycles, e.g., identifying a shortness of breath mostly related to pneumonia. The implementation of the most recent approaches on respiratory sound classification includes a recurrent neural network used for lung sound classification in [2], two deep learning ensemble model aimed to predict respiratory anomalies is proposed in [3], a deep learning architecture to detect possible lung disease in presented in [4] by classifying respiratory anomalies. A VGG16 CNN for automatic classification of respiratory sounds was proposed in [5] also by means of deep learning.

As well, COVID-19 aimed works have taken part on the research community. The work reported in [6] shows the efforts on the creation of an Android application aimed to collect different sounds from patients such as breath, cough, and speech; with this, they have created a dataset containing more than 459 samples from 378 patients through a crowdsourced methodology, named Cambridge Crowdsourced dataset. In this work, some machine learning (ML) techniques such as Support Vector Machines (SVM) were used as the classifier for COVID-19 detection. In [7], the composition of residual network blocks is used to classify COVID-19 based on audio spectrograms and motivates to a comprehensive follow-up research. On [8], respiratory audio recordings are treated as a visual representation through two different spectrogram configurations and as raw audio, each of these samples are inputted into a CNN layer and the output is concatenated and ensembled to classify COVID-19. Overall, it can be observed how deep learning is currently leading the state-of-the-art (SOTA) when it comes to audio classification for COVID-19.

In this work we propose the use of an end-to-end (e2e) deep learning-based model to identify healthy breath and/or coughing sounds from COVID-19 ones. We have arranged our work as follows: Section II describes the methodology followed for the implementation of the deep learning audio classification of healthy vs COVID-19 sounds; Section III presents a clear explanation on the experimental setup; Section IV presents the experimental results obtained and the discussion around them; and finally, Section V presents the conclusions drawn from this work.

II. METHODOLOGY

A deep learning approach for detection of COVID-19 respiratory sounds presented in this work, based on an end-to-end (e2e) convolutional neural network (CNN); this means that no additional audio spectral representation is needed since the time-domain signals are the input to the neural network architecture. This approach seems optimal when considering the dedicated hardware limitations for inference deployment. The core of this work is an ongoing effort of the e2e audio embeddings generator described in previous published works [9]–[12], where pre-trained models are created through an available large audio dataset, that efficiently generate robust

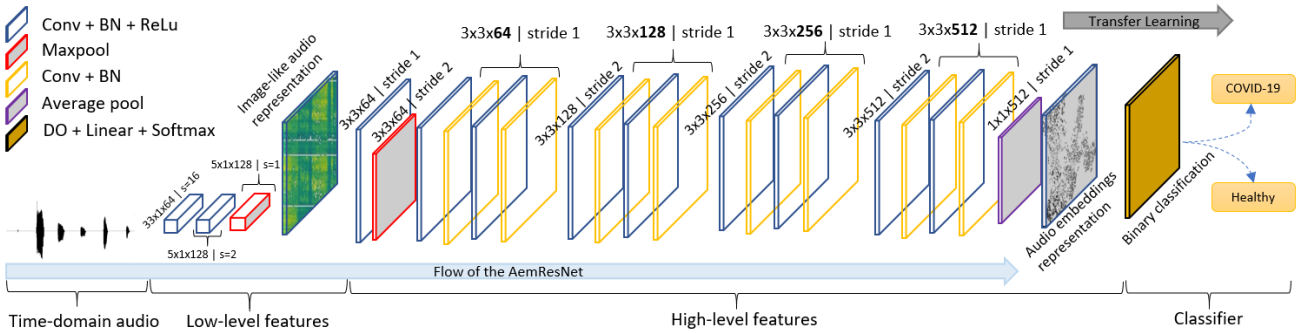


Fig. 1. AemResNet archictecture. The LLF and HLF blocks are pretrained with a large dataset to generate audio embeddings, and the classifier layer is trained by means of transfer learning.

audio embeddings aimed for different audio scene and events classification. The proposed e2e CNN architecture is named AemResNet, and it comprises three main blocks as seen in Fig 1: the low-level feature block (LLF) that acts as a front-end learnable feature extraction module, the high-level feature block (HLF) that is trained to become a deep learning-based audio embeddings generator, and a final classification block that is trained with the audio embeddings output by the HLF.

The purpose of the LLF block is to discriminate and extract features based purely from raw audio; this block replaces the visual representation of audio through spectrograms commonly used in most audio classification tasks. In Fig 1, the details of this block are described, where we find two 1-dimensional (1D) strided convolutional layers (Conv), each followed by a batch normalization layer (BN) and a ReLu activation function. The 16 kHz time-domain audio waveform inputted to the LLF block is converted to 128 channels using a time window resolution of 10ms after an added max-pool layer. For each second of audio input, the LLF block creates a [128, 1, 100] dimension tensor which act as a trainable correspondent to a spectral filterbank feature extraction. These signal handling values were chosen since we have observed heuristically that results in efficient development of CNNs for audio classification tasks. The sampling frequency of the audio signal is an important variable which can be set to a higher value that might result in a better audio quality, at the cost of increasing the model complexity (number of parameters and size). From this, we have found that 16 KHz is a good tradeoff between audio quality for classification and low complexity aimed to the purpose of deployment as mentioned before for e2e audio classification solutions.

The output of the LLF block creates an image-like tensor that is the direct input to the HLF block. The HLF block is built as a CNN architecture which is the most common approach for computational vision. For AemResNet, we set this HLF stage with a ResNet topology of 18 layers [13]. Details for this ResNet are also shown in Fig 1. The output of its last convolutional layer is average pooled to produce a vector of 512 audio embeddings that represent a condensed representation of the audio sample. This average pool layer brings flexibility when dealing with different lengths of audio inputs, while maintaining the same parameters of the architecture.

The last stage of the AemResNet acts as a classifier, which is the composition of a dropout layer (DO) to reduce overfitting and a fully connected layer with linear activation functions. At the last part of this block, a SoftMax layer is used at the output to present the normalized values based on the number of classes specified.

III. Experimentation

AemResNet was pre-trained over a large set of audio data, this resulted in a pre-trained model that is later fine-tuned based on the audio classification task such as COVID-19 diagnosis based on respiratory sounds. All experimentation was executed using the Pytorch framework [14].

A. *Pre-Training stage*

Both LLF and HLF stages are pre-trained using AudioSet, a large dataset of manually annotated audio events released by Google [15], containing 2.1 million samples equivalent to 5.8 thousand hours of recordings in which 527 different audio classes were labeled. Before using this embedding generator model for a specific classification application, the final classification block is removed, i.e. the fully connected layer, resulting in a 512-dimensional audio embeddings representation as the output. AemResNet used Audioset as pretraining as follows: the single channel raw audio is downsampled to 16 KHz, it is then standardized in amplitude by subtracting the mean and dividing it by the standard deviation of the signal. As well, data augmentation techniques such as random noise addition, random segment cropping of the audio sample, random gain variation and the widely used mixup data augmentation technique. During the training stage, a batch of audio clips were selected randomly into the form of mini batches to train the model. For validation, the complete standardized audio clips were used for inference.

Adam optimizer with a learning rate of $5x10^{-4}$ was used, with a weight decay of $1x10^{-8}$, and a mini batch size of 512 over 80 epochs. Cosine aligned learning rate schedule was used. This audio embedding was trained using the available unbalanced set and validated with the evaluation set for the 527 classes. This audio embedding generator model resulted in 11,744,143 number of trainable parameters, with a mean average precision (mAP) of 0.3690 over the AudioSet evaluation data, and it is the exact same one used in [12].

TABLE I. OPTIMAL HYPER-PARAMETERS FOUND FOR AEMRESNET PER TASK.

Task	Learning Rate	Learning Rate %	Dropout
Task 1	$1x10^{-3}$	80	0.2
Task 2	$1x10^{-3}$	60	0.2
Task 3	$1x10^{-6}$	90	0.9

TABLE II. EXPERIMENTAL VALIDATION RESULTS OBTAINED AS THE AVERAGE ACROSS 5 FOLDS FOR AEMRESNET COMPARED TO OTHER PUBLISHED WORKS. [a] APPROACH PROPOSED IN THIS WORK [b] F1-SCORE COMPUTED WITH EQUATION (3).

		TASK 1			TASK 2			TASK 3		
Model	***Folds***	***Precision***	***Recall***	***F1***	***Precision***	***Recall***	***F1***	***Precision***	***Recall***	***F1***
AemResNet[a]	**5**	**0.6975**	**0.7235**	**0.7332**	**0.8697**	**0.8850**	**0.8773**	**0.9040**	**0.8300**	**0.8654**
SVM [16]	10	0.7200	0.6900	0.7047[b]	0.8000	0.7200	0.7579[b]	0.6900	0.6900	0.6900[b]
Ensemble [17]	3	-	0.7020	-	-	-	-	-	-	-
1D CNN [18]	-	-	-	0.9078	-	-	0.8926	-	-	0.8913
CI-ResNet[19]	10	-	0.7700	-	-	0.5350	-	-	0.7740	-

B. End-to-End CNN for COVID-19 detection

The pre-trained audio embedding generator was used to train a COVID-19 classifier using the commonly adopted transfer learning technique [16]. For this purpose, the Cambridge Crowdsourced dataset described in [6] was used as the target application data. The University of Cambridge launched an application in Android and on a website [17] in which participants are asked to fill demographics general information and symptoms check. The dataset comprises 459 cough and breath samples from 378 users from Web and Android applications until May 2020. These data were annotated by experts and the audio samples were carefully checked to guarantee the quality of the data that contains only cough and breathing. As a preprocessing step, audio data was processed to be single channel with 16kHz sampling rate on a 16-bit resolution, and standardized in amplitude. Both web and Android app sources were used as samples for experimentation, and followed the authors proposal in [6] into three different experimental tasks:

- Task 1. Cough + breath sounds are used to classify COVID-19 vs healthy samples from 66 user (282 samples which represented 32% of the audio samples) and 220 users (596 samples representing 68% of the audio samples), respectively. Where COVID-19 samples included patients with and without cough or symptoms against healthy patients that have not reported symptoms as well as a clean medical history.
- Task 2. Cough sounds are used to classify COVID-19 vs healthy samples from 23 user (54 samples which represented 63% of the audio samples) and 29 users (32 samples representing 37% of the audio samples), respectively. Where COVID-19 samples included patients that reported cough as a symptom, and healthy patients that presented cough as well but have a clean medical history.
- Task 3. Breath sounds are used to classify healthy vs COVID-19 samples from 23 user (54 samples which represented 73% of the audio samples) and 18 users (20 samples representing 27% of the audio samples), respectively. Where COVID-19 samples included patients that reported cough as a symptom, and healthy patients that presented cough as well but have declared asthma in their medical history.

The training strategy followed for this application is similar to the pre-training of the audio embedding generator: Adam optimizer was used with a learning rate of $1x10^{-3}$ for Task 1 and Task 2, Task 3 used $1x10^{-6}$, weight decay of 1e-8, mini batch size of 32 over 400 epochs, cosine aligned learning rate schedule, and warm up of 20 epochs before mixup. It is important to notice that the Cambridge Crowdsourced dataset presents a highly imbalanced number of samples per condition, i.e. there is a significantly larger number of healthy breath and cough samples compared to the COVID-19 ones (approximately 73% against 27%, respectively.). Due to this issue, a focal loss approach was used in the loss function [18] for all of our experiments, resulting in a more efficient training process.

An exhaustive search was executed to find optimal learning rate and dropout values hyperparameters in the classifier block; learning rates from $1x10^{-3}$ to $1x10^{-6}$ and drop out values from 0.1 to 0.9 where explored. Additionally, as observed in previous works [9], [12], we also explored the flexibility of the audio embeddings generator to dynamically adapt to the current target application by allowing to adjust its weights during training. Different learning rate values for the LLF and the HLF were utilized as a percentage of the fully connected layer learning rate; to fine-tune the model, this percentage was swept through different values from 10% to 100% in increments of 10%. The optimal values found for AemResNet across the three different tasks can be found in Table I.

Since there is no suggested official data split available for training/validation of the developed classification models, we randomly defined a set of 5 custom folds with a split 80% of the data for training, and 20% for validation (80/20 split). In all 5 folds, the proportion of available healthy and COVID-19 samples in maintained in both the training and validation split. The model obtained after training the healthy vs and COVID-19 classifier using the pre-trained audio embedding generator resulted in 11,473,282 which is 2.3% less parameters due to a smaller classifier block with only 2 outputs. Additionally, the number of multiply-accumulate operations (MACs) results in 1.84×10^{9}.

For a quantitative assessment of the performance of the proposed AemResNet model, Precision, Recall, and F1-score metrics were used for better understanding of our proposed implementation. These metrics are defined by:

$$\boldsymbol{Prec} = \frac{TP}{TP+FP} \quad (1), \quad \boldsymbol{Rec} = \frac{TP}{TP+FN} \quad (2), \quad \boldsymbol{F1} = 2 \times \frac{Prec \times Rec}{Prec + Rec} \quad (3).$$

In Equations (1) and (2), the TP represents the true positives or the number of correctly classified breath and/or cough sounds into healthy or covid, FP represents an incorrect classification, and FN represents a miss classification. Finally, the computation of the F1-score comptured as in (3) to have a single metric that represents the performance of our model. The experimental results obtained based on the metrics defined above are presented in the following section.

IV. RESULTS AND DISCUSSION

All experimental results obtained with our proposed AemResnet model implementation, using the custom 5-fold random 80/20 splits, are analyzed in this section. To efficiently increase the robustness in the detection of COVID-

19 in respiratory sounds, we leveraged on the use of transfer learning for better performance. Table II presents the performance results of our approach averaged over the defined 5 folds, trained and validated for Task 1, Task 2, and Task 3; this table also shows how the performance obtained by the AemResNet compares to results reported in recent published works that benchmark over the same dataset [6]–[8], [19]. Although these works present their results based on different metrics, we made an effort to consolidate and compare the performance of our approach as much as possible.

We computed the F1-Score from the SVM system in [6] based on the reported Precision and Recall and using Equation (3). From this, it can be observed that AemResNet presents a slightly better F1-Score of around 3.0% for Task 1, but this difference is more significant for Task 2 (almost 12.0%), and for Task 3 (> 17.0%). This suggest that AemResNet can generalize better for COVID-19 detection if only one type of respiratory sounds is considered, i.e., cough or breath sounds in separate models.

Looking at the Recall results, we can compare with the works presented in [6] and [8]. In this context, the Recall metric represents how accurate are the models at correctly classifying healthy and COVID-19 sounds. We found that AemResNet yields better positive classification accuracy in Task 2 (>16.5%) and Task 3 (>5.6%). However, this was not the case for Task 1, where AemResNet results in <4.6% Recall. Lastly, we compared our F1-score results to the ones reported in [19], where AemResNet felt short to the 1D CNN used in their work, particularly for Task 1 (~17.5%). A major difference here could be the use of efficient data augmentation procedures, which would suggest that handling of more data would be expected to be beneficial. We believe we could adopt this type of data augmentation to increase the robustness of our own e2e model and constitutes part of our ongoing research. Overall, the results obtained by AemResNet suggest that the use of the pre-trained deep audio embeddings applied to the task of COVID-19 detection is a robust, convenient, and competitive approach.

V. Conclusion

The experimental results presented in this work prove that AemResNet can be applied to classify breath and cough sounds into healthy or COVID-19 samples, with comparable results to the existing SOTA reported in the literature. The attractive characteristic of this e2e approach is that it avoids the need of additional pre-processing steps for feature extraction at the front-end, thus facilitating its portability into an inference engine. Through the use of pre-trained deep audio embeddings generator, a COVID-19 detection classifier model was build through transfer learning that achieved a F1-score of 0.7332 for cough and breath sounds combined, 0.8773 for cough sounds, and 0.8654 for breath sounds, over the 2020 Cambridge Crowdsourced dataset.

References

[1] "Coronavirus." https://www.who.int/westernpacific/health-topics/coronavirus (accessed Sep. 18, 2021).

[2] A. Manzoor, Q. Pan, H. J. Khan, S. Siddeeq, H. M. A. Bhatti, and M. A. Wedagu, "Analysis and Detection of Lung Sounds Anomalies Based on NMA-RNN," in 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Dec. 2020, pp. 2498–2504. doi: 10.1109/BIBM49941.2020.9313197.

[3] L. Pham, I. McLoughlin, H. Phan, M. Tran, T. Nguyen, and R. Palaniappan, "Robust Deep Learning Framework For Predicting Respiratory Anomalies and Diseases," in 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), Jul. 2020, pp. 164–167. doi: 10.1109/EMBC44109.2020.9175704.

[4] L. D. Pham, H. Phan, R. Palaniappan, A. Mertins, and I. Mcloughlin, "CNN-MoE based framework for classification of respiratory anomalies and lung disease detection," IEEE J. Biomed. Health Inform., pp. 1–1, 2021, doi: 10.1109/JBHI.2021.3064237.

[5] K. Minami, H. Lu, H. Kim, S. Mabu, Y. Hirano, and S. Kido, "Automatic Classification of Large-Scale Respiratory Sound Dataset Based on Convolutional Neural Network," in 2019 19th International Conference on Control, Automation and Systems (ICCAS), Oct. 2019, pp. 804–807. doi: 10.23919/ICCAS47443.2019.8971689.

[6] C. Brown et al., "Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data," San Diego, p. 11.

[7] H. Coppock, A. Gaskell, P. Tzirakis, A. Baird, L. Jones, and B. Schuller, "End-to-end convolutional neural network enables COVID-19 detection from breath and cough audio: a pilot study," BMJ Innov., vol. 7, no. 2, pp. 356–362, Apr. 2021, doi: 10.1136/bmjinnov-2021-000668.

[8] M. A. Nessiem, M. M. Mohamed, H. Coppock, A. Gaskell, and B. W. Schuller, "Detecting COVID-19 from Breathing and Coughing Sounds using Deep Neural Networks," in 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS), Jun. 2021, pp. 183–188. doi: 10.1109/CBMS52027.2021.00069.

[9] P. Lopez-Meyer, J. A. del Hoyo Ontiveros, H. Lu, and G. Stemmer, "Efficient End-to-End Audio Embeddings Generation for Audio Classification on Target Applications," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, Jun. 2021, pp. 601–605. doi: 10.1109/ICASSP39728.2021.9414229.

[10] J. J. Huang and J. J. A. Leanos, "AclNet: efficient end-to-end audio classification CNN," ArXiv181106669 Cs Stat, Nov. 2018, Accessed: Aug. 17, 2021. [Online]. Available: http://arxiv.org/abs/1811.06669

[11] J. Huang, H. Lu, P. Lopez Meyer, H. Cordourier, and J. Del Hoyo Ontiveros, "Acoustic Scene Classification Using Deep Learning-based Ensemble Averaging," in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019), 2019, pp. 94–98. doi: 10.33682/8rd2-g787.

[12] C. A. Galindo-Meza, P. Lopez-Meyer, and J. A. del Hoyo Ontiveros, "Classification of Respiration Sounds Using Deep Pre-trained Audio Embeddings," p. 4.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," ArXiv151203385 Cs, Dec. 2015, Accessed: Apr. 18, 2021. [Online]. Available: http://arxiv.org/abs/1512.03385

[14] A. Paszke, S. Gross, and F. Massa, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024--8035.

[15] J. F. Gemmeke et al., "Audio Set: An ontology and human-labeled dataset for audio events," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Mar. 2017, pp. 776–780. doi: 10.1109/ICASSP.2017.7952261.

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," ArXiv14111792 Cs, Nov. 2014, Accessed: Sep. 18, 2021. [Online]. Available: http://arxiv.org/abs/1411.1792

[17] "New app collects the sounds of COVID-19," University of Cambridge, Apr. 06, 2020. https://www.cam.ac.uk/research/news/new-app-collects-the-sounds-of-covid-19 (accessed Sep. 18, 2021).

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," ArXiv170802002 Cs, Feb. 2018, Accessed: Apr. 18, 2021. [Online]. Available: http://arxiv.org/abs/1708.02002

[19] K. K. Lella, A. Pja, and Department of Computer Applications, NIT Tiruchirappalli, Tamil Nadu, India, "Automatic COVID-19 disease diagnosis using 1D convolutional neural network and augmentation with human respiratory sound based on parameters: cough, breath, and voice," AIMS Public Health, vol. 8, no. 2, pp. 240–264, 2021, doi: 10.3934/publichealth.2021019.