Spring 2021

Deep Learning Based Sound Event Detection and Classification

Alireza Nasiri

Deep Learning Based Sound Event Detection and Classification

by

Alireza Nasiri

Bachelor of Science
Isfahan University of Technology, 2010

Master of Science
University of South Carolina, 2019

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2021

Accepted by:

Jianjun Hu, Major Professor

John R. Rose, Committee Member

Song Wang, Committee Member

Yan Tong, Committee Member

Xinyu Huang, Committee Member

Tracey L. Weldon, Interim Vice Provost and Dean of the Graduate Studies

Acknowledgments

I would like to thank Dr. Jianjun Hu for his guidance and encouragement on my research journey. His insightful feedback and valuable comments were great help in pushing me forward in my research. It is because of his firm belief in me and his support that I have been able to get to this point.

I want to express my sincere gratitude to my dissertation committee members, Dr. Yan Tong, Dr. John R. Rose, Dr. Song Wang and Dr. Xinyu Huang, for their time and advice on my work. It is such an honor to have them serving in my committee.

I would like to acknowledge my collaborators from the Machine Learning and Evolution Lab for all the useful discussions and valuable comments. I want to specially thank Steph-Yves Louis, Dr. Zhonghao Liu, Yuxin Cui, Dr. Yanxia Liu, Jing Jin, Yuqi Song, Rui Xin, and Yong Zhao for their contribution to my research.

Most importantly, I want to express my gratitude to my parents for their unconditional love and support. Their sacrifices and confidence in me have made this possible. Without them I would not have come this far. Furthermore, I would like to express my heartfelt gratitude to my girlfriend Rachel, for her love and support in this journey. Her optimism and reassurances were invaluable through the most difficult moments of this path.

Abstract

Hearing sense has an important role in our daily lives. During the recent years, there has been many studies to transfer this capability to the computers. In this dissertation, we design and implement deep learning based algorithms to improve the ability of the computers in recognizing the different sound events.

In the first topic, we investigate sound event detection, which identifies the time boundaries of the sound events in addition to the type of the events. For sound event detection, we propose a new method, AudioMask, to benefit from the object-detection techniques in computer vision. In this method, we convert the question of identifying time boundaries for sound events, into the problem of identifying objects in images by treating the spectrograms of the sound as images. AudioMask first applies Mask R-CNN, an algorithm for detecting objects in images, to the log-scaled mel-spectrograms of the sound files. Then we use a frame-based sound event classifier trained independently from Mask R-CNN, to analyze each individual frame in the candidate segments. Our experiments show that, this approach has promising results and can successfully identify the exact time boundaries of the sound events. The code for this study is available at https://github.com/alireza-nasiri/AudioMask.

In the second topic, we present SoundCLR, a supervised contrastive learning based method for effective environmental sound classification with state-of-the-art performance, which works by learning representations that disentangle the samples of each class from those of other classes. We also exploit transfer learning and strong data augmentation to improve the results. Our extensive benchmark experiments show that our hybrid deep network models trained with combined contrastive and cross-entropy loss achieved the state-of-the-art performance on three benchmark datasets ESC-10, ESC-50, and US8K with validation accuracies of 99.75%, 93.4%, and 86.49% respectively. The ensemble version of our models also outperforms other top ensemble methods. The code for this study is available at https://github.com/alireza-nasiri/SoundCLR.

Finally, we analyze the acoustic emissions that are generated during the degradation process of SiC composites. The aim here is to identify the state of the degradation in the material, by classifying its emitted acoustic signals. As our baseline, we use random forest method on expert-defined features. Also we propose a deep neural network of convolutional layers to identify the patterns in the raw sound signals. Our experiments show that both of our methods are reliably capable of identifying the degradation state of the composite, and in average, the convolutional model significantly outperforms the random forest technique.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Identifying sounds is one of the major capabilities of the humans. Identifying specific sounds and associating them with certain actions, are stored in humans memory and helps us in the everyday life [145]. We react to the auditory warnings from the environment [159], like when we become fully alert when we hear a glass breaking or dodge when the sound is more similar to a gun shot.

Acoustics have far reaching impact in the scientific community too. Figure 1.1 by [88] shows the famous "Lindsay's Wheel" which depicts the scope of acoustics and its relation with other disciplines such as oceanography, medicine, mechanical, and architectural.

During the recent years, artificial neural networks have been in the heart of the research in many data processing tasks. In these models, high volume of data and computational power are employed in order to identify the patterns in the data. These models, because of the large number of parameters organized in the consecutive layers in them, are called deep learning models. Signal processing is one of the fields that have been greatly impacted by this new surge. Due to availability of the computational resources and large amounts of labeled data, deep learning based methods outperform traditional signal processing methods on a large scale [126].

Sound signal analysis in the shape of speech recognition was one of the earliest fields that used deep learning methods [65]. Shortly after, the environmental sound

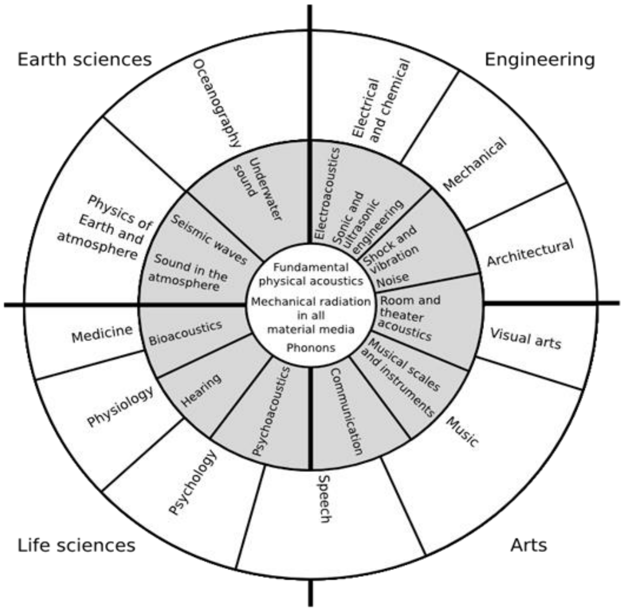


Figure 1.1 Lindsay's Wheel of Acoustics from [88], shows the far-reaching impact of study of acoustics in other scientific fields.

analysis followed the same path indulging the new advances [15, 64]. These studies include a range of tasks. One is sound event detection which is identifying the type of the event in the sound [78, 114]. Here, sound events refer to sound signals that are easily distinguishable from the constant sound of the environment. In the sound event detection, we deal with two different types of labels for the data. One is called the strong labels and the other one is the weak labels. In the strong label, the time boundary for the event is specified, where in the weak labels only presence of the event in the sound file is identified. sound event detection with strong labels can be challenging because there are not many datasets available which have strong labels. The main reason is that strong annotation of the sound can be too expensive since it usually requires a human to listen to the sound and identify the start and the end points of the event.

Another popular task in non-speech sound signal processing is environmental sound classification which identifies the type of the events based on the sound signals

[24, 48, 82, 164]. Environmental sound classification has a variety of applications in autonomous cars [32], robot navigation [81], and surveillance systems [21]. One of the more recent applications of the environmental sound classification is identifying chainsaw sound in the forests [111]. Identifying screaming sound in public and home environments [176, 70] is another practical application of the sound classification.

Due to the noise-like nature of some of these sound events, correctly identifying them, can be very challenging.

1.2 Scope of the Proposed Research

In this dissertation proposal, we focus on three topics:

1) As described before, audio event detection is one of the most important tasks in non-speech audio analysis. With so many work already done in the field of object detection in images, we investigate the adaptation of one of the state-of-the-art models in object detection, Mask R-CNN [58], into audio event detection field. We propose a method called AudioMask, which uses Mask R-CNN to identify potential events in the audio spectrograms and we use a classifier to look into these potential events more closely and analyze them in a finer level to decide whether to confirm those regions as true events or reject them.

2) Environmental audio classification is one of the focus points of the audio analysis studies. We present SoundCLR, our proposed method in using the supervised contrastive loss and the cross-entropy loss to achieve the state-of-the-art results on three benchmark datasets in the environmental sound classification task. In this study, we explore three different training schemes for the audio classifiers which are training with the cross-entropy, supervised contrastive, and the hybrid loss function. In our proposed hybrid loss function, the contrastive loss is applied on the representation space to disentangle the representations of the samples of the different classes from each other, while the cross-entropy function calculates the loss between the output probability values and the ground-truth labels. We run our our experiments using these loss functions and discuss our results on ESC-10, ESC-50 [122], and US8K [140] datasets.

3) Another part of our work is acoustic emission analysis. More specifically, we analyze very short audios generated by the SiC$_f$ - SiC$_m$ composite tubes, during their open-end burst tests to identify the state of the material. There are a group of expert-crafted features for the audios that are traditionally used for acoustic emission analysis. We use random forest with these features to build our baseline model. We also created a deep learning model using convolutional layers. Due the small length of the audios, we use raw waves as the one-dimensional input to the model rather than frequency-time representations. Our experiments shows that both random forest and deep learning models can be a reliable tool in monitoring state of the degradation in SiC composite tubes and in average our proposed convolutional neural network (CNN), outperform the random forest models.

1.3 Structure of the Dissertation

In chapter 2, we provide a brief review of the audio analysis and common research trends, along with some background knowledge on the machine learning methods that we used in our studies. In chapter 3, we present a method to identify the time boundaries of audio events and discuss our experiments and results. In chapter 4, we discuss the environmental audio classification, previous studies and our proposed method, SoundCLR, which is based on contrastive learning of the audio events. Furthermore, we explain our experimental setup and the state-of-the-art results in the environmental audio sounds classification task. In chapter 5, we analyze the short audio signals generated by the SiC composites during its degradation process. We introduce two methods based on the random forest and convolutional neural networks. The conclusion and future work is presented in chapter 6.

The related work to each one of our studies in audio event detection and classification, is presented in the corresponding chapters.

Chapter 2

Background

2.1 Sound Analysis

The research on sound signal analysis can be divided into speech, and non-speech sound analysis. In the speech analysis, the sound contains spoken words of one or more humans and it covers multiple topics. The most popular research topics in the speech analysis includes automatic speech recognition [74, 175], speaker identification [131, 132], emotion recognition [18, 36], identifying stage of some diseases such as Parkinson [50, 105], and noise removal [146, 3]. In the speech recognition, the research is being conducted in making the machines capable of understanding the spoken words. Also, the stage of some diseases such as Parkinson and Alzheimer can be identifies by analyzing the speech of the patients [17, 90].

In the non-speech sound analysis, sound signals contain sounds generated by sources other than humans, such as environmental sounds, sounds generated by animals, and music. The most popular topics in the non-speech sound processing are sound event detection and classification [125, 6, 96], acoustic scene classification [7, 95, 49], sound source separation [166, 165], and automated sound captioning [34, 35]. Since these research topics are more aligned with the scope of our research, we further explain each one of them here with more details.

2.1.1 sound Event Detection and Classification

The task of sound event detection is identifying a dominant event over the background sounds. This task aims to detect the start and end of the sound event in addition to the type of the event [120]. Identifying the type of the event in an sound segment is called sound classification. Figure 2.1 shows the general process in the sound classification and the sound event detection.

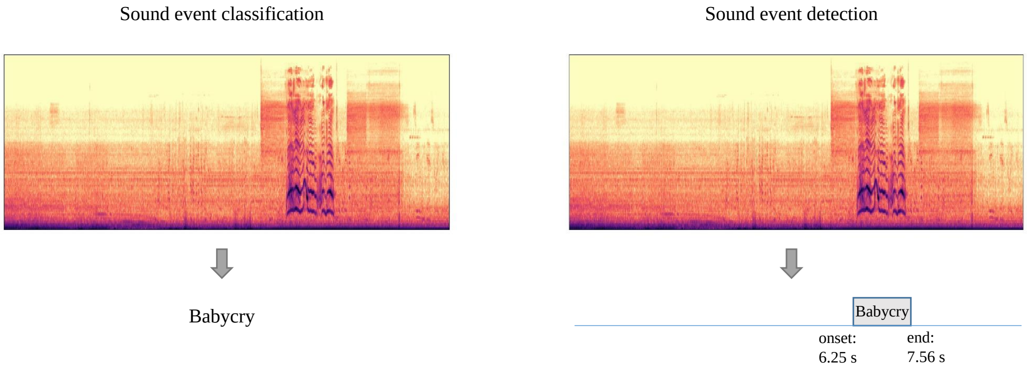


Figure 2.1 Outcome of the sound event classification and sound event detection.

Sound event detection and classification can be used in a variety of areas including wild-life monitoring [91] surveillance [25], and health monitoring [115, 101]. The general approach for detecting the time boundary of a sound event, is to break the sound to smaller segments of about 20 - 40 nanoseconds, and analyze each one of the segments. The conventional methods in machine learning such as Gaussian mixture models, analyze each segment in isolation, where hidden Markov models and recurrent neural networks can capture the temporal dependencies among the segments too.

2.1.2 Acoustic Scene Classification (ASC)

Computational auditory scene analysis was introduced first in 1994 in order to model humans' sound perception [163]. It deals with identifying the acoustic environment based on its audio signal. Scene analysis has great importance in devices that need environmental awareness [128]. The environment can be defined based on the social context or physical location, such as park, street, meeting, and party. The acoustic scene analysis, also has application in video scene analysis and classification [89].

The challenge in the ASC, is that the recording from a specific environment can include sounds from multiple sources where only a few of those sounds provide useful information on the scene of the recording. Some of the research studies in ASC focus on identification and separation of the few informative audio events from the rest of the signal [9]. More recent research trends in this field is shifted towards using deep neural networks to extract the discriminative features rather than using specific filters to calculate the features of the audio segments [116, 118].

2.1.3 Sound Source Separation

Sound source separation refers to the process of separating sounds coming from different sources in a mixture. This fields has a wide range of application such as identifying the noise signal and removing it from the sound, or separating the sound of an mistuned instrument in a music segment and modifying that sound [39]. The sound source separation is a more recent field of research compared to audio event detection and scene classification. Humans are capable of identifying the sound components easily, but the computational modeling of this capability is proven to be limited in their source separation power [167].

The methods to separate the sound sources can be divided to two groups of data-adaptive, and model-based separation [19]. In the data-adaptive technique, there is no prior information on the sources' signals and they are calculated from the data, where in the model-based methods aim to develop a parametric estimation of the source signals.

2.1.4 Automated sound Captioning

The automated sound captioning can be considered as the most recent field of study in the non-speech audio analysis. In this field, the task is to describe the audio with free text. It can be considered as a research field between the audio analysis and natural language processing since the input the model is an audio segment, and the output is a description of the events in that audio segment. The audio captioning here is different from the audio detection, as the model does not output the type of the events or their start and end time, rather it describes the audio segment in the same way that a human would [34].

The small number of studies that have been conducted in this field, are all using deep neural networks to map the audio to their captions [34, 154, 171].

2.2 Sound Representations

Due to the high dimensionality of the raw wave-form of the audio signals and their high sensitivity to the noise, they are rarely used as the input to the algorithm [69, 73, 113]. More compact representations of audio, which reduce the dimensionality by preserving only the most descriptive information, is a more popular choice to be used as an input. In this section, we briefly explain the most common feature extraction methods for the audio signals.

2.2.1 Spectrograms

Short-time Fourier Transform (STFT) is a common method to extract the frequency domain representation of a signal from its amplitude values [144]. A sliding window divides the signal to smaller segments. The signal inside each segment is multiplies by a window function such as Hamming or Hanning to prevent spectral leakage.

The result of extracting the frequency values of the audio segments through the sliding window is 2-D representation, called spectrogram, where each value specifies the strength of a frequency band in a specific time in the audio signal. Figure 2.2 shows an audio wave and its extracted spectrogram.

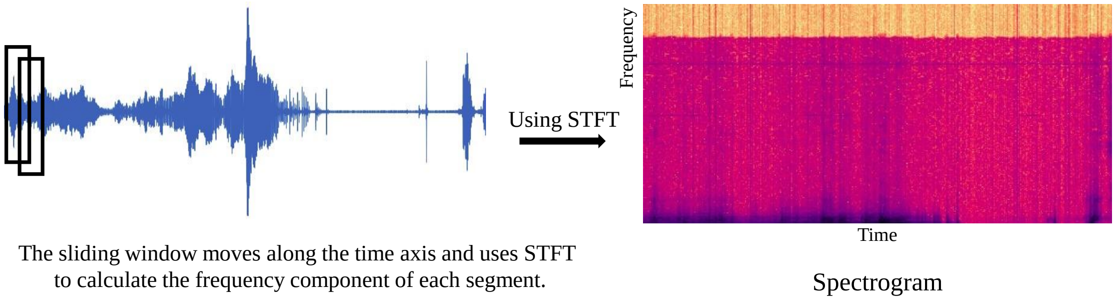

Frequency
Using STFT
Time
The sliding window moves along the time axis and uses STFT to calculate the frequency component of each segment.
Spectrogram

Figure 2.2 Extracting Frequency-Time values from the wave signal using STFT and sliding window.

2.2.2 Log-Scaled Mel-Spectrograms

The mel scale is developed based on the humans hearing sense [150, 160]. We are better in identifying the differences between the lower frequency values rather than the higher ones. Mel scale is designed to reflect this attribute of the human hearing perception and it is comprised of the bands that are smaller for low frequency values and wider for the higher ones. The spectrograms that use mel scale to combine the near-by frequency bins, are called mel-spectrograms.

Furthermore, for human ear to perceive a sound twice as loud, the amplitude of the sound should be about 10 times higher. To use this fact in making the audio representations closer to the human's hearing sense, we calculate the log-scale of the mel-spectrograms. Figure 2.3 shows an example of converting a spectrogram into a log-scaled mel-spectrograms.

2.3 Datasets

In the non-speech audio analysis and specifically in the audio event detection and classification, one of the major issues remains to be the lack of publicly-available large labeled datasets. The current datasets are much smaller compared to the image datasets or even the speech ones.

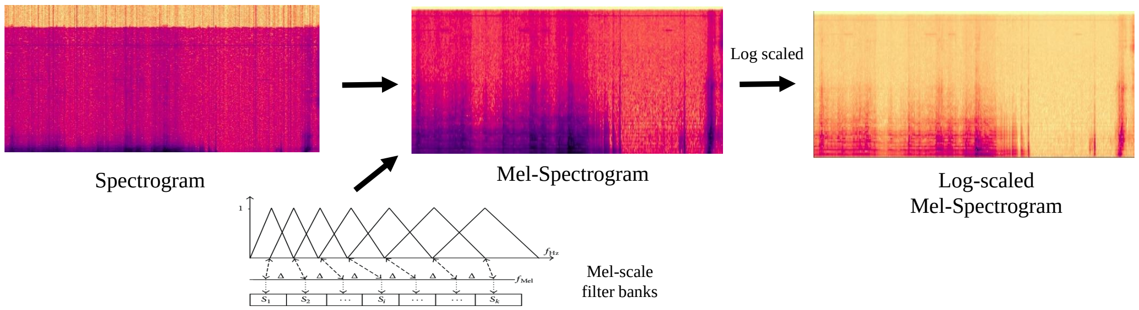


Figure 2.3 Using mel-scale filter banks to create a mel-spectrogram from the spectrogram and then calculating the log of the mel-spectrogram.

Here we introduce some of the most common datasets in the audio event detection and classification.

2.3.1 Audioset

Audioset [47] is the largest dataset in audio event detection with more than 2 million human-labeled sound segments with a hierarchical ontology of 632 classes. Each audio segments is extracted from a YouTube video and is 10 seconds long. This dataset is created by the researchers in Google, and its sound signals are not publicly available due to the privacy issues. Instead a feature vector, which is the output of a pre-trained Resnet model for each audio segment is published, which severely constraints the the utilization of this dataset in the audio analysis researches.

2.3.2 ESC-50 and ESC-10

The ESC-50 dataset [122] includes 2000 audio samples from 50 classes. Each sample is 5 seconds long. The samples are divided among 5 balanced folds. The classes can be divided into five major categories: animals, natural soundscapes and water sounds, non-speech human sounds, interior/domestic sounds, and exterior/urban sounds. The ESC-10 is a subset of ESC-50 with 400 samples in 10 classes.

2.3.3 UrbanSound8K

The UrbanSound8K or in short US8K [140], has 8732 samples from 10 different classes with varying length of up to 4 seconds. The classes are air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The sound segments in US8K, have different sampling rates and can be mono or stereo.

2.3.4 Freesound

Freesound is not a dataset but an online platform to collaboratively create open audio datasets [42]. Sounds are uploaded and labeled by its users. It contains more than 400k audio segments.

2.3.5 FSD50K

FSD50K is a new dataset that was introduced in 2020 [43]. It contains more than 51k audio samples from 200 classes. The samples have been labeled using the Audioset ontology. In total, this dataset contains 100h audio.

2.3.6 Million Song Dataset

The million song dataset contains more than one million contemporary music tracks where most of them are pop/rock songs [10]. This dataset was created for music information retrieval. However, the audios are not publicly available, and only a set of calculated features for each audio file is published.

2.3.7 TUT Acoustic Scenes

This dataset was released for the DCASE 2016 challenge [98]. Audio samples are 30 seconds long and they belong to one of 15 acoustic scenes: lakeside, beach, bus, cafe/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, urban park, residential area, train, and tramway. The dataset is divided to 4 folds for cross-validation.

2.3.8 GTZAN

GTZAN is primarily used in music genre classification and is composed of 1,000 music audio excerpts, each 30 seconds long, labeled in ten categories [161]. In addition to its small size, it was also discovered that it suffers from repetition and mislabeling [152].

2.4 Conventional Machine Learning Methods in Audio Event Detection and Classification

The conventional machine learning methods in this field includes Gaussian mixture models (GMMs) [6, 96, 183], hidden markov models (HMMs) [181, 115], support vector machines (SVMs) [22, 53, 155], and random forests (RFs) [119, 177]. We briefly review each one of these methods and their application in sound event detection and classification here.

Gaussian Mixture Models (GMMs) are generative models which identify the probability density of the components within a mixture. Expectation Maximization [102] is a common method in identifying the probability distribution of each one of the components. In recognizing sound events, each sound class acts as a component and the model is trained to identify the parameters related to the distribution of each one of the sound classes [6]. Since in GMM, each segment of the audio signal is processed in isolation, it does not capture the temporal dependencies in the signal. In [61], Hidden Markov Models (HMMs) are used to capture the context information to detect the audio events. A HMM considers previous audio segments along with the one that is being processed, to classify the current segment [143].

Support Vector Machines (SVMs) are discriminative models that separate data samples using hyperplanes in the high-dimensional space [60]. SVMs are applied in audio event classification where a sliding-window breaks the audio into smaller segments and each segment is classified independently [155, 123]. However, SVM scales superlinearly with the size of the dataset and is not able to handle large scale data efficiently.

Random forest (RF) is a machine learning method that uses an ensemble of decision trees, and can be used for classification or regression [13]. Each decision tree if a RF model is trained with a subset of the training data, independently. In inference, each tree outputs a label for the input sample and the final prediction is calculated by the majority voting technique. The RF models are used to classify the audio event classification and detection [119, 177], where a group of calculated features for each audio segment is used to train the RF model to identify the event in that segment.

In the recent years, deep neural networks have become the dominant method in audio event detection and classification [28, 173]. We will briefly introduce the common layers in the neural networks later in this chapter and we will review their application in audio event detection and classification in the chapters 3, 4, and 5.

2.5 Deep Learning

Artificial Neural Networks (ANNs) are inspired by how neurons in human brain perform various cognitive functions [124]. In the humans, different signals activate different neurons. The neurons map the input signal to a cognitive representation. In ANNs, each neuron computes a weighted sum of its input signal and passes the result through a linear or non-linear activation function. Similar to the human brain, the input signals passes through the layers of inter-connected neurons to be mapped to the output signal. The first layer of neurons that receives the input signal is called the input layer, and the last layer that generates the output signal, is called the output layer. All of the layers between the input and the output layers, are called the hidden layers. Figure 2.4 depicts the architecture of a multi-layer preceptron (MLP) model with two hidden layers.

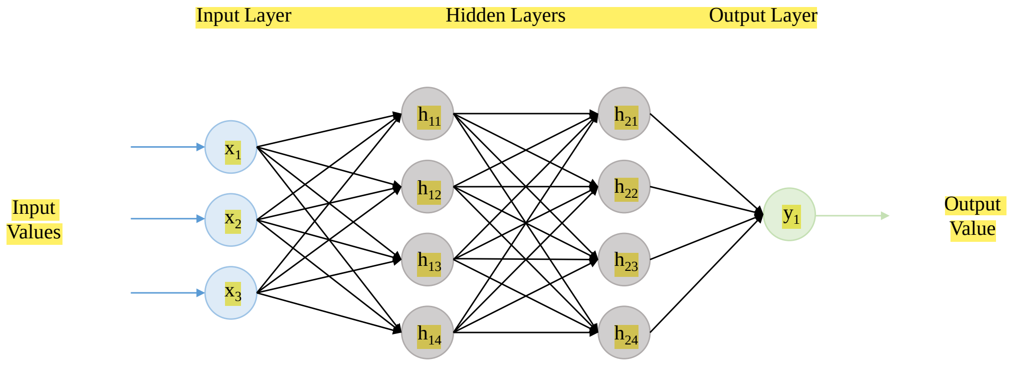


Figure 2.4 Architecture of a MLP model. The size of input is 3 and there is only 1 output value. Each hidden layer has 4 neurons.

The origins of the ANNs can be traced back to 1943, where McCulloch and Pitts created a linear model to recognize two different types of input [92]. In 1958, Rosenblatt proposed an algorithm for pattern recognition, perceptron, with two learning layers [134]. The backpropagation algorithm formalized in [135], was applied to modify the weights of the connections in the model to minimize the difference between the calculated output and the desired one. The next breakthrough happened when in 2006, Geoffrey Hinton showed that multiple stacked layers of neural networks can be trained using a method called greedy layer-wise pre-training [66]. The term "deep learning" or "deep neural networks" started to be commonly used to refer to the neural nets with two or more hidden layers. By adding more layers and more units within a layer, a deep network can represent functions of increasing complexity [51]. One constraint for training deep models with millions of parameters is the need for large datasets. In the supervised learning scheme, having a large number of labeled data makes it possible for the model to learn the mapping function from the input data samples to the output values. More recently, the power of internet have made it easier to create large labeled datasets. ImageNet with about 3.2 million labeled images first was publicly presented in 2009 [30], and by now, the number of its samples has grown to about 14 million.

The availability of the large datasets, made it possible to train deeper and larger neural nets. In 2012, AlexNet [77], a model with 5 convolutional layers and 3 fully-connected ones, won the 2012 ImageNet LSVRC-2012 competition by a large margin compared to the second one. From then, neural networks with larger number of parameters and more stacked layers have substantially increased the performance of the models in different tasks. For example, the ResNet model introduced in 2016 [57], can have up to 152 layers and 60 million parameters. The ResNet-152 has improved the top-1 classification accuracy on the ImageNet to 78.3 %. This achievement is more impressive if we consider that only 4 years prior, AlexNet achieved the state-of-the-art results at that time on the ImageNet with the top-1 classification accuracy of about 57 %.

2.5.1 Hyperparameters

Hyperparameters define the structure of the model, along with the training methods in the deep neural networks. Determining appropriate values for these hyperparameters can directly affect the performance of the model [71]. Here, we describe some of the most important hyperparameters.

- **Batch-size** is the number of training samples that would be used as the input to the model, in the training stage. The loss value will be calculated based on the difference between the desired output and the calculated one for all of the samples, and the parameters of the model will be updated based on that loss signal. So, if the batch size is too small, the training process will become unstable, and if the batch size is too large, then the loss signals of the samples can cancel each other and the model won't be trained to perform its best.

- **Learning-rate** identifies the step size that is used to update the parameters of the model at each iteration, in order to minimize the estimated error. Too small learning rate can result in a longer training process and too large value for the learning rate can cause the model to diverge or converge to a sub-optimal parameter values.

- **Number of hidden layers** is closely related to the complexity of the function that the model is learning. Higher number of the layers, means that the function can identify more complex relations between the input and the output. Also, to train a deeper model, more data is required too.

- **Number of epochs** identifies the number of the iterations in the training stage. At each iteration, model is exposed to all of the samples in the training dataset and adjusts its parameters based on the calculated loss value. These iterations should continue till the model has reached a stable performance and the values of the parameters are not changing anymore. One important issue to consider here, is that the model should not be overfitted on the training set. Overfitting means the model has learned the patterns in the training set so well, that it does not have a generalization power and cannot perform well on the unseen data. Overfitting can happen either by training the model with too many iterations, or because the training dataset is not large enough.

2.5.2 Fully-Connected Neural Networks

The neurons in a fully-connected layer are connected to all of the activation values from the previous layer. The major advantage of these layers is that they are "structure agnostic", which means they don't make any assumptions about the structure

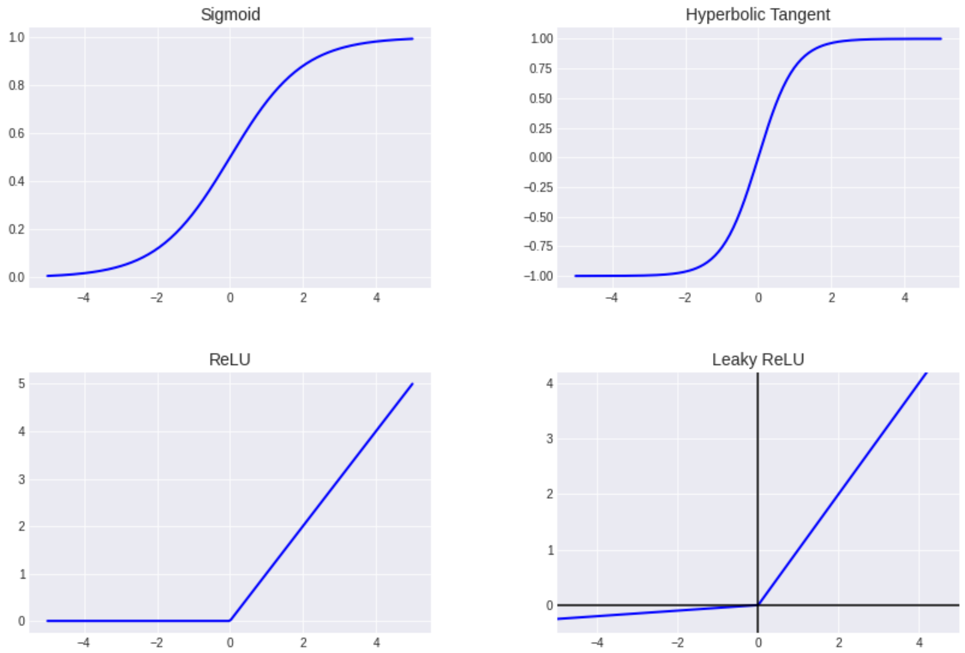


Figure 2.5 Activation functions are used to increase the capacity of the neural networks in identifying non-linear relations between the input and output values. The most commonly used activation functions are sigmoid, Hyperbolic Tangent, ReLU, and Leaky ReLU.

of the input data [129]. Assuming h^{l-1} is the input to the l-th layer, the output of this layer, h^l is calculated using this formula:

$$h^l = \sigma(W^l h^{l-1} + b^l) \tag{2.1}$$

where W^l and b^l are the weight and bias parameters of layer l, and σ is the activation function. The activation functions are continuous, differentiable, and non-linear functions that increase the capacity of the neural networks in learning the non-linear relation between the input and the target values. Figure 2.5 shows the four most common activation functions.

2.5.3 Convolutional Neural Networks (CNNs)

Convolutional layers were first used in the visual recognition tasks [77, 148]. These layers use a number of kernels to convolve over their input by striding the kernels over them. The parameters of the kernels are shared among the different parts of the input value which substantially reduces the number of the parameters compared to the fully-connected layers. Convolving the input with each one of the kernels, creates a feature map which indicates the location and the strength of a specific feature in that input. The pooling layers are usually applied to the output of the convolutional layers in order to reduce the size of the feature maps and make the detection of the features in the input shift-invariant [141]. Two most common pooling layers are max-pooling and average-pooling layers which maps a number of values in the feature map to their maximum, and average values, respectively [12, 174]. More recently, attention pooling, which calculates weights for different parts of the input, is also shown to improve the performance of the model [38, 84].

In designing the convoltional neural networks (CNNs), usually convolutional layers are followed by the pooling layers. In these models, the early layers detects basic features and by going deeper in the model, those basic features are combined to detect more complex features. Figure 2.6 shows the structure of a CNN, where pooling layers reduce the dimensionality of the feature maps.

2.5.4 Recurrent Neural Networks (RNNs)

The RNNs are usually used in analyzing the sequential data such as text or audio, in order to capture the dependencies among the values in the sequence [100, 52]. In addition to the current input, recurrent layers also use the activations from the previous time-frames. The recurrent layers can also be bi-directional, where the past and future activations are used in addition to the current input, to calculate the output for a specific time-frame [142].

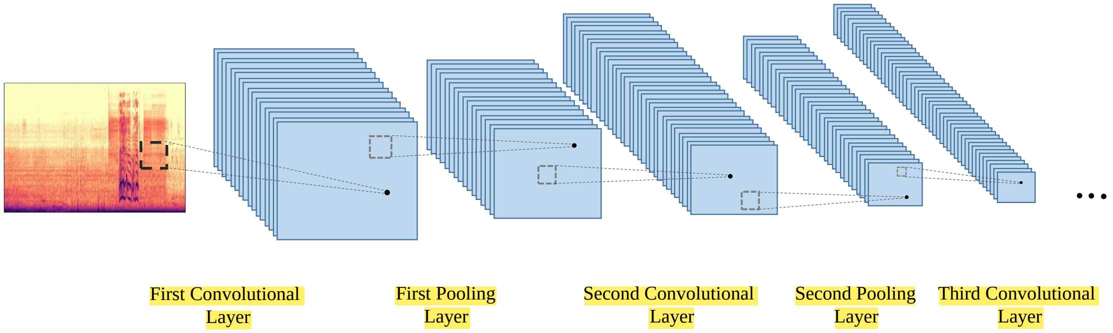

First Convolutional Layer
First Pooling Layer
Second Convolutional Layer
Second Pooling Layer
Third Convolutional Layer
...

Figure 2.6 Architecture of a typical CNN model with the pooling layers after each convolutional layer.

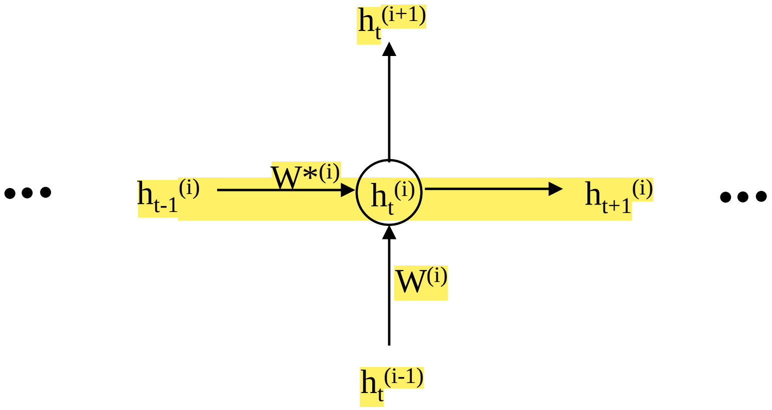


Figure 2.7 An unfolded recurrent neural layer, where the input from the previous layer along with the activation of the previous time-step is used to calculate the output of the layer for the current time-step t.

In a typical RNN, assuming $h_t^{(i-1)}$ to be the input to the i^{th} hidden layer at time-step t, the output $h_t^{(i)}$ is calculated as:

$$h_t^{(i)} = \sigma(W^{(i)} h_t^{(i-1)} + W^{*(i)} h_{t-1}^{(i)} + b) \tag{2.2}$$

where $W^{(i)}$ is the weight matrix between layer $i-1$ and i, $W^{*(i)}$ is the weight matrix for the output of the layer i for the past time-step, b is the bias, and the σ is the activation function. Figure 2.7 shows the unfolded recurrent neural layer that receives the activation of the previous layer for the time-step t and the activation of the previous time-step to calculate the output.

Chapter 3

AudioMask: Robust Sound Event Detection Using Mask R-CNN and Frame-Level Classifier [1]

[1] Alireza Nasiri, Yuxin Cui, Zhonghao Liu, Jing Jin, Yong Zhao, and Jianjun Hu. 2019. Proceedings of IEEE International Conference on Tools with Artificial Intelligence. 485-492. Reprinted here with permission of publisher.

3.1 Introduction and Related Work

Sound event detection (SED), which deals with sound analysis in order to identify everyday audio events, has received a lot of attention recently. It has many interesting applications in environmental and wildlife surveillance systems[24, 14], smart homes [56, 29], and video event detection [170]. SED is also used in autonomous driving to prevent the accidents in the occasions that the visual detection of the object is difficult. Some examples of this are detecting sirens in the road [32] and identifying dangerous situations like car crashes and tire skidding [40]. SED is also used in detecting scream and loud speech in subway trains [79]. Different signal processing and machine learning techniques have been proposed for SED. The initial approaches usually used mel frequency cepstral coefficients (MFCCs) as features with algorithms like hidden Markov model [96], Gaussian mixture models [62], non-negative matrix factorization [46, 99, 180], support vector machines (SVM) [156], and random forest [119].

Deep learning (DL) models have become very popular in detecting sound events in recent years [85, 75, 16]. These DL models either use frame-level approaches or region-based ones. In the frame-level approach [85, 16], they analyze every small frame in audio to determine if it belongs to an event or not. The architectures of these models are usually based on convolutional and recurrent layers. Convolutional layers are applied to extract high-level features before recurrent layers are applied to learn the longer term temporal dependencies among them. One of the biggest downsides of the frame-level methods is that they fail to consider the longer contextual dependencies in the audio. This is addressed by the second group of DL-based SED models: the region-based approaches [169, 75]. These algorithms treat mel-spectrograms of audio as images and use object detection models from the computer vision field, to identify segments of audio that belong to different types of events. Wang et. al [169] used a region-based approach by employing a slightly modified version of R-FCN to detect rare sound events. R-FCN is a fully convolutional neural network for object detection [26]. Another region-based approach is proposed by Kao et. al [75]. They used a network called CRNN, which has a similar architecture to the object detection algorithm Faster-RCNN [130], to get the regions of interest and classify them as the event/not-event. Region-based models can be useful in addressing some of the main challenges in SED according to [120], which is creating tight boundaries around the event considering the background noise. These models detect events by finding their patterns in log mel-spectrograms. This strategy might be problematic because some events do not have very distinguishable shapes such that these methods might end up with non-trivial number of false positive predictions.

In this study, we propose AudioMask, a novel algorithm for rare sound event detection by taking advantage of both region-based and frame-based methods. To identify candidate event-regions from the audio, we first take a region-based approach by using Mask R-CNN model [58] as a region proposal network to identify potential event-regions. Mask R-CNN is a state-of-the-art object detection and segmentation model that outperformed all of the winners of the Common Objects in Context (COCO) 2016 challenge [86]. It has a special capability to create tight and highly accurate boxes around the target objects, which allows us to identify candidate audio events with tight, pixel-accurate bounding boxes. After potential event-region detection, we then analyze each of these regions by looking into the small frames inside them using a frame-level classifier composed of several convolutional and recurrent layers. This technique achieves competitive results in detecting the audio events with highly variant and non-regular shapes in log-mel spectrograms. The contribution of this study includes the following:

- We take advantage of Mask R-CNN, a state-of-the-art object detection model in computer vision, to propose regions of audio as candidate event-regions that have similar patterns to the target events.

- We introduce a frame-level classifier, based on the combination of convolutional and recurrent layers to analyze frames in each candidate segment and identify the true event-regions out of the proposed ones identified by the Mask R-CNN model.

- Experimental results indicate that our AudioMask algorithm outperforms all of the non-ensemble methods of DCASE 2017 [97] challenge and improves the baseline model by 13.3% in F-score.

The rest of this chapter is organized as: In section 3.2, we show an overview the components of the AudioMask. Then we discuss the components which are feature extraction, region-proposal by Mask R-CNN, frame-level classifier, and post-processing in sections 3.3, 3.4, and 3.5. Section 3.6 goes over the experimental setup, results and their analysis. Finally we conclude the chapter in section 3.7.

3.2 AudioMask: Overview

We introduce AudioMask, a novel method that takes advantages of both region-based methods and frame-level analyses. Our algorithm consists of four stages: 1) extracting log mel-scaled energies from audio, normalizing and pre-processing them, 2) training a Mask R-CNN to identify chunks of the audio that potentially belong to the target event, 3) training a frame-level classifier that analyzes the frames inside each candidate segment of the audio and outputs a probability value for them, and 4) post-processing based on the confidence values generated via Mask R-CNN and probabilities from the frame-level classifier to identify the true event-regions. Fig 3.1. shows the general framework of our proposed AudioMask algorithm.

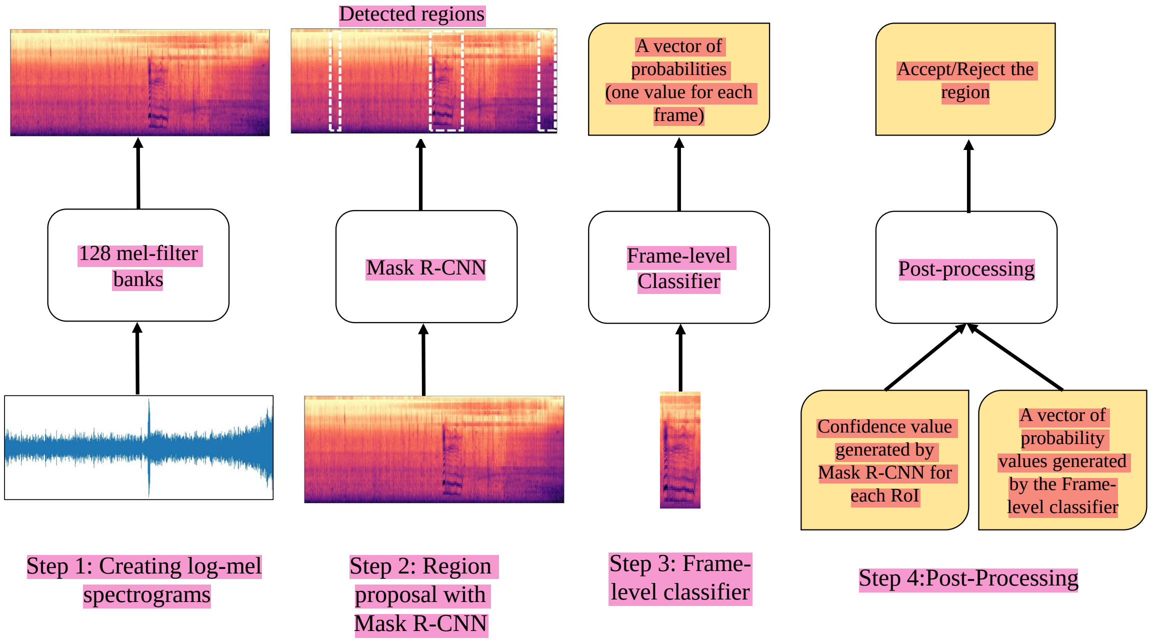

Detected regions
A vector of probabilities (one value for each frame)
Accept/Reject the region
128 mel-filter banks
Mask R-CNN
Frame-level Classifier
Post-processing
Confidence value generated by Mask R-CNN for each RoI
A vector of probability values generated by the Frame-level classifier
Step 1: Creating log-mel spectrograms
Step 2: Region proposal with Mask R-CNN
Step 3: Frame-level classifier
Step 4:Post-Processing

Figure 3.1 Four steps in AudioMask. Each detected region by Mask R-CNN is mapped to a segment of the audio with a fixed length on the mel spectrogram, before being fed to the frame-level classifier

3.3 Feature Extraction

We use pre-processed and normalized log-scaled mel spectrograms as input features to our model. Spectrograms are 2D representations of audio in time-frequency domain with brightness or color representing strength of signals of specific frequencies in that time frame. They preserve more information than most of hand-crafted features [172]. The log-mel feature-map exhibits locality in both time and frequency domains [1].

Mel-spectrograms have proven to be good features of audio for many deep learning based sound event detection methods [85, 169, 75, 16]. Their difference with normal spectrograms is that log mel-scale filter banks are used in them to imitate the non-linear human ear perception of sound [85].

We apply a window size of 46 ms, being overlapped with half of its size to the audio signal to extract 128 mel-filter banks for each frame of the audio signal. The window size and number of the mel-filter banks are the same as [10]. We then calculate the logarithms of these filter bank values and normalize them.

3.4 Region-Proposal with Mask R-CNN

We use Mask R-CNN to detect precise bounding boxes around areas of the spectrogram that possibly corresponds to a target event. Mask R-CNN is a fully convolutional network that has shown great performance in object detection and segmentation in computer vision. It outputs a class label and a tight bounding-box offsets for each candidate object, along with a mask for each object. Fig 3.2. shows how Mask R-CNN works in general.

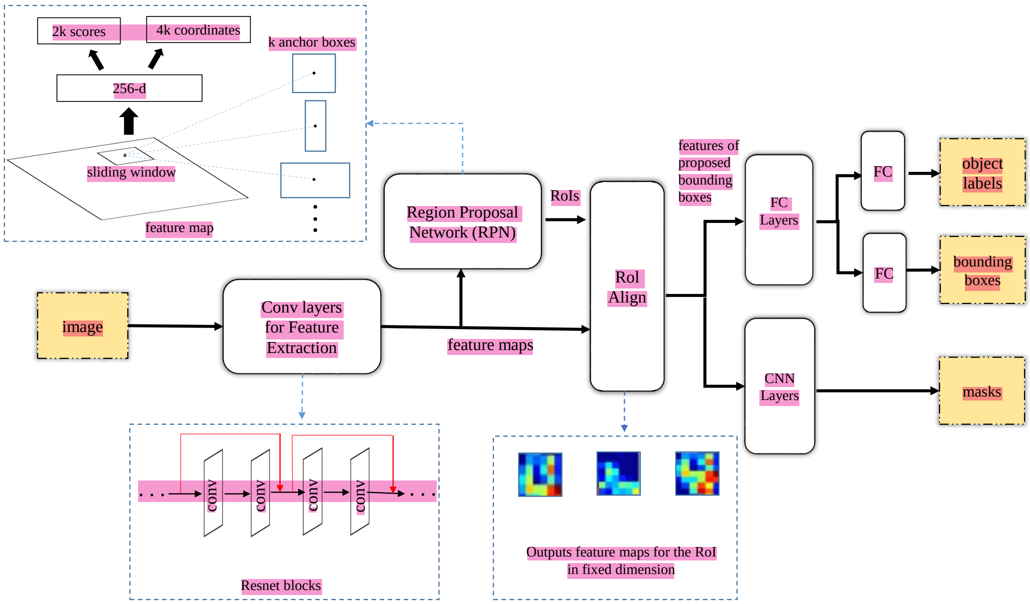


Figure 3.2 Mask R-CNN framework for object detection

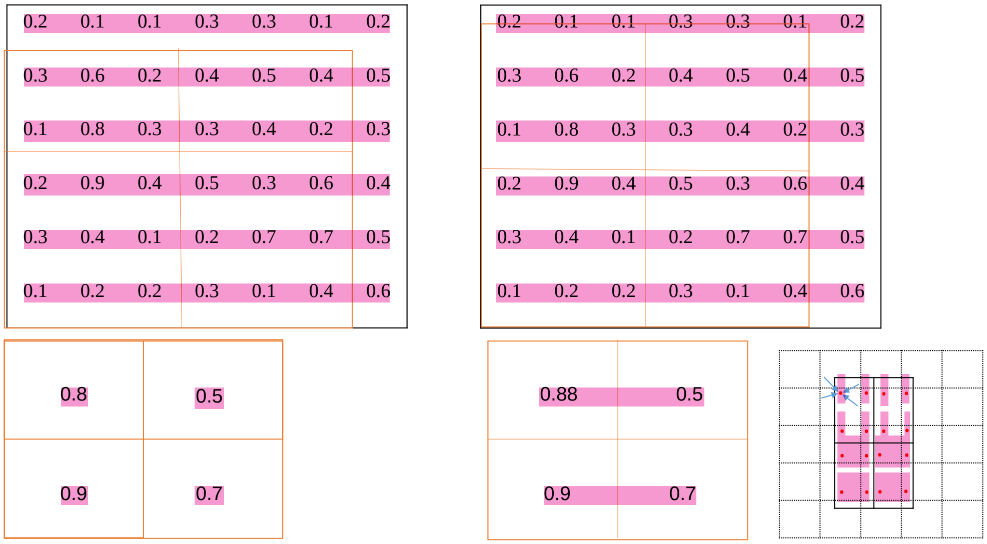
Mask R-CNN framework for object detection consists of two stages: Region Proposal Network (RPN) is the first stage which outputs approximate bounding boxes for potential objects in the image. The second stage is responsible for predicting object class labels, tightening bounding boxes and creating object segmentation masks.

In RPN stage, a backbone of the residual blocks of ResNet-101 network is used to extract high-level features out of the images. Later, a small network slides over this feature map to produce a set of region proposals along with an objectness score for each region. This small network predicts k region proposals at each sliding-window location, by outputting 4k values as coordinates and 2k values as probability estimates of object / not-object for each proposed rectangular region [130].

Poposed regions can be of different dimensions and need to be represented with a fixed-length feature map before being fed to the fully connected layers. Mask R-CNN uses a method called RoIAlign to create fixed-size feature maps for each region while ensuring a close correspondence between the values on the feature map and the regions on the actual image. In RoIAlign, when a RoI is not aligned exactly with the values in the feature map, instead of using harsh quantization and mapping the RoI to the discrete granularity of the feature map as done in Faster R-CNN[130], bi-linear interpolation is used to compute more accurate values of the features [58] . Fig 3.3. shows an example of how RoIAlign uses bi-linear interpolation to create a better estimation of features for the boxes of the objects.

Mask R-CNN through the RoIAlign procedure, makes sure that the masks and the bounding boxes have a pixel-to-pixel alignment on the real objects in the image. This makes non-trivial improvements in object detection accuracy [58]. This feature of Mask R-CNN is a major reason that makes it very suitable for SED, in which detecting the exact onset of the event is crucial. Indeed, according to the evaluation metrics of sound event detection as DCASE 2017 did [97], onset detection is the defining factor in correctly detecting an audio event.

A major module of Mask R-CNN is a parallel branch for segmentation mask prediction as shown in Fig 3.2. This feature is believed to improve the performance of Mask R-CNN in object detection in images. We did not utilize the segmentation branch because generating segmentation masks for audio events like ‘gunshot’ is problematic as there is no clear borders for audio events and are also not provided in any of the known data sets.



0.2	0.1	0.1	0.3	0.3	0.1	0.2
0.3	0.6	0.2	0.4	0.5	0.4	0.5
0.1	0.8	0.3	0.3	0.4	0.2	0.3
0.2	0.9	0.4	0.5	0.3	0.6	0.4
0.3	0.4	0.1	0.2	0.7	0.7	0.5
0.1	0.2	0.2	0.3	0.1	0.4	0.6

0.8	0.5
0.9	0.7

a. RoIPooling

0.2	0.1	0.1	0.3	0.3	0.1	0.2
0.3	0.6	0.2	0.4	0.5	0.4	0.5
0.1	0.8	0.3	0.3	0.4	0.2	0.3
0.2	0.9	0.4	0.5	0.3	0.6	0.4
0.3	0.4	0.1	0.2	0.7	0.7	0.5
0.1	0.2	0.2	0.3	0.1	0.4	0.6

0.88	0.5
0.9	0.7

b. RoIAlign

Figure 3.3 (a) RoIPooling used in Faster R-CNN; (b) RoIAlign used in Mask R-CNN. RoIPooling gets the bounding box for each RoI and divides it to smaller sub-windows by quantization, which is mapping the borders of these sub-windows to the closest cells on the feature map. Then it max-pools each one of these sub-windows to get a feature representation for the RoI. The quantization process will cause misaligned feature values for RoIs. RoIAlign avoids this problem by using bilinear interpolation. The bottom right picture shows how this process works in general [58]. The dashed grid is the feature map and the solid one represents an RoI and red dots are 4 sampling points in each bin. RoIAlign computes the value for each one of these bins, by bilinear interpolation from the overlapping grid cells. Notice how the interpolation operation in RoIAlign changes the maximum feature value for the top left cell from 0.8 to 0.88.

We train a Mask R-CNN model for each event using log-mel spectrograms extracted from the audio files. We utilize Mask R-CNN to generate bounding boxes around spectrogram areas that show high resemblance to the target event. We identify these regions of the spectrograms as the potential event-regions that might contain the event. Since we define target objects as audio events that can occur with variant lengths across the time axis, we modify the Mask R-CNN models to only generate rectangular bounding boxes that cover all of the frequency bins. In other words, all of the potential event-regions, have the same height as the frequency bins in the log-mel spectrogram and they cover variant ranges on the time axis. Mask R-CNN assigns a confidence value to each of these boxes based on their similarity to the general pattern of the target audio event.

After training, we choose a model that has the lowest validation loss to be used on the test stage. During the test stage, we collect all of the event-regions with a confidence value above a specific threshold. A list of these proposed regions will be forwarded to the frame-level classifier.

At the end of this step, we have a list of candidate event-regions on the log-mel spectrograms for each target event. Now we need to filter out true events out of them, which is done by the frame-level classifier.

3.5 Frame-Level Classifier and Post-Processing

While Mask R-CNN generates a large number of potential event-regions with tight boundaries, it may also report many false positives. This is partially due to the non-regular shapes of the events in the time-frequency context that can lead to detecting parts of the background noise as the target event. This problem is more severe for short events like gunshots which have highly variant patterns in terms of their mel-spectrogram representations.

To filter the real events out of the proposed event-regions, we train a classifier that receives segments of the audio represented as log-mel values and determines if that region belongs to a specific event or not. These segments are consisted of log-mel frequency vectors of a number of consecutive time-steps. We call each one of these vectors, a frame. Each frame gets a label of 0 or 1, based on whether it belongs to the event or to the background noise, respectively. The output of the frame-level classifier is a one-dimensional vector that consists of probability of belonging to the target event for all of the frames in the segment.

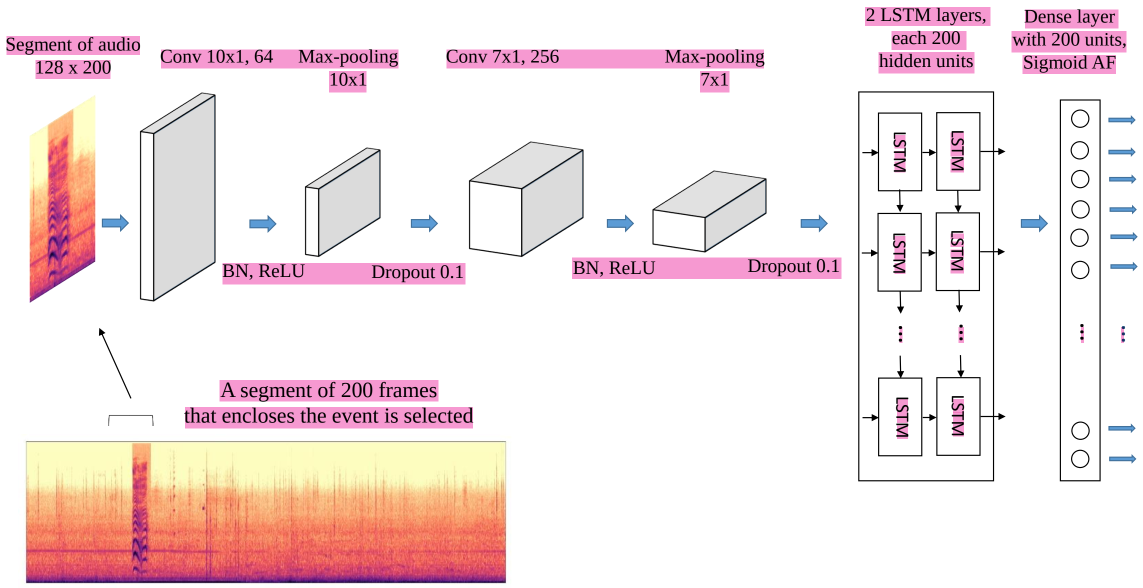


Figure 3.4 Frame-level classifier

This classifier consists of two convolution layers, two LSTM layers and a fully connected layer. The convolution layers have 64/256 one-dimensional filters of size 10/7 that stride over the mel-spectrograms across the frequency axis. Each convolution layer is followed by a batch normalization(BN), rectified linear unit (ReLU) activation function, and a max-pooling layer. The size of filters in the first and second max-pooling layers are 10 and 7, respectively. A dropout layer with a rate of 0.1 is applied to the outputs of the max-pooling layers.

The LSTM layers are used to capture the temporal context across the segment using the high-level features extracted by the convolution layers. The last layer of the segment classifier is a fully connected layer (FC) with Sigmoid activation function that produces a probability value for each frame in the segment. Our segment classifier model is inspired from the model in [85] , although we observed that using 2 convolution layers with smaller filter sizes has better performance. Fig 3.4. shows the architecture of our frame-level classifier.

In the post-processing step, we utilize both block-level analysis by Mask R-CNN and frame-level analyses by the classifier to detect the true event boundaries. Frame-level classifier outputs a probability value for each frame in the segment. In the post-processing step, we calculate the average of the probability values of the frames to obtain a single probability value for the whole segment. We refer to this value as α. Also, we represent the confidence value calculated by the Mask R-CNN for each event-region as β. We use formula (1) to utilize both probability values from the frame-level classifier(α) and confidence values from Mask R-CNN(β) to calculate a score value ($0 \leq D \leq 1$) for each segment.

$$D = \lambda * \alpha + (1 - \lambda) * \beta \tag{1}$$

In the DCASE 2017 benchmark dataset, each audio file can have at most one event in it. So we compare the D values of all of the proposed regions for one audio and choose the event-region with the highest score as the true boundaries for the target event. We also reject the regions with a D value below a certain threshold, which can be determined using the training set.

3.6 Experiments and Discussion

3.6.1 Data

We use the data set from task 2 of the DCASE 2017 challenge to demonstrate the performance of our method. This dataset consists of isolated recordings for three target events including ‘babycry’, ‘glassbreak’ and ‘gunshot’. It also contains recordings of 15 different audio scenes to be used as background sounds. We used the synthesizer provided by the task 2 of DCASE 2017 challenge to create 5000 monophonic audio files for each event with 44.1 KHz sampling rate and event-to-background ratio (EBR) of -6, 0 and 6 dB.

To ease the problem of data imbalance, we set the probability of event being present in each audio to one as the DCASE challenge lets this modification for the training set. We also use a validation set of 1000 audio files for each event. We evaluate our method on the development and evaluation set provided by task 2 of the DCASE 2017 challenge.

3.6.2 Evaluation Metrics

We use event-based error rate(ER) and F-score to evaluate our method [94]. These criteria can be calculated based on counting the true positives(TP), false positives(FP) and false negatives(FN). In the event-based metrics, there is not any meaningful true negatives(TN) [94]. Based on the description of the task 2 of DCASE 2017 challenge [97], the challenge considers a detected event to be a TP, if its detected onset time would be within 500 ms collar of the actual onset time.

Error-rate and F-score are defined as:

$$ER = \frac{FN + FP}{N} \quad \text{and} \quad F = \frac{2TP}{2TP + FP + FN}$$

We have used the sed-eval toolbox provided by the challenge organizer to calculate these metrics [94].

Table 3.1 AudiMask's performance with different λ values on DCASE 2017 development set

	babycry		glassbreak		gunshot		Average	
	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER
$\lambda = 0$	90.8	0.18	76.6	0.51	62.8	0.86	76.7	0.52
$\lambda = 0.25$	91.2	0.17	87.2	0.26	70.2	0.66	82.9	0.36
$\lambda = 0.5$	92.5	0.14	87.2	0.26	72.0	0.55	83.9	0.32
$\lambda = 0.75$	92.4	0.15	88.2	0.24	73.5	0.50	84.7	0.30
$\lambda = 1$	91.6	0.16	83.8	0.32	72.3	0.51	82.6	0.33

Table 3.2 Performance comparison of our models versus best non-ensemble methods on DCASE 2017 Evaluation set

	babycry		glassbreak		gunshot		Average	
	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER
AudioMask (Our method)	90.2	0.19	91.2	0.18	76.4	0.46	85.9	0.28
DNN/CNN [117]	85.7	0.28	88.8	0.22	81.6	0.33	85.3	0.28
SLR-NMF[180]	91.4	0.17	89.1	0.22	72.0	0.55	84.2	0.31
R-FCN [169]	87.3	0.26	91.5	0.16	67.2	0.53	82.0	0.32
Baseline [97]	70.7	0.57	81.0	0.36	66.0	0.57	72.6	0.50

3.6.3 Results and Discussion

The synthesizer for DCASE 2017 task 2 puts events inside 30 seconds long audio files with specific start and finish times for the event. We use these start and finish times to create rectangular bounding boxes for the purpose of training a Mask R-CNN model for them. The height of the bounding boxes is fixed and is equal to the number of mel-bands we used to create mel-spectrograms and their width is equal to the length of the event.

For each event type, we train a Mask R-CNN model with ResNet-101 backbone for 50 epochs and save the model at each epoch. We have modified the RPN module in the Mask R-CNN to only generate rectangular regions with the fixed height of 128 to cover all of the frequency bins and variant width of t. We choose the model with the least error on the validation set and run it on the test set to generate a list of regions that potentially contain the target event with a confidence value of 0.6 or higher. We have chosen 0.6 as the detection threshold because we found that almost all of the events in the development set, are detectable by Mask R-CNN at this threshold level.

Gunshot is the only event type that Mask R-CNN is unable to detect all the true instances for it. We observed that lowering the confidence threshold on the development set did not lead into proposing more true regions by the Mask R-CNN and it rather increased the number of falsely detected regions.

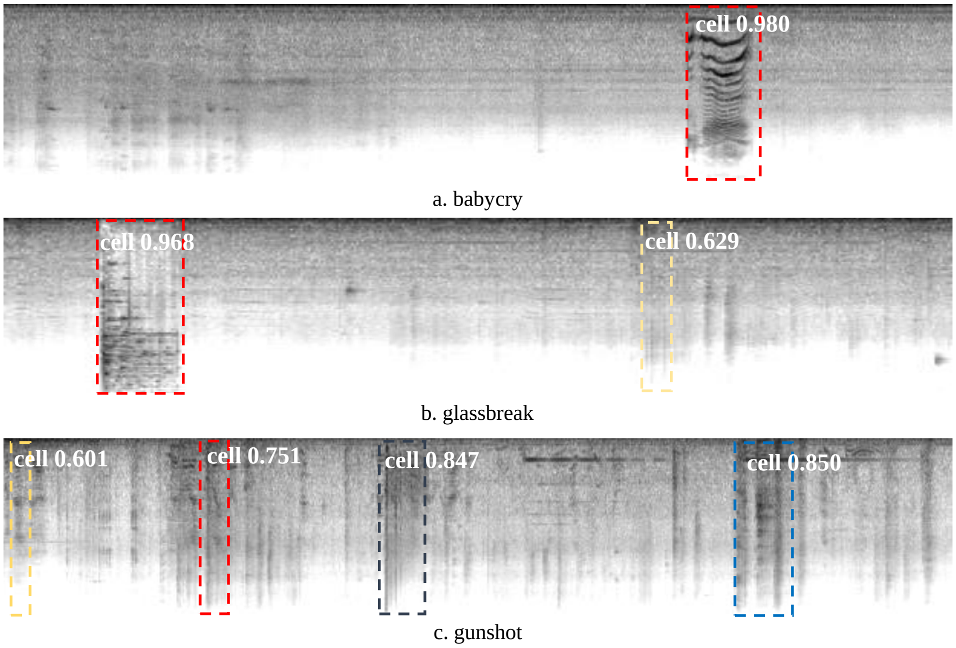
We also train a frame-level classifier for each event type. To prepare the inputs for training these classifiers, we only consider frames of the event and frames in their vicinity. In training step, we select a segment of audio with 200 frames where the specific event happens at some point inside it. We randomly choose the start of these segments so the event might happen at the start, middle or end of it. In the test step, we convert the proposed event-regions by Mask R-CNN to segments with the fixed length of 200 frames, so if the proposed region is larger than 200 frames, it will lose some of its frames before being analyzed by the frame-level classifier and if the proposed region is smaller than 200 frames, we would add some neighbor frames to it. We choose 200 frames as the segment size, because it covers 4.6 seconds of each audio and based on our calculations, this is enough to cover almost all of the events and gives us a balanced data set of audio segments with almost equal numbers of overall positive and negative frames. To label each frame in the segment, binary values are used based on whether that frame belongs to the event or not. As a result, for a training set of 5000 audios that is generated using the DCASE synthesizer with event being present in all of them, we create a new data set of 1000 audio segments of 200 frames where each segment has a label vector of 200 binary values.

The frame-level classifier analyzes each frame in the segment and assigns a probability value of belonging to the specific event to each frame. So the output of the classifier is a vector of 200 probability values.

We have conducted a grid search to determine the best filter size, number of filters, batch size, learning rate and dropout rate for the frame-level classifier using the development set. Our model is trained using ADAM optimizer with a batch size of 150 and learning rate of 0.001 for 100 epochs. The rest of the hyperparameters are shown in Fig 3.4. We use early stopping with minimum improvement of 0.0001 on validation accuracy and patience of 10 epochs to prevent over-fitting the model on the training set.

In the test stage, log-mel values for the audio are fed to Mask R-CNN to detect the regions potentially containing an event. We extract a segment of 200 frames from each of these proposed regions ,using the method that was explained above, and feed them to the frame-level classifiers. In the post-processing step, we calculate the average of probability values generated by the frame-level classifiers and use formula 1 to detect the true event out of the proposed ones. We conducted experiments with different values for λ to study the effect of the frame-level analysis on the final results as shown in Table 1. When $\lambda = 0$, only the confidence values from the Mask R-CNN is used and when $lambda = 1$, we are only using the probability values from the classifier to detect the true events out of the proposed regions by the Mask R-CNN. It is obvious from the table that different values of λ does not affect our model's performance on 'babycry'. This is due to more distinctive shape of this event in the mel-spectrograms that allows Mask R-CNN to successfully detect them with high confidence. Fig 3.5.a, shows a 'babycry' event-region detected by Mask R-CNN, that is clearly distinctive from its background noise.

Frame-level analysis plays more important role on the detection of true 'glassbreak' and 'gunshot' events. Because of the shorter and less identifiable shapes of these events, Mask R-CNN proposes more regions as the potential event. Parts b and c of Fig 3.5, shows examples of the generated bounding boxes by Mask R-CNN for these events along with their confidence values. Our results show that event-detection in audio by only employing object-detection models from the computer vision can not yield the best results for the shorter, less distinctive events and a finer analysis is required. As shown in the table, by increasing the weight of the frame-level classifier from 0 to 0.75, the F-score on the 'glassbreak' event increases significantly from 76.6% to 88.2%. This increase in λ value has a non-trivial impact on our model's performance over the 'gunshot' event too, where its F-score increases about 11.7%. We choose 0.75 as the λ value since it leads to the best performance on the development set with average F-score of 84.7% and ER of 0.30 over all three events. Overall, Our model shows outstanding results in detecting ‘babycry’ and ‘glassbreak’ events, which is due to the detectable patterns of these events in the mel-spectrograms. The ‘gunshot’ events can be very short and there are some background sounds in the audio that look very similar to the patterns of the ‘gunshot’ event in the mel-spectrograms. Our model has its lowest performance on this event compared to the other two events. It achieves an F-score of 73.5% on the development set for the ‘gunshot’ event. This comparatively lower performance is mostly due the fact that Mask R-CNN is an object detection model developed to detect objects with clear shapes or patterns in the images, and its performance drops in detecting boundaries for the ‘gunshot’ events with non-regular shapes in their log-mel spectrograms. Our method shows much better performance with events like ‘babycry’ and ‘glassbreak’ which have more regular and detectable shapes in the mel-spectrograms.



a. babycry

b. glassbreak

c. gunshot

Figure 3.5 Regions detected by Mask R-CNN for a.babycry, b.glassbreak and c.gunshot events. The red regions are the actual events. The value assigned to each region, is the confidence output for that region by Mask R-CNN model.

In Table 2, we compare the average F-scores and ERs achieved by our model with four non-ensemble sound event detection methods that had the highest performance at DCASE 2017 challenge. The evaluation is done over the DCASE 2017 evaluation data set. DNN/CNN model as reported in [117] has the best performance among all non-ensemble methods in DCASE 2017 and was ranked at third place in the challenge (The top two methods [85, 16], use sophisticated ensemble methods). The DNN/CNN model uses frame-level analysis to detect the onset of events. In Table 2, we cited the F-scores and ERs calculated by the challenge's official website for this method, which is slightly less than what authors reported in their paper. Our method achieves a slightly higher average F-score of 85.9% compared to 85.3% of DNN/CNN. Also we improved the F-score by 4.5% and ER by 0.09 for 'babycry' events. The improvement for 'glassbreak' events is 1.4% in F-score and 0.04 in ER. The only event that our method shows worse performance compared to DNN/CNN is 'gunshot' for which they use a network based on fully connected layers to detect this event.

We also compare our model against SLR-NMF method which was ranked 4th in the challenge and uses non-negative matrix factorization to filter out the noise and detect the event [180]. Although, this method has better performance on 'babycry' with F-score of 91.4%, its overall average F-score across all three events is 84.2% which is 1.7% less that AudioMask's overall average F-score. Our model also outperforms R-FCN [169], a region-based method that is inspired by an object detection model. Our AudioMask algorithm improved their average F-score by 3.9% and the average ER by 0.04 over the evaluation set. Our algorithm shows relatively similar results over 'glassbreak' events but increased the F-scores by 2.9% and 9.2% on 'babycry' and 'gunshot' events respectively. Furthermore, our model in average has 13.3% higher F-score than the baseline model and performs significantly better in detecting all of the three events. Overall, AudioMask performs better than all of the challenge's non-ensemble methods on the evaluation set of DCASE 2017.

Table 3.3 AudioMask's performance on two extra test sets generated using DCASE 2017 Synthesizer

	babycry		glassbreak		gunshot		Average	
	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER
test-set-1	95.6	0.09	93.7	0.13	88.7	0.22	92.7	0.15
test-set-2	93.5	0.13	96.4	0.07	87.1	0.26	92.3	0.15

To ensure the strength of our model, we used DCASE 2017's data synthesizer to create two more data sets using random seeds. This synthesizer can generate different data sets by changing the seed values. The values of 32 and 57 were randomly chosen as the seed values to create test-set-1 and test-set-2. We follow the same settings as the evaluation set of the DCASE 2017 challenge with the probability of event presence of 0.5 and 500 generated audios for each event. Table 3 shows the performance of AudioMask on these data sets. We achieve an average F-score of 92.7% and 92.3% on these sets using the exact same models as we used on the evaluation set. On these sets, AudioMask shows a better performance on 'babycry' and 'glassbreak' compared to 'gunshot' which confirms our assumption that our model has better performance on detecting events that have more distinguishable shape on the mel-spectrogram.

3.7 Chapter Summary

We proposed AudioMask, a novel sound event detection algorithm based on the Mask R-CNN framework and frame-level audio analysis. Our algorithm exploits the power of Mask R-CNN to create bounding boxes around candidate events, which are then fed as input to a frame-level classifier for finer analysis to determine if the candidate regions can be categorized as the target event or not. By comparing to the top non-ensemble algorithms of the DCASE 2017 task 2, we show that our method achieves higher performance. We believe our AudioMask algorithm can be generalized to detect other audio events of interest in the environment, especially those that are relatively longer and have a specific shape in the mel-spectrogram representation.

Chapter 4

SoundCLR: Contrastive Learning of Representations For Improved Environmental Sound Classification [1]

[1] Alireza Nasiri, and Jianjun Hu. Submitted to IEEE/ACM Transactions on Audio Speech and Language Processing, 03/04/2021.

4.1 Introduction and Related Work

Sound classification has a wide variety of applications in robot navigation [81], surveillance systems [127], alert systems [72], damage monitoring of materials [108], wildlife monitoring [24], [14], and designing autonomous cars [32], [41]. Environmental sounds have much higher variety than speech and this diversity and their noise-like characteristics makes ESC much more challenging than speech recognition. However, in recent years, sound recognition has seen great progress, which is partially due to the availability of large-scaled labeled datasets such as Environmental Sound Classification (ESC-50, ESC-10) [122], and Urbansound8k(US8K) [140]. The other reason is due to the shift from the traditional machine learning methods to deep learning models in the sound analysis tasks [64], [107]. Signal processing and machine learning methods such as matrix factorization [99], [11], dictionary learning [139], Support Vector Machines (SVM) [24], [162], Gaussian Mixture Model (GMM) [31], [98], and K-Nearest Neighbor (KNN) [122], [147], are widely adopted in ESC. However, currently deep learning models mostly consisting of convolutional layers, are achieving the highest accuracies in ESC benchmark studies [54], [158], [2].

More recently, there has been studies that utilize transfer learning to improve ESC with significant improvements [54], [112]. These studies employ the deep fully convolutional models, which are, surprisingly, pre-trained on ImageNet, and fine-tune them using the spectrogram representation of the audio samples. Since spectrograms of the audio events show some kind of local correlation similar to the images, models pre-trained on the large image datasets, perform well in the audio classification too.

In most supervised learning approaches for audio classification, the training is guided by the cross-entropy loss [54], [112], [121], [179]. The cross-entropy loss measures the difference between the output probability distribution and 1-hot encoded label and the model tries to reduce this loss by mapping the samples of the same class to their respective label vector.

Since a cross-entropy loss does not explicitly push away the samples of different classes from each other, it can suffer from poor margins between the representations of the samples from different classes [37]. This problem can negatively affect the generalization power of the model. We argue that we can strengthen the training signal by explicitly disentangling the samples of different classes from each other in the representation space. This can be done by introducing another term into the loss function which tries to pull the samples of the same class together while pushing them away from the samples of other classes in the representation space. The supervised contrastive loss [76] can be a suitable candidate for this.

The contrastive loss function which recently made significant progress in self-supervised learning [110], [59], [67], is based on defining positive and negative pairs by which it aims to pull together the samples in the positive pair (different augmented views of the same sample), while pushing away the samples in the negative pair (augmented views of different samples in the dataset). In self-supervised learning, the goal is to create a strong representation for the samples without using the ground-truth labels. By forcing the model to distinguish the different views of the same sample from the views of other samples, the model learns to focus on the most discriminative features of the samples.

Khosla et al. [76] has introduced a modified version of the contrastive loss for image classification, where the actual labels are employed to generate the positive and negative pairs. In the so-called supervised contrastive loss function, the samples of the same class are being pulled together in the representation space, while being pushed away from the representations of the samples from the other classes. In [76], they propose to train a model using supervised contrastive loss as a feature extractor in the first stage, and then freeze the weights of the model in the second stage and train a shallow classifier on top of it using cross-entropy loss. This model achieves higher accuracy than the same neural network model trained using the cross-entropy loss for image classification. Inspired by this study, here we aim to examine the effect of the supervised contrastive loss for ESC. In this study, We propose a new framework for ESC by introducing the supervised contrastive loss as the complementary to the cross-entropy loss for ESC. By calculating the training signals using both loss functions simultaneously, we allow the network parameters of the feature extraction model to be adjusted by a stronger signal with a single-stage training process.

In this chapter, we propose SoundCLR, a new framework for ESC which is based on strong data augmentation, supervised contrastive loss on the representation space along with the cross-entropy loss on the final output of the model. We show that by adding the cross-entropy loss as a factor to the supervised contrastive loss, we can further separate audio event classes from each other and reach higher accuracy than the exact same models trained by either cross-entropy or supervised contrastive loss alone.

Due to the small sizes of environmental sound datasets [122], [140], applying data augmentation techniques is crucial in achieving high accuracy in ESC. Here in addition to applying the common strong data augmentation methods, we introduce a simple but effective masking based data augmentation technique for ESC. Extensive experiments over three common environmental sound benchmark datasets show that our new framework achieves the state-of-the-art performance when compared to existing independent and ensemble models. Also, we show that increasing the number of channels by just triplicating the input mel-spectrograms can improve the accuracy of the classifiers trained on ImageNet.

Our major contributions in this research are as follows:

1. We propose a new framework for environmental sound classification based on a hybrid loss function consisting of a supervised contrastive loss and a cross-entropy loss. The supervised contrastive loss is applied on the representation space to disentangle the samples of the classes from each other, and the cross-

entropy loss is applied on the output of the classifier to map the representation vectors of the samples to their respective ground-truth labels.

2. We build a strong data augmentation pipeline, modifying the input data in both the wave signals and the mel-spectrograms. We show that triplicating input mel-spectrograms when using models pre-trained on ImageNet and random-masking of the mel-spectrograms can significantly improve the results.

3. We show that the deep neural network models trained with our proposed hybrid loss outperform the models with the same architecture, but trained by using only either the cross-entropy loss or the supervised contrastive loss. Also, we show that our model achieves state-of-the-art results on ESC-50, ESC-10 and US8K datasets.

4.1.1 Previous Studies in ESC

Environmental sound classification (ESC) deals with identifying some of the everyday audio events with varying lengths in a given audio signal. Even though a wide range of machine learning methods are proposed for sound classification [122], [31], [98], the deep learning methods have proven to achieve the best results in this field in recent years.

One of the earliest deep learning models for ESC was introduced by Piczak [121]. He used a 2-D structure, derived from the log-mel features of the audio signal as input to a model with two convolutional layers and two fully connected layers. This small model achieved 64.5% accuracy on ESC-50, and 81.0% on ESC-10, which is an increase of 20.6% for ESC-50 and 7.8% for ESC-10 compared to other traditional machine learning approaches such as random forest[122]. In the following studies, the researchers developed deeper convolutional models which achieve even higher accuracies in ESC [182], [157]. In [157], a model made of a combination of 1-D convolutional layers and fully connected layers extracts the features from the raw waveforms and achieves 71.0% accuracy on ESC-50. Zhu et. al [182] proposed a model with six convolutional layers to extract features from both the raw waves and the spectrograms. They gain accuracy values of 93.75% and 79.1% on ESC-10 and ESC-50, respectively.

Training deep neural networks with millions of parameters needs large amount of data. Current ESC benchmark datasets are considered to be comparatively small in the deep learning paradigm. Applying data augmentation techniques partially addresses this challenge [138], [178], [158]. Salamon et. al extensively studied the effects of data augmentation techniques like time-stretching, pitch-shifting and adding background noise on improving the performance of their proposed CNN model [138]. Zhang et. al proposed using mixup on audio signals to train a model of stacked convolutional and pooling layers [178]. Similar to mixup, between-class learning proposed in [158], mixes signals of the samples from different classes according to a random ratio, and a deep model of convolutional layers is trained to output the mixing ratio. By outputting the mixing ratio, the model learns the most important features in the sound signal. They achieve accuracy of 84.9%, 91.4%, and 78.3% over three datasets ESC-50, ESC-10, and US8K, respectively.

It is clear that deeper convolutional models and data augmentation can both improve the ESC classification performance. However the sizes of the common environmental datasets put a constraint on training models with around fifteen convolutional layers. To train deeper models that can potentially achieve better results, transfer learning with models pre-trained on ImageNet has proven to be successful for ESC[54], [112], [112]. Since spectrograms, which are commonly used to train audio classifiers, show image-like characteristics such as close correspondence between the local points, using the pre-trained models on ImageNet contributes to a better feature-extraction and in turn to a higher classification accuracy. More recently, ESResNet-Attention [54] uses a ResNet-50 model pre-trained on ImageNet with parallel attention blocks for both frequency and time domains. It achieves accuracies of 97%, 91.5%, and 85.4% on ESC-10, ESC-50 and US8K. Also, Palanisamy et. al [112], studied the performance of different well-known pre-trained models on ImageNet in audio classification. They report the validation accuracies of 90.65% and 84.76% on ESC-50 and US8K data sets when they use a single ResNet model. However, their best results comes from an ensemble of 5 DenseNet models where they achieve 92.89% and 87.42% accuracies on ESC-50 and US8K. They set a new state-of-the-art performance on these datasets.

The rest of this chapter is organized as: In section 4.2, we discuss contrastive learning in more details. Then we present the different modules of our method in details. We start by specifying our data augmentation techniques in 4.3. Then we discuss the characteristics of the three methods that we use to train the classifiers for environmental sounds in sections 4.4 and 4.5. Experimental setup and their results are shown in section 4.6. Finally we conclude the chapter in section 4.7.

4.2 Loss Functions in ESC

All of the classifiers in the previous studies in ESC, use the cross-entropy loss to train their deep neural network models [54], [158], [112], [178]. Cross-entropy loss was introduced to train neural networks with probability outputs [135], [8], and now is one of the most commonly used loss functions in classification tasks. This loss function measures the entropy between the actual probability distribution of the samples and the output probability distribution of the model. It can be calculated as:

$$H(p,q) = -\sum_{x} p(x) log(q(x)) \tag{4.1}$$

where p is the true probability distribution and q is the calculated one. By training the model to minimize the cross-entropy loss, the samples in the same class are mapped to the near-by points in the embedding space and the intra-class distances are minimized.

One of the downsides of the cross-entropy loss is the poor margins between the samples of different classes [37], which reduces the generality of the model trained by this loss function. The poor margins, or low inter-class distances, are related to the fact that there is no term in current loss function to push samples of different classes away from each other in the representation space.

There are a few loss functions proposed to improve the discrimination power of classifiers. Soft nearest neighbor loss [137], [45], and triplet loss [68] leverage the euklidean distances among the representation vectors of the samples to separate different classes from each other in the representation space. In both of them, minimizing the loss will force the model to output values which would minimize the intra-class to inter-class distance ratios. The main difference is that in the soft nearest neighbor loss, these distances are calculated among all of the samples present in the batch, where the triplet loss only considers 3 samples at a time: anchor, positive, and negative, where anchor and positive have the same label and the negative sample has a different label.

Another related work is contrastive learning [55], which has recently seen a lot of progress in the self-supervised learning field [110], [59], [67], [83]. Their main idea is the contrastive loss, which is based on building groups of samples as positives and negatives. It aims to decrease the distances among the representations of the samples in the positive group while increasing the distance among the representations of the samples in the negative group. It trains the model by pulling the representations of the samples from the positive pair together and pushing the representations of the samples in negative pairs, away from each other. In self-supervised learning [59],[83], [23], this loss is applied to pull together embeddings of two augmented views of a single data sample at a time, while pushing them away from the augmented views of the other samples. In the supervised version of the contrastive loss proposed by [76], instead of augmented views of the same sample, the positive pairs are built using

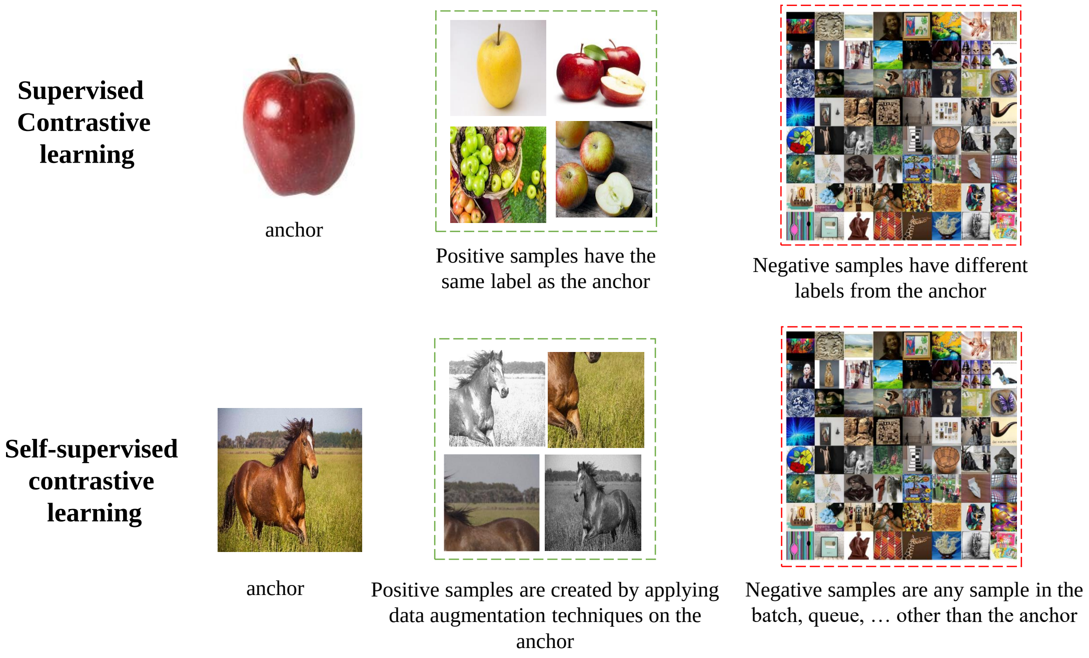

Supervised Contrastive learning
anchor
Positive samples have the same label as the anchor
Negative samples have different labels from the anchor
Self-supervised contrastive learning
anchor
Positive samples are created by applying data augmentation techniques on the anchor
Negative samples are any sample in the batch, queue, … other than the anchor

Figure 4.1 In the self-supervised contrastive learning, data augmentation methods are used to create the positive and negative samples, where in the the supervised contrastive learning, we use the labels of the samples to create the positive and negative samples.

samples with the same labels. This is very similar to what the triplet loss calculates. The major distinction is that the triplet loss only uses one negative and one positive pair while in supervised contrastive learning, we can have more than two samples in the positive or the negative group, which improves the training gradients by building a stronger contrast among the multiple samples and then may achieve higher accuracy. Figure 4.1 shows the difference between the self-supervised and supervised contrastive learning.

The proposed supervised contrastive learning [76] consists of two stages. In the first stage, the supervised contrastive loss is applied to disentangle the samples of the different classes from each other and in the second stage, the model's parameters are frozen and a classifier is trained on top of the model using the cross-entropy loss. Here, the supervised contrastive loss is introduced as an alternative to the cross-entropy loss. However, it seems that these two loss functions can be applied together as they are complementary to each other, where the representation space is simultaneously divided between the samples of the different classes using the contrastive loss, and the representation vectors are mapped to the ground-truth labels using the cross-entropy loss. In this study, we investigate whether the contrastive loss can train better models for environmental audio classification and propose a simple framework, in which both contrastive and cross-entropy loss functions are applied together to train a classifier, which achieves higher accuracy than the models trained with either one of these loss functions.

4.3 Data Augmentation

In the small datasets like ESC-50 and ESC-10, data augmentation plays a big role in improving the performance of deep neural network models. Data augmentation is a collection of deformations applied to the signal that creates a slightly different signal while preserving its ground-truth label. This technique exposes the model to a wider variety of samples of a class and as a result, improves the generalization power of the model.

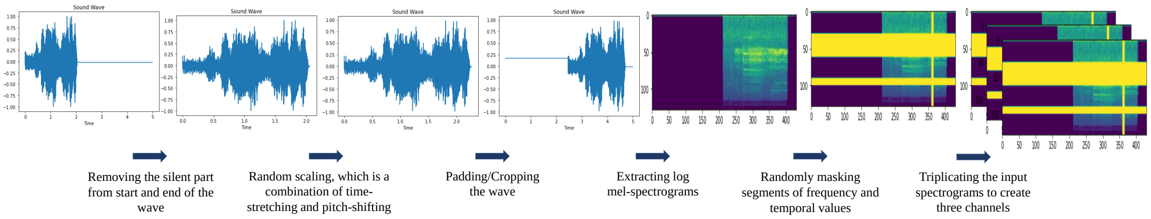

Sound Wave
Time
Sound Wave
Time
Sound Wave
Time
Sound Wave
Time
Removing the silent part from start and end of the wave
Random scaling, which is a combination of time-stretching and pitch-shifting
Padding/Cropping the wave
Extracting log mel-spectrograms
Randomly masking segments of frequency and temporal values
Triplicating the input spectrograms to create three channels

Figure 4.2 Data augmentation pipeline: The first three steps of removing the silent parts from the start and end of the signals, random scaling, and random padding/cropping is applied on the wave signals. Then we extract the log-scaled mel-spectrograms and randomly mask segments of the spectrograms across both time and frequency axes. At the end, we triplicate the masked spectrogram before inputting to the pre-trained model on the ImageNet. The number of the masked segments and their width is determined through our experiments.

We design several augmentation techniques to be applied on the audio signals consecutively on both the wave and the spectrogram forms of the audio signals. Because of the randomness of these techniques, the same signal can be augmented differently at each training iteration. Figure 4.2 shows the order in which the data augmentation processes are applied on the samples. The techniques that we use on the wave form of the audio signals are:

1. Removing Silent Beginning and End: We remove the silent parts from the start and the end of the signals. Some of the environmental sounds have a silent part in at least one end, which does not carry useful information. Removing the silent part allows more variety in the augmented signal, since the non-silent signal can be moved across the time axis more freely.

2. Random Scaling: This is the same technique used in [158], [54]. A random scaling factor is sampled uniformly from $[1.25^{-1}, 1.25]$, and is used for linear interpolation of the wave. This is equivalent to the combined pitch-shifting and time-stretching. We use this method because it only uses the waves and therefore is computationally much cheaper and faster than the time-stretching or pitch-shifting, where two short-time Fourier transforms are needed for each one of them.

3. Random Padding / Random Crop: Removing the silent part and scaling the signal, changes its length. To have signals with uniform lengths, we use the random padding or cropping depending on the length of the modified signal.

After applying these augmentations on the wave form of the audio, we extract the log-scaled mel-spectrograms from the waves. The log-scaled mel-spectrograms are two dimensional features which imitate the human hearing perception and are widely used in audio analysis. We apply another set of augmentations on these spectrograms including:

1. Frequency Masking: We randomly select f segments of the spectrograms in the frequency axis where width of each segment comes uniformly from $[0, F]$. The frequency values in these segments are set to 0.

2. Time Masking: We randomly select t segments of the spectrograms in the time axis where width of each segment is drawn uniformly from $[0, T]$. The frequency values in these segments are set to 0.

By masking segments of the spectrograms along the time and frequency axes, we are forcing the model to focus more on the temporal-frequency patterns in order to classify them, rather than a specific frequency or temporal value. This is similar to how dropout acts in making the model to not rely on any specific neuron [149].

4.4 Environmental Sound Classification with Cross-Entropy (Baseline)

We use the network model $f(.)$ to create the representations for the augmented audio samples. We choose ResNet-50 [57] as the representation model in all of our experiments. The output of the final average-pooling layer with 2048 dimensions in the ResNet model is normalized and we refer to it as the representation vectors for the audio samples. The classifier model is composed of a fully connected layer with c hidden units, where c specifies the number of the classes in the dataset. This classifier model uses the representation vectors as input, and outputs the probability of the samples belonging to each one of the c classes. One-hot encoded vectors and cross-entropy loss are used to train the representation model and the classifier in an end-to-end manner. The loss is calculated as:

$$\mathcal{L}_i^{cross-entropy} = -\sum_i y_i^{1-hot} log(y_i^{logits}) \tag{4.2}$$

, where y_i^{1-hot} is the 1-hot encoded ground-truth label for the sample i and y_i^{logits} is the output of the model for that sample.

4.5 Environmental Sound Classification with Contrastive and Hybrid Loss

Contrastive loss is based on differentiation between positive and negative sample pairs. We want to maximize the similarity between the samples in the positive pairs and minimize the similarity between the samples in the negative pairs. We build the positive and negative pairs based on the labels of the samples, where samples with the same label in the mini-batch create the positives. Since all of the samples with the same label are grouped together, all of the samples in the mini-batch which are not in the same group with a given sample, are considered to be negatives in regard to that sample. Figure 4.3 shows an overview of our three implemented models, where the first model is trained using the cross-entropy loss, the second model is trained using the supervised contrastive loss and the cross-entropy loss in two stages, and finally our proposed hybrid models are trained with both cross-entropy and supervised contrastive loss simultaneously within one stage.

Our supervised contrastive learning model consists of two networks: the representation network, and the projection network. The representation network $f(.)$ maps the sample x_i to the representation vector h_i, where $h_i = f(x_i) \in \mathbb{R}^{D_f}$. Applying the contrastive loss on the representation vectors h_i, because of the high dimensionality of the representation space, does not separate the samples of the different classes from each other. Reducing the dimension of the representation space causes loss of information by shrinking the size of the representation vectors. To prevent shrinking the representation vectors and also to separate the samples of different classes from each other, we use an additional network, which we call the projection network. The projection network $g(.)$ performs dimension-reduction by linearly mapping the representation vectors to the projection space with lower dimension, $z(h_i) = g(h_i) \in \mathbb{R}^{D_g}$, where $D_g << D_f$. Our experiments show that applying the contrastive loss on the projection space, improves the performance of the classifier significantly.

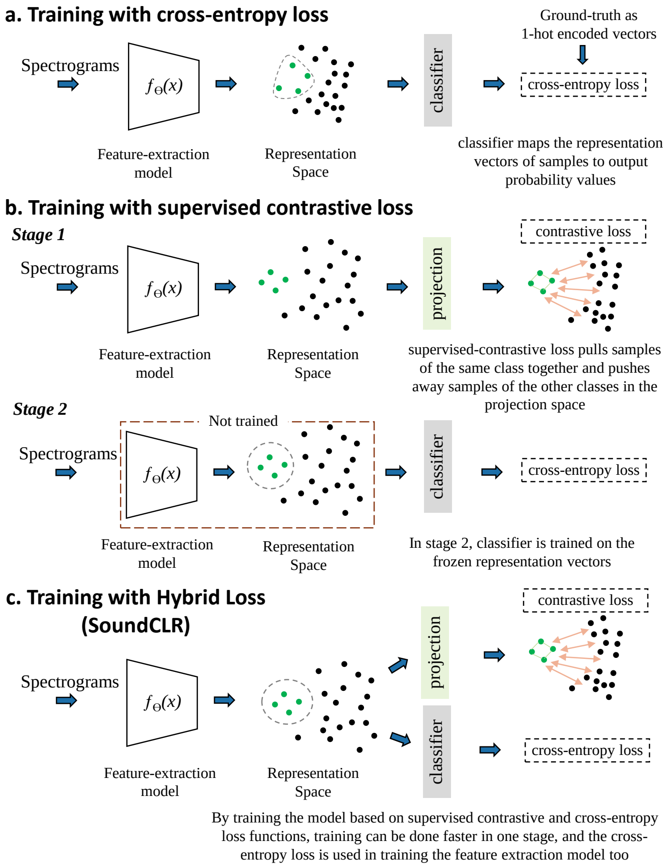

a. Training with cross-entropy loss
Spectrograms
$f_\Theta(x)$
Feature-extraction model
Representation Space
classifier
Ground-truth as 1-hot encoded vectors
cross-entropy loss
classifier maps the representation vectors of samples to output probability values
b. Training with supervised contrastive loss
Stage 1
Spectrograms
$f_\Theta(x)$
Feature-extraction model
Representation Space
projection
contrastive loss
supervised-contrastive loss pulls samples of the same class together and pushes away samples of the other classes in the projection space
Stage 2
Not trained
Spectrograms
$f_\Theta(x)$
Feature-extraction model
Representation Space
classifier
cross-entropy loss
In stage 2, classifier is trained on the frozen representation vectors
c. Training with Hybrid Loss (SoundCLR)
Spectrograms
$f_\Theta(x)$
Feature-extraction model
Representation Space
projection
contrastive loss
classifier
cross-entropy loss
By training the model based on supervised contrastive and cross-entropy loss functions, training can be done faster in one stage, and the cross-entropy loss is used in training the feature extraction model too

Figure 4.3 Overview of our three training schemes: a. model is trained with cross-entropy loss which measures the difference between the output probabilities and the 1-hot encoded ground-truth labels, b. training a classifier with supervised contrastive loss within two stages. In the first stage, the supervised contrastive loss disentangles the samples of different classes from each other in the representation space, and in the second stage, the feature-extraction model is frozen and a classifier layer is trained on top of the representation vectors. Here, the projection layer is only used for dimension reduction in stage 1, since applying the contrastive loss directly on the high-dimensional representation space is not as effective in separating the samples, c. training scheme using our proposed hybrid loss (SoundCLR), which utilizes both cross-entropy and supervised contrastive loss functions to disentangle the samples and map them to their corresponding labels, simultaneously.

We run our experiments by using ResNet-50 as the representation network. Our projection model, which is a linear fully connected layer, uses the normalized representation vectors to output the projection vectors. The supervised contrastive loss [76], is applied on the normalized projection vectors of the audio signals. Given labels $Y = \{y_1, y_2, ..., y_N\}$ corresponding to samples $X = \{x_1, x_2, ..., x_N\}$, the loss is calculated as:

$$\mathcal{L}_i^{sup-cont} = -\frac{1}{N_{y_i}} log \frac{\sum_{j=1}^{N} \mathbb{1}_{[y_i=y_j]} exp(z_i.z_j/\tau)}{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} exp(z_i.z_k/\tau)} \tag{4.3}$$

where N_{y_i} is the number of samples in the mini-batch with the same label as y_i, $\mathbb{1}_{[y_i=y_j]} \in \{0, 1\}$ evaluates to 1 iff $y_i = y_j$, and $u.v$ measures the cosine similarity between the vectors u and v which is calculated as $u.v = u^T v/||u||||v||$. The N refers to the batch size and τ is the temperature, which controls the effect of the hard negatives in the training process [168].

After training the representation and the projection networks with the contrastive loss, we freeze f and train a classifier on the representation vectors, using the cross-entropy loss. In this way, the contrastive and the cross-entropy loss functions are being used to train the model in two separate stages, where the loss signal from the cross-entropy loss does not modify the parameters in the representation model. We believe these two losses can be applied as complementary to each other in one single training stage. In our hybrid approach, we propose to use the supervised contrastive loss to disentangle the representations of the samples of a class from the representations of the samples of the other classes, and to use the cross-entropy loss in mapping the logits of the samples of each class to the one-hot encoded vector. We believe that by combining these two loss signals, we can provide a stronger training signal from the representation model, which can lead to a better classification accuracy. Assuming that the cross-entropy loss identifies the similarities between the samples with the same label by mapping them to the same 1-hot encoded vector, and the contrastive loss tries to minimize the intra-class to inter-class distance ratio, combining these two loss signals provides the model with a much more informative signal for the classification task. In this hybrid method, we calculate the loss as:

$$\mathcal{L}_i = \alpha \mathcal{L}_i^{sup-cont} + (1 - \alpha) \mathcal{L}_i^{cross-entropy} \tag{4.4}$$

where α is a hyper-parameter. We show that using this combined loss, results in a faster and better convergence. To run the experiments using this loss function, we add two parallel branches to the output of the embedding network. As shown in Figure 4.3c, the logits from the last layer of the ResNet-50 are normalized to create the representation vectors. We apply two models in parallel to these vectors. One is the classifier which is a linear fully connected layer which has as many hidden units as the number of the classes and is used to calculate the cross-entropy loss. The other is a projection model with the same architecture as the classifier. The projection layer downsamples the representation vector and normalizes the output in order to calculate the supervised contrastive loss in a more efficient way within a lower dimension space.

4.6 Experiments and Discussion

In this section, we describe the settings for our experiments, evaluate our proposed methods, and compare the results with the other methods in ESC.

4.6.1 Datasets

We train and evaluate our models using three publicly available datasets, which are commonly used in environmental sound classification studies. These datasets are ESC-50, ESC-10 [122], and UrbanSound8K [140].

The ESC-50 dataset includes 2000 audio samples (each is 5 seconds long), which are divided equally into 50 classes in 5 folds. Each fold is balanced regarding the number of the samples from different classes. This means that there are 8 samples for each class inside every fold. The classes can be divided into five major categories: animals, natural soundscapes and water sounds, non-speech human sounds, interior/domestic sounds, and exterior/urban sounds. The sampling rate of each audio signal is 44.1 kHz.

The ESC-10 is a subset of ESC-50 with 400 samples in 10 classes. Like ESC-50, the ESC-10 is also divided into five folds. The classes in ESC-10 are dog bark, rain, sea waves, baby cry, clock tick, person sneeze, helicopter, chainsaw, rooster, and fire crackling. The length of each one of the sound samples is 5 seconds.

The UrbanSound8K or in short US8K, has 8732 samples from 10 different classes. The length of the audio clips are different and the maximum length is 4 seconds. The classes are air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The sound segments in US8K, have different sampling rates and can be mono or stereo. For the stereo audio segments, we calculate an average of the channels before augmenting the samples.

We use the official split of these three datasets in our experiments.

4.6.2 Preprocessing and Data Augmentation

We normalize all of the wave signals into values in the range of [-1, 1]. Then in training, we make the signals go through the data augmentation pipeline described in section 4.3. For ESC-10 and ESC-50, we perform the random padding and random crop with the output size of 22500 to include 5 seconds of data. For US8K, this output size is 176400 to have signals with length of 4 seconds which is the maximum length for the audio clips in this dataset.

To calculate the mel spectrograms, we use hamming window with a size of 1024 with 50% overlap and 128 mel-bands. We get the logarithm of the mel-spectrograms to make our features more compatible with the humans hearing sense.

We are augmenting the data in two stages: wave, and spectrogram. We explained the techniques used in each stage in 4.3. To show the contribution of the different modules in our augmentations pipeline, we run experiments on ESC-50 using our baseline model and show the results in Figure 4.4. Our first experiment is run by randomly initializing ResNet-50, without using any of the data augmentation techniques, and feeding the mel-spectrograms in one channel to the model which achieves an average validation accuracy of 74.6 % across all 5 folds of ESC-50. Triplicating number of channels at this point (not using transfer learning) does not improve the results. By not using the transfer learning, increasing the number of channels just increases the size of the input without providing any extra information to the model and the randomly initialized model cannot utilize this extra redundant information. Initializing the weights of the model with the parameters learnt by training on the ImageNet, increases the average accuracy from 74.6% to 84.6%. This significant rise in the accuracy is due to the correlation among the nearby spectrotemporal values in the spectrograms which is similar to the correlation among the nearby pixels in an image. Since the parameters of the ResNet-50 model that we use as our feature extractor, is trained on images with three channels, to utilize all of the parameters of the model, we present our input in three channels. By simply copying the log-scaled mel-spectrogram values to increase the number of the channels to three. As shown in Figure 4.4, this simple technique increases the average accuracy value by more than 4%. Each set of the augmentations on the wave signals and mel-spectrograms, as outlined in section 4.3 and shown in Figure 4.4, further improve the average accuracy by 1.6% and 1.45%, respectively.

Table 4.1 Accuracy of the baseline model on ESC-50 based on the different maximum widths and number of the masked segments

	F = 16, T = 32				F = 32, T = 32				F = 32, T = 64			
	t = 0	t = 1	t = 2	t = 3	t = 0	t = 1	t = 2	t = 3	t = 0	t = 1	t = 2	t = 3
f = 0	90.2	90.35	90.4	90.45	90.2	90.35	90.4	90.45	90.2	90.75	90.6	90.35
f = 1	91.2	91.6	91.35	91.1	91.45	91.7	91.5	91.05	91.6	91.85	91.5	91.25
f = 2	91.05	91.35	91.55	91.4	91.6	**92.1**	91.8	91.4	91.85	91.65	91.3	91.6
f = 3	90.8	91.2	91.3	91.55	91.65	91.7	91.45	90.95	91.5	91.4	91.8	91.05

To identify the best parameters for the number of the masked segments (f and t) and their maximum-width (F and T), we run experiments using values of $\{16, 32\}$ for F and $\{32, 64\}$ for T, where F and T are the maximum width of each frequency

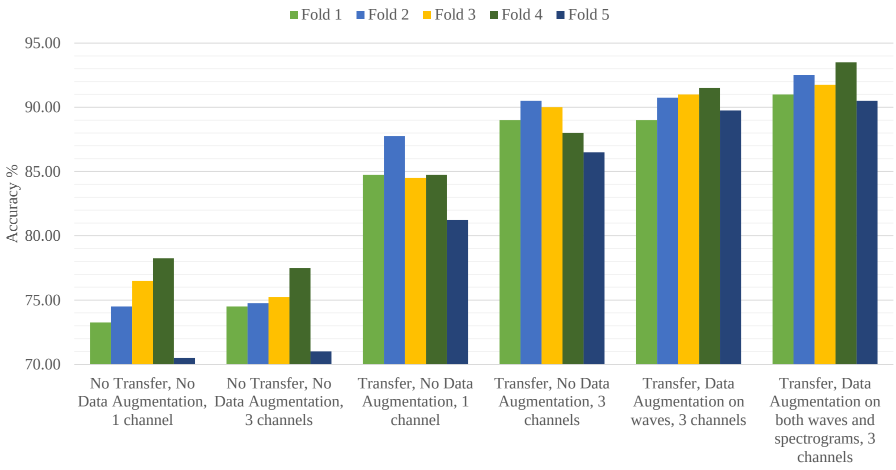


Figure 4.4 Results on application of transfer learning, data augmentation and number of input channels, by using the baseline model on ESC-50. Transfer learning has significant effect on improving the results. Increasing the number of the input channels from 1 to 3, improves the accuracies only when transfer learning is used.

and temporal masked segments. To find the best number for the masked segments across the frequency and time axes, we perform a grid-search for $f, t \in \{0, 1, 2, 3\}$, where f is number of the masked segments in frequency values and t is the number of masked temporal segments. Table 4.1 shows the accuracy values from running the baseline model on ESC-50. It turns out that the best accuracy for the baseline model happens when $F = 32, T = 32$ and $f = 2, t = 1$. Using these settings for masking, improves the accuracy of the baseline model on ESC-50 from 90.2% to 92.1%. We believe that this improvement is caused by forcing the model not to be dependent on any specific spectrotemporal values. Our results show that the frequency masking is comparatively more effective than the temporal masking. This might be in part due to the higher importance of the information in the frequency bands compared to most of the short temporal segments, which forces the model to identify more global patterns in classifying the audio events. Increasing the number of the masked segments beyond two frequency and one temporal segments, eliminates some of the essential information and reduces the overall accuracy of the model.

4.6.3 Training Settings

We use ResNet-50 as the representation model in our experiments. The outputs of the final pooling layer with 2048 dimension is normalized to generate the representation vector for each sample. In the baseline model we add a linear fully connected layer with h_c hidden units where h_c is the number of the classes in the dataset. The $h_c = 10$ for ESC-10 and US8K and it is 50 for ESC-50. We train the baseline model using the cross-entropy loss.

In training with supervised contrastive loss, we use a linear projection model with one fully connected layer with linear activation. Based on our experiments using a linear layer with 64 hidden-units, achieves the best accuracy as it makes the representation space small-enough to be able to efficiently separate the samples from each other, and yet large enough to preserve the most important features. For the supervised contrastive loss, we use the implementation from [76]. We use the samples in the mini-batches to draw the positives and negatives for each sample in the batch. It has been shown both in self-supervised and supervised contrastive loss [23], [76], that having larger number of negatives can help the training process. For this reason, we use batches of 128 samples to be able to include more samples in the positive and negative groups and create a more informative contrast. In the case of ESC-50, for each sample, we have more than 2 samples from the same class (positive samples), and more than 125 samples from other classes (negative samples) in average.

We used our baseline model to perform a grid-search to identify the optimal hyperparameters for the training stage. We use Adam optimizer with a base learning rate of $5e-4$ for ESC-50 and ESC-10, and $1e-4$ for US8K. We also applied an exponential learning rate decay with a factor of 0.98 and 10 warm-up epochs. To ensure a fair comparison among our models, we use the same hyperparameters in training the baseline, models with supervised contrastive loss, and the hybrid loss.

4.6.4 Results of Our Three Models

We run experiments to identify the optimal value for α in Formula 4.4, from the possible values of $\{0.01, 0.25, 0.5, 0.75, 0.99\}$. When $\alpha = 0.01$, most of the hybrid loss value comes from the cross-entropy loss, and when $\alpha = 0.99$, the supervised contrastive loss is the dominant factor. As shown in Table 4.2, the highest accuracy is achieved when $\alpha = 0.5$ for all three datasets. This shows that by equal contribution from the cross-entropy and contrastive loss functions, we can achieve higher accuracy and the model can achieve higher accuracy when these two loss functions act as complementary to each other.

Table 4.2 Results of different alpha values on the hybrid loss. The best results are achieved when the cross-entropy and contrastive loss contribute equally to calculating the loss signal.

	ESC-50 (%)	ESC-10 (%)	US8K (%)
$\alpha = 0.01$	91.85	99.25	85.03
$\alpha = 0.25$	92.25	99.5	85.46
$\alpha = 0.5$	**93.2**	**99.75**	**85.96**
$\alpha = 0.75$	92.8	99.25	84.94
$\alpha = 0.99$	92.7	99.25	84.72

We also compare the average of five runs and the best results of our three models trained with cross-entropy, supervised contrastive, and hybrid loss with each other in Table 4.3. On average, ResNet-50 model trained with our proposed hybrid loss achieves validation accuracies of 92.86%, 99.6%, and 85.78% on ESC-50, ESC-10, and US8K. This shows that not only by combining the two stages of training in the supervised contrastive learning we can make the training faster, but also the hybrid loss actually increases the performance of the model by providing a stronger loss signal. In Figure 4.5, we show the average performance of our models on each fold in the experimented datasets. This figure shows that the model trained with the hybrid loss either outperforms or have the same accuracy across all of the folds in ESC-50 and ESC-10, and most of the folds in US8K.

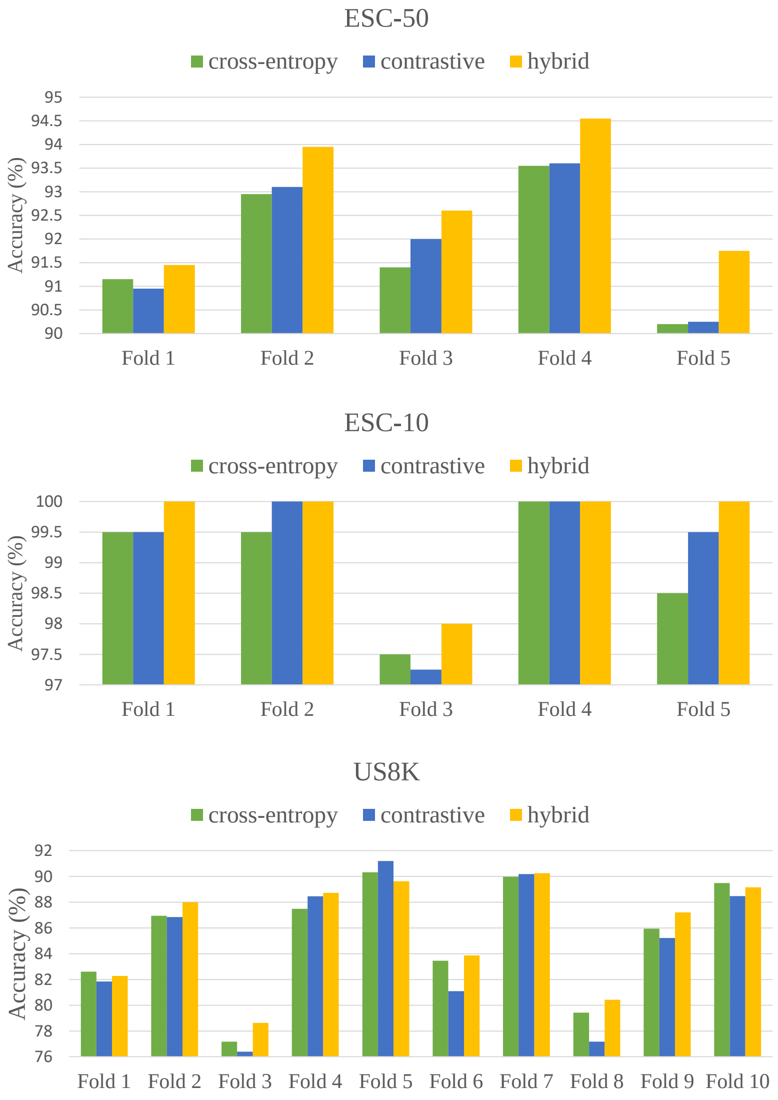


Figure 4.5 The validation accuracies over different folds of the datasets averaged over 5 times training. In most of the folds, the models trained with the hybrid loss reach higher accuracies that the ones trained with the cross-entropy and contrastive loss functions.

In Table 4.4, we compare our hybrid approach, which is based on training a single ResNet-50 model with the hybrid loss, with other methods in ESC. Since in [112], an ensemble of ResNet and DenseNet models are used in ESC, to have a fair comparison,

Table 4.3 Results of Our Three Methods. The average values shows the average validation accuracies of our models over 5 times training.

		ESC-50 (%)	ESC-10 (%)	US8K (%)
ResNet-50, Cross-Entropy Loss	Average	91.85 ± 0.64	99.0 ± 0.47	85.28 ± 0.69
	Best	92.65	99.5	86.16
ResNet-50, Supervised Contrastive Loss	Average	91.98 ± 0.68	99.25 ± 0.59	84.69 ± 0.98
	Best	92.85	99.75	86.09
ResNet-50, Hybrid Loss	Average	92.86 ± 0.46	99.6 ± 0.14	85.78 ± 0.51
	Best	93.4	99.75	86.49

we calculate the performance of an ensemble of five independently trained ResNet-50 models based on our proposed hybrid loss. Here, we use the average of the softmax output of these five models to get the accuracy value for the ensemble model.

As shown in Table 4.4, a single ResNet-50 trained with the hybrid loss, increases the accuracies on ESC-50, ESC-10, and US8K by 1.88%, 2.75% and 1.18% respectively compared to the previous state-of-the-art results from DenseNet in [112]. The improvement is more significant, if we compare our results with the ResNet-50 model trained in [112]. We believe that our data augmentation techniques and use of the contrastive loss has a large role in achieving these state-of-the-art results. Ensembling our five ResNet-50 models increases the accuracy over ESC-50 and US8K datasets to 93.6% and 88.01%.

Table 4.4 Comparing the results of our proposed training method with the hybrid loss and other state-of-the-art methods. In order to have a fair comparison, we have separated the results achieved by the single models from the ensemble ones.

Model	Architecture	Features	ESC-50 (%)	ESC-10 (%)	US8K (%)
Human [122]	-	-	81.3	95.7	-
Piczak-CNN [121]	2 conv + 2 FC layers	Mel-Spectrogram	64.5	90.2	73.7
EnvNet-v2 [158]	10 conv + 3 FC + 5 max-pooling	Raw Wave	84.7	91.3	78.3
ESResNet [54]	ResNet-50	Log Spectorgram	90.8	96.75	84.9
ESResNet-Attention [54]	ResNet-50 + Attention	Log Spectrogram	91.5	97.0	85.42
ResNet [112]	ResNet-50	Log Mel-Spectrogram	90.93	-	85.03
DenseNet [112]	DenseNet-201	Log Mel-Spectrogram	91.52	-	85.31
SoundCLR	ResNet-50	Log Mel-Spectrogram	**93.4**	**99.75**	**86.49**
ResNet-Ensemble [112]	Ensemble of 5 ResNet-50	Log Mel-Spectrogram	92.64	-	87.35
DenseNet-Ensemble [112]	Ensemble of 5 DenseNet-201	Log Mel-Spectrogram	92.89	-	87.42
SoundCLR-Ensemble	Ensemble of 5 ResNet-50	Log Mel-Spectrogram	**93.6**	**99.75**	**88.01**

4.6.5 Robustness to Noise

The environmental sound classifiers need to be robust to the noise, since environmental sound recordings usually include several types of background noise. We use white noise signal with normal distribution to study the effect of the noise in our classifiers. These noise signals are created randomly with mean of 0 and different standard deviation values of $1e-4$, $5e-4$, and $1e-3$. We add the noise signals to the audios in the test set.

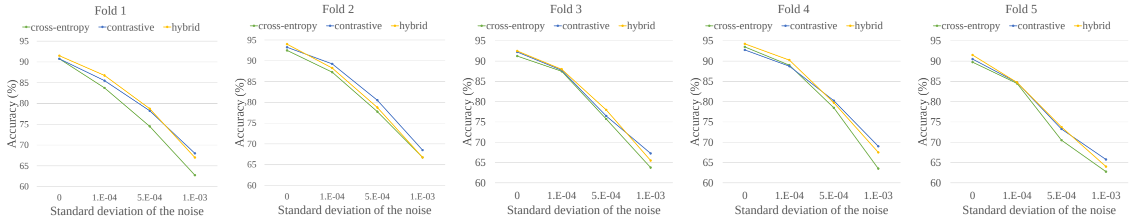


Figure 4.6 Effect of white noise on the accuracy values of the classifiers trained with cross-entropy, supervised contrastive, and hybrid loss on ESC-50. The models trained with the supervised contrastive loss show more resilience towards the noise and the accuracies of the models trained with this loss decreases slower than the models trained with the other two losses.

Figure 4.6, shows the effect of the noise on the performance of our classifiers trained on ESC-50 dataset. This figure shows that the models trained with the supervised contrastive loss are more resistant against the noise and the accuracies of these model decreases slower than the other two models. We think this is due the higher distance between the samples in the representation space of the models trained with the contrastive loss. The contrastive loss is based on increasing the distance among the samples of the different classes and this increased margin contributes to the better performance on the noisy signals.

4.7 Chapter Summary

In this chapter, we introduced SoundCLR, a novel framework based on strong data augmentation and contrastive learning to classify environmental audio events. We introduced the supervised contrastive learning concept into ESC, and showed that our model can achieve state-of-the-art results in ESC by using the contrastive loss to disentangle the samples of the different classes from each other in the representation space, combined with the cross-entropy loss used to map the representation vectors to the output labels. In our proposed method, we demonstrated how the contrastive and cross-entropy loss functions can be used as complementary to each other in audio classification to provide a stronger error signal in training the audio classifiers. Furthermore, we showed how simple data augmentation techniques such as masking and increasing the number of the channels in the input mel-spectrograms can significantly increase the classification accuracy of the model.

Chapter 5

Online Damage Monitoring of SiC$_F$-SiC$_M$ Composite Materials using Acoustic Emission and Deep Learning [1]

[1] Alireza Nasiri, Jingjing Bao, Donald Mccleeary, Steph-Yves M Louis, Xinyu Huang, Jianjun Hu. 2019. IEEE Access. 7: 140534-140541. Reprinted here with permission of publisher.

5.1 Introduction and Related Work

SiC$_f$ - SiC$_m$ composite tubes have been proposed to replace Zircaloy cladding for nuclear fuel [63] with the purpose of improving the accident tolerance. These tubes have demonstrated excellent high temperature stability, good fracture toughness, better oxidation resistance, lower hydrogen production rate and very low thermal and irradiation creep [151]. Degradation detection is essential to allow intervention before failure. One of the widely-used non-destructive degradation monitoring methods for SiCf-SiCm composite is acoustic emission (AE) [133, 4] . In this technique, acoustic signals produced during the material degradation process are recorded by a piezoelectric sensor and analyzed.

Numerous studies have been conducted to analyze the correlation between AE parameters and the progress of damage in materials. Forio et. al [44] studied the state of the damage in the SiC composites by analyzing their microstructures. In their research, AE was used as a complementary tool to confirm the correlation of the progression of damage and the cumulative number of the acoustic events. Nozawa et. al [109] studied the failure behavior of SiC composites by analyzing the cumulative AE energy. Morscher et. al [104] showed the relation between AE and stress vs. strain curve of a SiC fiber reinforced SiC matrix composite at elevated temperatures. More recently, Alva [5] used normalized cumulative AE energy and time intervals between the acoustic events to calculate the damage variable and re-built the stress vs. strain figure for the SiC composites. All of these studies have two things in common. First, they only use a small number of hand-picked features from AE events to investigate the damage behavior of the material. Second, they analyze AE to identify the location or type of damage in SiC composites as a post-mortem analysis technique, not quite an online tool to monitor the degradation process.

Of interest to this research is monitoring the current stage of the material as it undergoes the degradation process. There are two critical points in the stress vs.

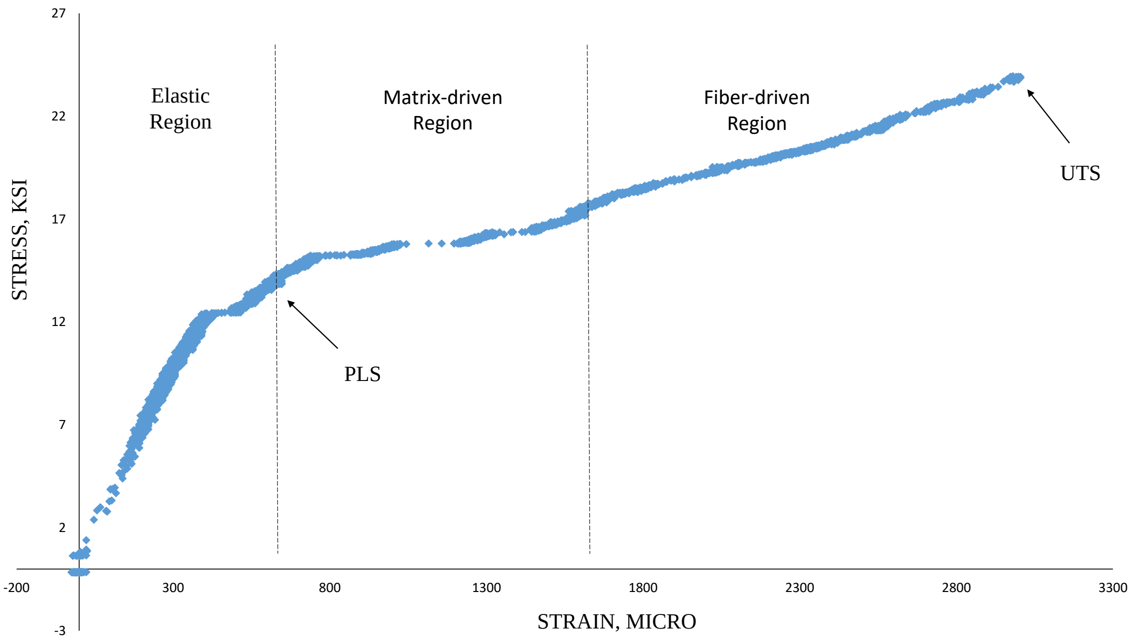


Figure 5.1 A stress vs. strain curve divided into three different stages using PLS and UTS points

strain figure of the composite called Proportional Limit Stress (PLS) and the Ultimate Tensile Stress (UTS). We use these points to define three stages in the SiC composites as it undergoes internal pressurization. The stage before PLS is called elastic region which is dominated with very few small, low-energy cracking events. Around PLS, a large number of AE events with high energy occur. After PLS, the stress vs. strain curve bends due to significant amount of damage in the material. Immediately following the PLS is a matrix-driven damage regime in which the AE events are believed to be derived mostly from matrix cracking. As matrix cracks saturate, load is being transferred to the fibers. At some point between PLS and UTS, the material degradation is believed to shift towards a mainly fiber break dominated regime called the fiber-driven region. Figure 5.1 shows these three regions on a representative stress vs. strain curve. By using these regions, we define the online monitoring of the SiC composites degradation using the AE analysis as a classification problem in which each AE event belongs to one of the three regions. So given an AE event, if we can map it to one of these three stages, we can identify that stage as the current degradation state of the material. Deep learning algorithms have been widely applied to degradation detection problems like predicting the corrosion rate of steel [136], monitoring cracks in civil structures [20, 87, 153], and identifying defects in carbon fiber reinforced polymers [93]. These algorithms are capable of building models based on complex relations of numerous parameters and performing prediction in real time. Deep learning methods are end-to-end learning paradigms, which means that unlike traditional machine learning algorithms feature extraction and classification is done consecutively in them. Sadowski et. al [136] trained a multilayer perceptron model to predict the corrosion current rate of steel in concrete using corrosion current density. Lin et. al [87] used a Convolutional Neural Network (CNN), which is a type of deep neural networks first proposed by LeCun et. al [80], to extract high-level features from the sensor signals of large-scale structures and identify the damage locations. Another CNN model including two convolutional layers was proposed by Meng et. al [93] to identify the defects in carbon fiber reinforced polymer by classifying the ultrasonic signals. In all of these works, deep learning methods have shown higher accuracy compared to models that only rely on hand-picked features.

In this study, we propose two supervised learning methods of Random Forest (RF) and CNN, to analyze the AE of SiC composites and classify them into one of the stages in the material's degradation process. RF models have already been used to study corrosion by analyzing AE [103], but it has never been used to analyze AE of composites to detect the state of their degradation. We train RF models using all of the relevant hand-picked features of the AE data to exploit the capacity of these features. We also built a deep CNN model to extract high-level features from the AE and classify them. We use raw sound signals as input to the CNN model. This allows the model to extract the most descriptive features from the AE through the training process.

We also examined combining the features for a number of AE samples together and training our models with them. This was done because not all of the events that are happening at the different stages of the degradation are uniquely inclusive to that stage and there are events of the same type that happen across all three stages of the degradation process. Taking this into account, analyzing a single event at a time would likely provide poor results. To ensure the reliability of these models, we train the RF and DL models using the AE data from eight out of the nine experiments. The model's performance is measured on withheld data of the ninth experiment. By completely separating the training and testing sets, we ensure that our results are reliable and unbiased.

The contribution of this study includes the following:

1. We exploit expert-defined features of the AE data by training RF models to learn the patterns of AE from SiC composite tubes and use those patterns to monitor the state of the degradation in the SiC composite.

2. We train end-to-end deep CNN models to extract high-level features from the raw audio signals of AE from SiC composite tubes and utilize these models to monitor the degradation process in SiC composite tubes.

3. We test our models on the AE from an experiment not represented in the training set to ensure the reliability of our results.

The rest of this chapter is organized as: In section 5.2, we go over the data acquisition process and the experimental setup used to gather the data. In sections 5.3, we describe the random forest method, which we use as our baseline. Section 5.4 describes our proposed convolutional neural network method to classify the acoustic signals. We present our results and analysis in section 5.5, and finally we conclude the chapter in section 5.6.

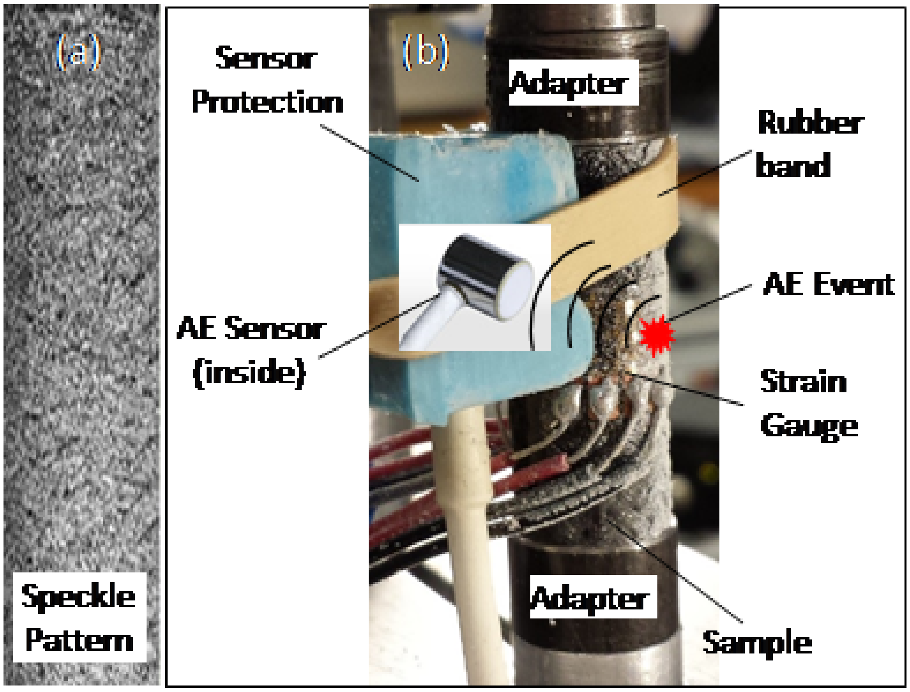


Figure 5.2 (a) Speckle pattern and (b) Assembly.

5.2 Data Acquisition

The AE equipment used in this work is a Micro-II Digital AE System (Physical Acoustics Corporation) equipped with a NANO-30 AE sensor (125 – 750 kHz range) and 20 dB preamplifier. Figure 5.2 shows how the AE sensor, tube and strain gauges were set up for the experimentation.

SiGA^{TM} composite samples were manufactured and provided by General Atomics in the form of SiC_f - SiC_m composite tubes made of nuclear grade fiber, pyrolytic carbon interface coating and CVI SiC matrix. Nine samples from three different batches were tested. The setup is the same as [3], where a bladder-based burst rig was used for an open-end burst test that uses hydraulic oil to exert a uniform pressure to the inner surface of the sample via a soft polymer tubing.

Before testing, the samples are prepared by painting a speckle pattern on the outer surface and attaching a rosette strain gauge. The rosette strain gauge measures the strains in in both the axial and hoop directions. The speckle pattern is used with the digital image correlation (DIC) technique to measure the strain distribution on the outer surface of the sample. We define three different regions of elastic, matrix-driven and fiber-driven behavior, and label each AE sample based on the region in which it occurs. We can identify different stages of the material degradation by classifying the AE events into one of the three different classes. In this classification problem, we want to output the current stage of the material, given the acoustic features of xi for a generated audio sample i. A model is trained to map AE signals to the stage in which they belong. We calculated a set of features such as amplitude, duration, rise-time, absolute energy, peak counts and signal strength for each audio signal recorded by the sensor. Here x_i for each audio sample is defined to be these set of features expressed as,

$$\begin{aligned} x_i = [&rise - time_i, rise - counts_i, energy_i, duration_i, amplitude_i, RMS_i, ASL_i, \\ &average - frequency_i, peak - counts_i, r - freq_i, l - freq_i, signal - strength_i, \\ &absolute - energy_i, frequency - c_i, p - frequency_i] \end{aligned} \tag{5.1}$$

We examine the capacity of these hand-picked features by using them to train a RF model. The input of the RF model is the x$_i$ features for each sample. Since the cracks that happen at different stages are not inclusive to that stage and for instance we might have some matrix cracks in all three stages, only depending on one generated audio for identifying the stage of the material's degradation is not a reliable method. For this reason, we have also combined a number of consecutive cracks and calculate the average, max and min of their feature vectors to train a RF model.

5.3 Random Forest (RF): The Baseline Algorithm

RF is an ensemble machine learning method that can be used for classification or regression [13]. RF classifier employs a number of decision trees to fit the data set and uses majority voting to combine the results from its decision trees. The algorithm begins by creating subsets of the training data where many of the samples are generated using bootstrap technique. Each tree will be fitted to one subset by applying binary partitioning at each node of the tree. One major constraint for partitioning tree is that the predictor variable can be selected from n random variables at each node. The trees are fully grown and each tree produces a label for each sample. The final predicted class of a sample is calculated by the majority vote of the predictions from all of the trees for that sample. Figure 5.3 shows how each one of the decision trees in the RF model outputs a label and the majority voting method is used to identify the output of the model.

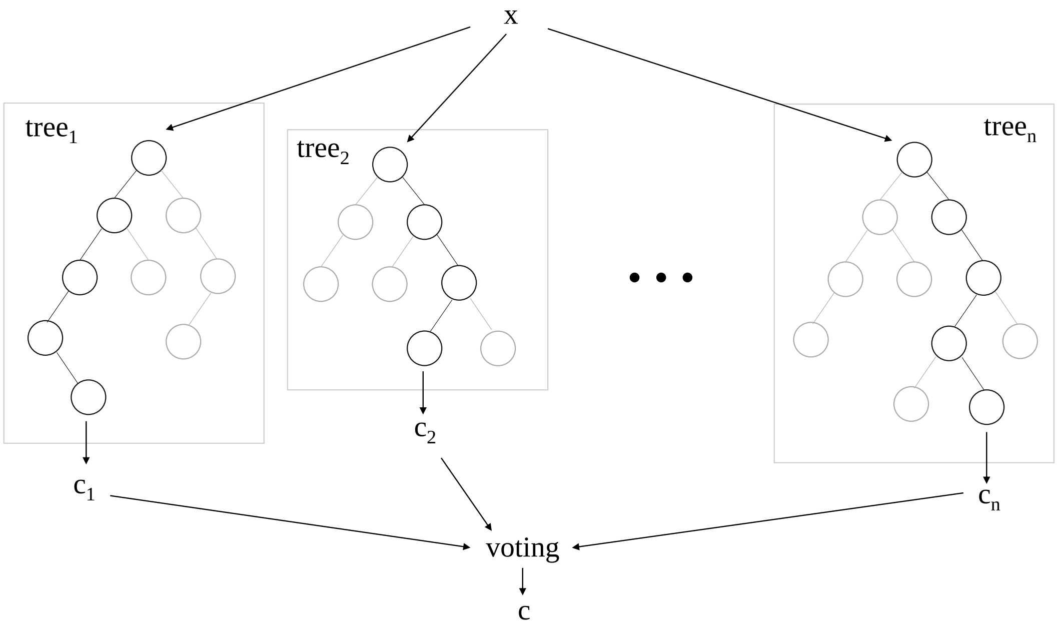


Figure 5.3 The random forests (RFs) are composed of a number of decision trees, where the output of the RF can be determined using the voting mechanism.

5.4 Deep Convolutional Neural Networks (CNN) in AE Classification

CNN models were first introduced to detect patterns in computer vision [80]. They have achieved numerous state-of-the-art results in the computer vision tasks during recent years [33, 27]. CNN has also been successfully applied to the sound tasks [121, 64]. A CNN model usually consists of a number of components that are stacked one after another in layers. These components are convolutional, pooling and classifier layers.

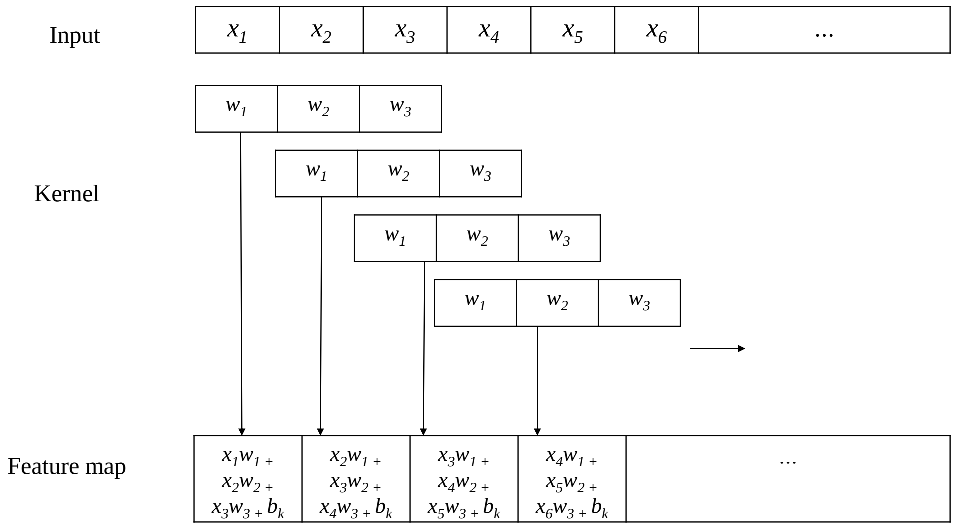

Input
x_1
x_2
x_3
x_4
x_5
x_6
...
Kernel
w_1
w_2
w_3
w_1
w_2
w_3
w_1
w_2
w_3
w_1
w_2
w_3
Feature map
$x_1w_1 + x_2w_2 + x_3w_3 + b_k$
$x_2w_1 + x_3w_2 + x_4w_3 + b_k$
$x_3w_1 + x_4w_2 + x_5w_3 + b_k$
$x_4w_1 + x_5w_2 + x_6w_3 + b_k$
...

Figure 5.4 One dimensional convolution.

In the **convolution layer**, a kernel convolves over the input to produce the feature map. Equation 5.2 shows the formula for the one-dimensional convolution operation.

$$(h_k)_i = (W_k * x)_i + b_k \tag{5.2}$$

Here $*$ represents convolution operation in which, h is the output of neuron i in the k^{th} feature map of the convolutional layer. W and b are trainable parameters which represent weights and bias of the one-dimensional kernel. Figure 5.4 shows how a one-dimensional kernel slides over the input vector to produce its feature map.

The **Pooling layer** is a nonlinear down-sampling layer that calculates either maximum or average values from each sub-region of the input data. The **Activation function** is usually applied to the output of the convolution layers as a non-linear function. A rectified linear function (ReLU) of *relu(x) = max(0,x)* is the most commonly used activation function. The **Softmax** activation function is usually used at the top of all the layers as a classifier. This layer outputs a vector of values in range of (0,1), where each value represents the probability of the input sample belonging to each one of the possible categories. The Softmax activation function in Equation 5.3 is defined by

$$softmax(z)_i = \frac{exp^{d_i}}{\sum_k exp^{d_k}} \tag{5.3}$$

where d_i and d_k are the input values and z is the output vector of the softmax.

The **Loss function** is used to calculate the true difference between the actual class labels and the probability values that the CNN model has calculated. This difference is back-propagated through the network to adjust the trainable parameters in the model. One **epoch** is when the entire training set is fed into the model and its error has been back-propagated. The overall forward calculations through the layers and back-propagation of the error are called the **training process** which can include multiple epochs. We continue training the model for multiple epochs, until it can map the input data to the correct categories with an acceptable accuracy. To ensure that the model learns the most general patterns from the data and it does not overfit the training data set, **dropout** is applied on the output of each convolution layer [149]. By using dropout, we randomly select some of the values in the feature map. Also, we use a validation set, which does not overlap with the training set and measures the progress of the learning in the model. Once the model's accuracy stops improving over the validation set for a number of epochs, we terminate the training process. This technique is called **early-stopping**.

We design a CNN model to classify AE event samples from the SiC composites into three categories. In our model, a combination of convolutional and pooling layers are used to extract high-level features from the audio signals and a softmax layer is employed to classify each audio to one of the three stages in the composite's degradation process. In each convolutional layer a number of kernels convolve over the input of that layer to produce feature maps. These feature maps themselves are the input to the next layer. We apply Rectified Linear Unit (relu) to the output of convolutional layers [106]. Average-pooling and max-pooling layers are used to down-sample the data.

For the DL model, we have done experiments using samples with one or multiple AE events as input. Figure 5.5 shows the structure of our model with the audio signals of 40 consecutive cracks as input.

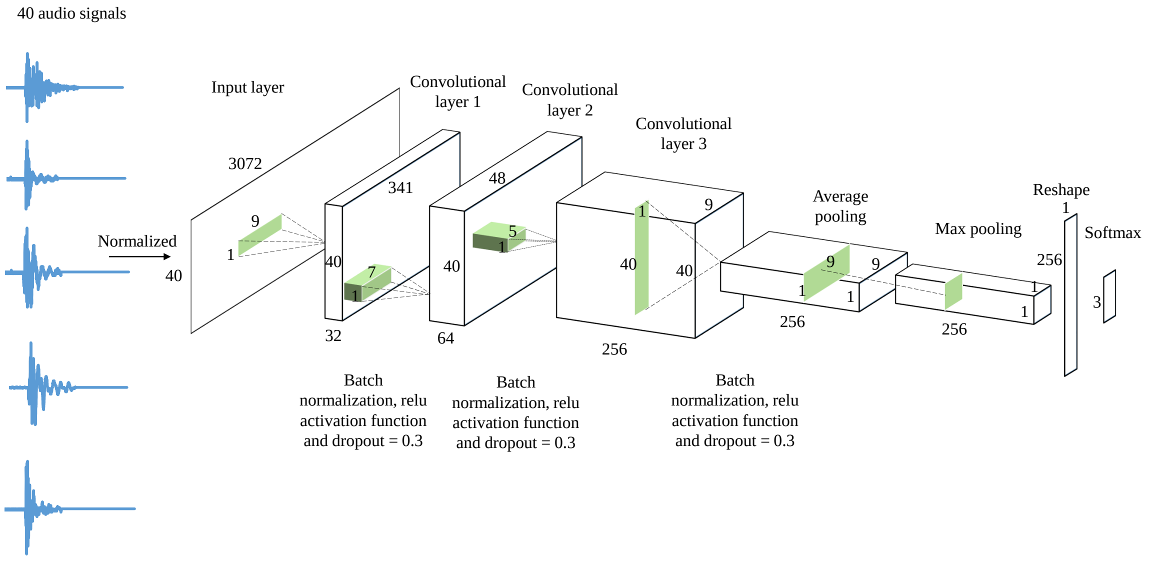


Figure 5.5 Architecture of our deep CNN model.

The first convolutional layers use 32 kernels of size 1x9, where the second one has 64 kernels of size 1x7 and the last convolutional layer used 256 kernels of size 1x5. Batch normalization, relu activation function and dropout with rate of 0.3 is applied to the output of each convolutional layer. The dropout layer helps to regularize the network and avoid overfitting [149]. The last layer uses Softmax activation function to output probabilities for each class.

We applied a binary cross-entropy function to calculate the loss of the calculated values during the training. This function is shown in Equation 5.4:

$$Loss = -\frac{1}{B}\sum_{i=1}^{B} y_i.log(y_i^{'}) + (1 - y_i).log(1 - y_i^{'}) \tag{5.4}$$

where y is the actual label and y' is the calculated label by the model.

We train the CNN models for at most 50 epochs and at the end of each epoch, the model is evaluated on a validation set. We apply early-stopping with min-delta of 0.1 and patience of 10 epochs.

5.5 Result and Discussion

Our dataset consists of the AE data from 9 individual tests that have been conducted over 9 different test samples from 3 batches of SiC$_f$ - SiC$_m$ composite tubes. Table5.1 shows the number of the AE events created in each one of these tests during each of the three stages of behavior.

Table 5.1 Number of generated audios in different stages of the material degradation

Stages	Test 1	Test 2	Test 3	Test 4	Test5	Test 6	Test 7	Test 8	Test 9
Elastic Region	1589	2133	2663	2073	1651	823	1074	2200	2566
Matrix-driven Region	1217	507	85	614	881	940	1353	147	256
Fiber-driven Region	409	1863	1062	1819	352	845	1111	1095	4948
Total	3215	4503	3810	4506	2884	2608	3538	3442	7770

On average, most of the events of each test are generated in the first stage of the degradation, before the PLS point. In test 9, an abnormally large number of events are recorded during the fiber-driven region. Also because of the quick transition from the matrix-driven region to the fiber-driven region in test 3, test 8 and test 9, the number of the events on the matrix-driven region for these tests are relatively low.

Each test, based on the characteristics of the composite and the place of the crack, has a different number of AE events in its three stages. To make sure that our models are learning the most general patterns from the AE data and our methods are reliable in monitoring SiC composites' degradation process, we train each one of our models on the data from the eight tests and evaluate them on the remaining ninth set. For example, this means that to evaluate our model on test 5, we have trained it using the data from all of the experiments other than test 5.

Table 5.2 shows the results from our models' evaluation on each one of the test sets. RF_1 denotes the RF model trained and tested with a vector of hand-picked features for each individual audio sample. RF_20 and RF_40 stands for the RF model trained with the average, max and min values for the hand-picked features of 20 and 40 consecutive audio samples, respectively. As a result, the size of the input vector for R_20 and RF_40 is triple the size of the input vector for RF_1. CNN_1, CNN_20 and CNN_40 refer to the deep CNN model that is trained and tested using 1, 20 and 40 raw audio signals as input. We have randomly shuffled the training and testing data sets five times to train and evaluate our RF models on each data set. Also each CNN model is trained and tested five times for each data set, where both training and testing sets were randomly shuffled each time. The percentage values on Table 5.2 are the average of the accuracy for each model.

As shown in Table 5.2, CNN models in general outperform the RF models. On average CNN_40 has the highest performance with 86.6% accuracy although the CNN_40 model takes longer to train compared to the other CNN models due to its

Table 5.2 Accuracy (%) of our models evaluated on different tests (accuracy for the RF models is the average of 5 runs and the accuracy of the CNN models is the average of 5 runs)

Models	Test 1	Test 2	Test3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Average
RF_1	71.6	74.3	70.5	80.3	81.2	64.8	72.8	83.9	66.9	74.0
RF_20	84.2	76.0	83.1	87.9	86.6	80.2	81.4	92.3	70.5	82.5
RF_40	85.3	74.4	82.9	87.9	87.7	79.3	82.2	93.7	71.7	82.8
CNN_1	85.2	**84.7**	91.1	89.7	84.5	77.6	81.2	95.0	**73.5**	84.4
CNN_20	**93.3**	82.2	**92.4**	**91.5**	94.9	76.2	80.4	**97.7**	69.3	86.4
CNN_40	91.6	81.8	91.2	**91.5**	**95.2**	**82.4**	**86.2**	97.4	62.2	**86.6**

input size. On the other hand, CNN_1 models train very quickly (less than a minute on average) and they reach an average accuracy of 84.4%. CNN_20 only takes 6-8 minutes to train on average and reaches a high accuracy of 86.4% which is just 0.2% less than CNN_40. Our results show that combining the features for a number of samples together and training our models with them improves the prediction accuracy. This improvement is more obvious in test 1 and test 5, where CNN_20 shows 11.1% and 10.4% improvement over CNN_1. This can be related to the possibility of having the instances of the same type of the cracks across all of the three stages which makes it difficult for the model to be able to judge the current state of the material based solely on one sound sample. However, combining larger numbers of samples might not lead to a better result, since it can smooth out the features and make them less differentiable for the model. It seems that this is the case for test 2 and test 9, where combining samples has lowered the accuracy of the RF and CNN models. Overall, it seems that CNN_20 is the most reliable choice for prediction, considering its high average accuracy, fast training process and smaller variance in its results.

In these models, the performance over different test data sets are similar with a few exceptions. Our models have their highest prediction accuracy over the AE of test 8 with the best value of 97.7% for the CNN_20 model. This model also has high accuracy of 93.3% and 94.9% for test 1 and test 5, respectively. The accuracy of our model for test 9 is comparatively lower than other test sets. The highest accuracy on this data set belongs to CNN_1 model with accuracy of 73.5%, which is significantly lower than the accuracy of the other data sets. Based on the investigation of our models, this is likely due to the abnormally large number of the events on the fiber-driven stage of test 9. Table 5.1 shows that the number of the events in fiber-driven stage varies from 352 to 1863 for tests 1-8, where test 9 has 4948 samples at this stage. Our models misclassify at least 2100 samples out of the 4948 samples in the fiber-driven region of test 9. This can be related to the fact that some of the events recorded in this stage are not damage related. The previous studies of the AE from SiC composite tubes [3] state that there are non-damage events recorded at the fiber-driven stage that cannot be differentiated from the damage-related events.

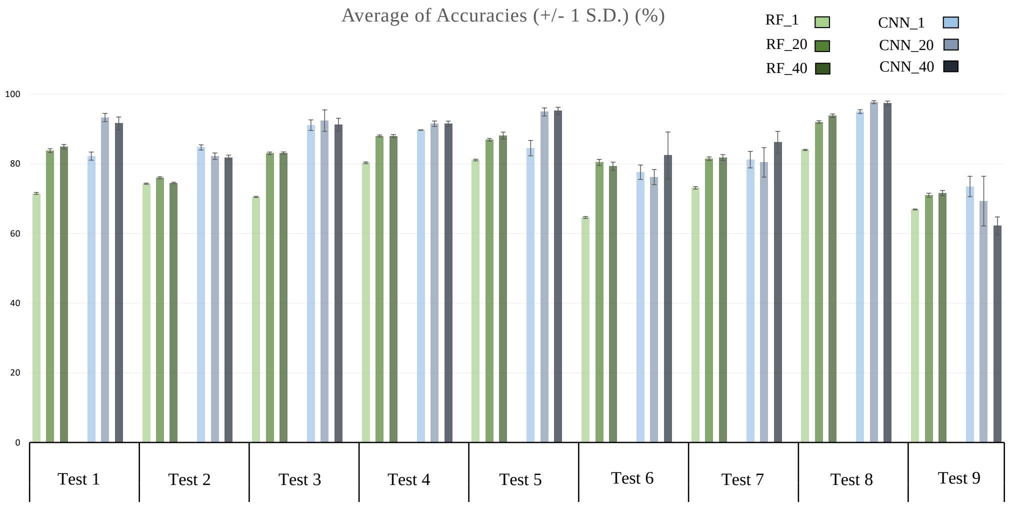


Figure 5.6 Average and Standard Deviation of accuracy of our proposed models over all of the tests

Figure 5.6 shows the average accuracy and the standard deviations of our models on tests 1-9. It is clear based on this Figure and Table 5.2, that the CNN models achieve higher accuracy than the RF models. This shows that using convolutional layers in extracting the high-level features from the AE is a better tool than using hand-picked features in AE classification of the SiC composites.

5.6 Chapter Summary

In this study, we analyzed the AE signals from nine different open-end burst tests of SiC$_f$ - SiC$_m$ composite tubes. We used RF and CNN to classify the AE signals of each test into one of three stages in its degradation process. These stages are defined based on the PLS and UTS points in the stress vs. strain curve of the composite and are called elastic region, matrix-driven region and fiber-driven region.

We conducted multiple experiments by training RF and CNN models using the features from one and multiple AE signals. We show that combining the features of a number of samples together can result in a higher accuracy in detecting the current stage of the material. To ensure the reliability of our results, we completely separated the training and testing sets by training on the AE data of eight experiments and testing on the ninth one. Furthermore, we run each model several times and average the results. Our results indicate that AE analysis using state-of-the-art machine learning methods is a reliable and efficient way to monitor the degradation process in the SiC$_f$ - SiC$_m$ composite tubes. Across our data sets, CNN models show a higher accuracy compared to the RF models. This proves that using deep convolutional layers to extract high-level features from the AE test data of the SiC composites is a more robust tool in representing the audio samples and can yield better classification results.

Based on our experiment results, it can be expected that RF and CNN approaches as proposed here can also be used to monitor the degradation process of other materials by analyzing their AE signals.

Chapter 6

Conclusion

6.1 Conclusion

In this dissertation proposal, we introduced our work on audio event detection, acoustic emission analysis, and environmental sound classification based on the contrastive learning. For the first work, we We proposed AudioMask, a novel sound event detection algorithm based on the Mask R-CNN framework and frame-level audio analysis. Our algorithm exploits the power of Mask R-CNN to create bounding boxes around candidate events, which are then fed as input to a frame-level classifier for finer analysis to determine if the candidate regions can be categorized as the target event or not. By comparing to the top non-ensemble algorithms of the DCASE 2017 task 2, we show that our method achieves higher performance. We believe our AudioMask algorithm can be generalized to detect other audio events of interest in the environment, especially those that are relatively longer and have a specific shape in the mel-spectrogram representation.

In the second work, we introduced SoundCLR, a novel framework based on strong data augmentation and contrastive learning to classify environmental audio events. We introduced the supervised contrastive learning concept into ESC, and showed that our model can achieve state-of-the-art results in ESC by using the contrastive loss to disentangle the samples of the different classes from each other in the representation space, combined with the cross-entropy loss used to map the representation vectors to the output labels. In our proposed method, we demonstrated how the contrastive and cross-entropy loss functions can be used as complementary to each other in audio classification to provide a stronger error signal in training the audio classifiers. Furthermore, we showed how simple data augmentation techniques such as masking and increasing the number of the channels in the input mel-spectrograms can significantly increase the classification accuracy of the model.

Finally, we analyzed the AE signals from nine different open-end burst tests of $\mathrm{S}i\mathrm{C}_f$ - $\mathrm{S}i\mathrm{C}_m$ composite tubes. We used RF and CNN to classify the AE signals of each test into one of three stages in its degradation process. These stages are defined based on the PLS and UTS points in the stress vs. strain curve of the composite and are called elastic region, matrix-driven region and fiber-driven region. We conducted multiple experiments by training RF and CNN models using the features from one and multiple AE signals. We show that combining the features of a number of samples together can result in a higher accuracy in detecting the current stage of the material. To ensure the reliability of our results, we completely separated the training and testing sets by training on the AE data of eight experiments and testing on the ninth one. Furthermore, we run each model several times and average the results. Our results indicate that AE analysis using state-of-the-art machine learning methods is a reliable and efficient way to monitor the degradation process in the $\mathrm{S}i\mathrm{C}_f$ - $\mathrm{S}i\mathrm{C}_m$ composite tubes. Across our data sets, CNN models show a higher accuracy compared to the RF models. This proves that using deep convolutional layers to extract high-level features from the AE test data of the SiC composites is a more robust tool in representing the audio samples and can yield better classification results. Based on our experiment results, it can be expected that RF and CNN approaches as proposed here can also be used to monitor the degradation process of other materials by analyzing their AE signals.

6.2 Future Work

6.2.1 Extending AudioMask to Identify Overlapping Audio Events

Overlapping audio events significantly modifies the audio signals. This change can make it very difficult for the AudioMask or other vision-based methods to identify the type of the overlapping events. For these samples, a stronger version of the frame-level analysis might be required to identify the audio events. As the next step, we plan to develop a method based on AudioMask, where the finer-level analysis is the major part of the method and can identify the portions of the audio that belong to different events.

6.2.2 Self-Supervised Contrastive Learning to Pre-Train Audio Classifiers

Pre-training the audio classifiers using self-supervised contrastive learning can significantly improve their performance specially on the small datasets. As larger non-speech audio datasets become available, a major focus of the research should be on utilizing these samples to pre-train the audio classifiers. Contrastive learning is a suitable candidate for this purpose.

Bibliography

[1] Ossama Abdel-Hamid et al. "Convolutional neural networks for speech recognition". In: *IEEE/ACM Transactions on audio, speech, and language processing* 22.10 (2014), pp. 1533–1545.

[2] Vinayak Abrol and Pulkit Sharma. "Learning Hierarchy Aware Embedding From Raw Audio for Acoustic Scene Classification". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 1964–1973.

[3] Rajeev Aggarwal et al. "Noise reduction of speech signal using wavelet transform with modified universal threshold". In: *International Journal of Computer Applications* 20.5 (2011), pp. 14–19.

[4] DG Aggelis et al. "Acoustic emission monitoring of degradation of cross ply laminates". In: *The Journal of the Acoustical Society of America* 127.6 (2010), EL246–EL251.

[5] Luis H Alva. "Monitoring The Progress Of Damage In A SICf-SICm Composite Nuclear Fuel Cladding Under Internal Pressure Using Acoustic Emission". In: (2018).

[6] Pradeep K Atrey, Namunu C Maddage, and Mohan S Kankanhalli. "Audio based event detection for multimedia surveillance". In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 5. IEEE. 2006, pp. V–V.

[7] Daniele Barchiesi et al. "Acoustic scene classification: Classifying environments from the sounds they produce". In: *IEEE Signal Processing Magazine* 32.3 (2015), pp. 16–34.

[8] Eric Baum and Frank Wilczek. "Supervised learning of probability distributions by neural networks". In: *Neural information processing systems*. 1987, pp. 52–61.

[9] Emmanouil Benetos, Mathieu Lagrange, and Simon Dixon. "Characterisation of acoustic scenes using a temporally constrained shit-invariant model". In: *DAFx*. 2012.

[10] Thierry Bertin-Mahieux et al. “The million song dataset”. In: (2011).

[11] Victor Bisot et al. “Acoustic scene classification with matrix factorization for unsupervised feature learning”. In: *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2016, pp. 6445–6449.

[12] Y-Lan Boureau, Jean Ponce, and Yann LeCun. “A theoretical analysis of feature pooling in visual recognition”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 111–118.

[13] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[14] Emre Cakir et al. “Convolutional recurrent neural networks for bird audio detection”. In: *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE. 2017, pp. 1744–1748.

[15] Emre Cakir et al. “Polyphonic sound event detection using multi label deep neural networks”. In: *2015 international joint conference on neural networks (IJCNN)*. IEEE. 2015, pp. 1–7.

[16] Emre Cakır and Tuomas Virtanen. “Convolutional recurrent neural networks for rare sound event detection”. In: *Deep Neural Networks for Sound Event Detection* 12 (2019).

[17] Abdullah Caliskan et al. “Diagnosis of the parkinson disease by using deep neural network classifier”. In: *Istanbul University-Journal of Electrical & Electronics Engineering* 17.2 (2017), pp. 3311–3318.

[18] Salvatore Casale et al. “Speech emotion classification using machine learning algorithms”. In: *2008 IEEE international conference on semantic computing*. IEEE. 2008, pp. 158–165.

[19] Michael A Casey and Alex Westner. “Separation of mixed audio sources by independent subspace analysis”. In: *ICMC*. 2000, pp. 154–161.

[20] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. “Deep learning-based crack damage detection using convolutional neural networks”. In: *Computer-Aided Civil and Infrastructure Engineering* 32.5 (2017), pp. 361–378.

[21] S Chandrakala and SL Jayalakshmi. “Environmental audio scene and sound event recognition for autonomous surveillance: A survey and comparative studies”. In: *ACM Computing Surveys (CSUR)* 52.3 (2019), pp. 1–34.

[22] Lei Chen, Sule Gunduz, and M Tamer Ozsu. “Mixed type audio classification with support vector machine”. In: *2006 IEEE International Conference on Multimedia and Expo*. IEEE. 2006, pp. 781–784.

[23] Ting Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *arXiv preprint arXiv:2002.05709* (2020).

[24] Selina Chu, Shrikanth Narayanan, and C-C Jay Kuo. “Environmental sound recognition with time–frequency audio features”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.6 (2009), pp. 1142–1158.

[25] Marco Crocco et al. “Audio surveillance: A systematic review”. In: *ACM Computing Surveys (CSUR)* 48.4 (2016), pp. 1–46.

[26] Jifeng Dai et al. “R-fcn: Object detection via region-based fully convolutional networks”. In: *Advances in neural information processing systems*. 2016, pp. 379–387.

[27] Wei Dai et al. “Very deep convolutional neural networks for raw waveforms”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 421–425.

[28] An Dang, Toan H Vu, and Jia-Ching Wang. “A survey of deep learning for polyphonic sound event detection”. In: *2017 International Conference on Orange Technologies (ICOT)*. IEEE. 2017, pp. 75–78.

[29] Gert Dekkers et al. “The SINS database for detection of daily activities in a home environment using an acoustic sensor network”. In: *Detection and Classification of Acoustic Scenes and Events 2017* (2017).

[30] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[31] P Dhanalakshmi, S Palanivel, and Vennila Ramalingam. “Classification of audio signals using AANN and GMM”. In: *Applied soft computing* 11.1 (2011), pp. 716–723.

[32] RA Dobre et al. “Low computational method for siren detection”. In: *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME)*. IEEE. 2015, pp. 291–295.

[33] Chao Dong et al. “Image super-resolution using deep convolutional networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307.

[34] Konstantinos Drossos, Sharath Adavanne, and Tuomas Virtanen. “Automated audio captioning with recurrent neural networks”. In: *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE. 2017, pp. 374–378.

[35] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. “Clotho: An audio captioning dataset”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 736–740.

[36] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. “Survey on speech emotion recognition: Features, classification schemes, and databases”. In: *Pattern recognition* 44.3 (2011), pp. 572–587.

[37] Gamaleldin Elsayed et al. “Large margin deep networks for classification”. In: *Advances in neural information processing systems*. 2018, pp. 842–852.

[38] Meng Joo Er et al. “Attention pooling-based convolutional neural network for sentence modelling”. In: *Information Sciences* 373 (2016), pp. 388–403.

[39] Gianpaolo Evangelista et al. *Sound source separation*. 2011.

[40] Pasquale Foggia et al. “Audio surveillance of roads: A system for detecting anomalous sounds”. In: *IEEE transactions on intelligent transportation systems* 17.1 (2015), pp. 279–288.

[41] Pasquale Foggia et al. “Reliable detection of audio events in highly noisy environments”. In: *Pattern Recognition Letters* 65 (2015), pp. 22–28.

[42] Eduardo Fonseca et al. “Freesound datasets: a platform for the creation of open audio datasets”. In: *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR). 2017.

[43] Eduardo Fonseca et al. “FSD50k: an open dataset of human-labeled sound events”. In: *arXiv preprint arXiv:2010.00475* (2020).

[44] Ph Forio and J Lamon. “Mechanical behavior of a 2D SiC/SiC composite with a multilayered matrix”. In: *Proc. 12th Int. Conf. Composite Mater.(ICCM12)*. 1999, pp. 5–9.

[45] Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. "Analyzing and Improving Representations with the Soft Nearest Neighbor Loss". In: *International Conference on Machine Learning*. 2019, pp. 2012–2020.

[46] Jort F Gemmeke et al. "An exemplar-based NMF approach to audio event detection". In: *2013 IEEE workshop on applications of signal processing to audio and acoustics*. IEEE. 2013, pp. 1–4.

[47] Jort 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)*. IEEE. 2017, pp. 776–780.

[48] Behnaz Ghoraani and Sridhar Krishnan. "Time–frequency matrix feature extraction and classification of environmental audio signals". In: *IEEE transactions on audio, speech, and language processing* 19.7 (2011), pp. 2197–2209.

[49] Dimitrios Giannoulis et al. "A database and challenge for acoustic scene classification and event detection". In: *21st European Signal Processing Conference (EUSIPCO 2013)*. IEEE. 2013, pp. 1–5.

[50] Alexander M Goberman and Lawrence W Elmer. "Acoustic analysis of clear versus conversational speech in individuals with Parkinson disease". In: *Journal of Communication Disorders* 38.3 (2005), pp. 215–230.

[51] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep learning. Book in preparation for MIT Press". In: *URL¡ http://www. deeplearningbook. org* 1 (2016).

[52] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks". In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, pp. 6645–6649.

[53] Guodong Guo and Stan Z Li. "Content-based audio classification and retrieval by support vector machines". In: *IEEE transactions on Neural Networks* 14.1 (2003), pp. 209–215.

[54] Andrey Guzhov et al. "ESResNet: Environmental Sound Classification Based on Visual Domain Models". In: *arXiv preprint arXiv:2004.07301* (2020).

[55] Raia Hadsell, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 1735–1742.

[56] Aki Harma, Martin F McKinney, and Janto Skowronek. “Automatic surveillance of the acoustic activity in our living environment”. In: *2005 IEEE international conference on multimedia and expo*. IEEE. 2005, 4–pp.

[57] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[58] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[59] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9729–9738.

[60] Marti A. Hearst et al. “Support vector machines”. In: *IEEE Intelligent Systems and their applications* 13.4 (1998), pp. 18–28.

[61] Toni Heittola et al. “Context-dependent sound event detection”. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2013.1 (2013), pp. 1–13.

[62] Toni Heittola et al. “Supervised model training for overlapping sound events based on unsupervised source separation”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 8677–8681.

[63] Edward D Herderick, Kirk Cooper, and Nate Ames. “New approach to join SiC for accident-tolerant nuclear fuel cladding”. In: *Advanced Materials & Processes* 170.1 (2012), p. 24.

[64] Shawn Hershey et al. “CNN architectures for large-scale audio classification”. In: *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE. 2017, pp. 131–135.

[65] Geoffrey Hinton et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal processing magazine* 29.6 (2012), pp. 82–97.

[66] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets”. In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[67] R Devon Hjelm et al. “Learning deep representations by mutual information estimation and maximization”. In: *International Conference on Learning Representations*. 2018.

[68] Elad Hoffer and Nir Ailon. “Deep metric learning using triplet network”. In: *International Workshop on Similarity-Based Pattern Recognition*. Springer. 2015, pp. 84–92.

[69] Yedid Hoshen, Ron J Weiss, and Kevin W Wilson. “Speech acoustic modeling from raw multichannel waveforms”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 4624–4628.

[70] Weimin Huang et al. “Scream detection for home applications”. In: *2010 5th IEEE Conference on Industrial Electronics and Applications*. IEEE. 2010, pp. 2115–2120.

[71] Ilija Ilievski et al. “Efficient hyperparameter optimization for deep learning algorithms using deterministic rbf surrogates”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[72] Pattana Intani and Teerapong Orachon. “Crime warning system using image and sound processing”. In: *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*. IEEE. 2013, pp. 1751–1753.

[73] Navdeep Jaitly and Geoffrey Hinton. “Learning a better representation of speech soundwaves using restricted boltzmann machines”. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2011, pp. 5884–5887.

[74] Biing-Hwang Juang and Lawrence R Rabiner. “Automatic speech recognition–a brief history of the technology development”. In: *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara* 1 (2005), p. 67.

[75] Chieh-Chi Kao et al. “R-CRNN: Region-based convolutional recurrent neural network for audio event detection”. In: *arXiv preprint arXiv:1808.06627* (2018).

[76] Prannay Khosla et al. “Supervised contrastive learning”. In: *arXiv preprint arXiv:2004.11362* (2020).

[77] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[78] Anurag Kumar and Bhiksha Raj. “Audio event detection using weakly labeled data”. In: *Proceedings of the 24th ACM international conference on Multimedia*. 2016, pp. 1038–1047.

[79] Pierre Laffitte et al. “Deep neural networks for automatic detection of screams and shouted speech in subway trains”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 6460–6464.

[80] Yann LeCun et al. “Object recognition with gradient-based learning”. In: *Shape, contour and grouping in computer vision*. Springer, 1999, pp. 319–345.

[81] Huakang Li et al. “Robot navigation and sound based position identification”. In: *2007 IEEE International Conference on Systems, Man and Cybernetics*.

[82] Juncheng Li et al. “A comparison of deep learning methods for environmental sound detection”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 126–130.

[83] Junnan Li et al. “Prototypical Contrastive Learning of Unsupervised Representations”. In: *arXiv preprint arXiv:2005.04966* (2020).

[84] Pengcheng Li et al. “An attention pooling based representation learning method for speech emotion recognition”. In: (2018).

[85] Hyungui Lim, Jeongsoo Park, and Y Han. “Rare sound event detection using 1D convolutional recurrent neural networks”. In: *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*. 2017, pp. 80–84.

[86] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[87] Yi-zhou Lin, Zhen-hua Nie, and Hong-wei Ma. “Structural damage detection with automatic feature-extraction through deep learning”. In: *Computer-Aided Civil and Infrastructure Engineering* 32.12 (2017), pp. 1025–1046.

[88] R Bruce Lindsay and RS Shankland. “Acoustics: historical and philosophical development”. In: *Physics Today* 26.12 (1973), p. 55.

[89] Zhu Liu et al. “Audio feature extraction and analysis for scene classification”. In: *Proceedings of First Signal Processing Society Workshop on Multimedia Signal Processing*. IEEE. 1997, pp. 343–348.

[90] Karmele López-de-Ipiña et al. “On the selection of non-invasive methods based on speech analysis oriented to automatic Alzheimer disease diagnosis”. In: *Sensors* 13.5 (2013), pp. 6730–6745.

[91] Carlo Lopez-Tello and V Muthukumar. "Classifying acoustic signals for wildlife monitoring and poacher detection on UAVs". In: *2018 21st Euromicro Conference on Digital System Design (DSD)*. IEEE. 2018, pp. 685–690.

[92] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[93] Min Meng et al. "Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks". In: *Neurocomputing* 257 (2017), pp. 128–135.

[94] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. "Metrics for polyphonic sound event detection". In: *Applied Sciences* 6.6 (2016), p. 162.

[95] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. "TUT database for acoustic scene classification and sound event detection". In: *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE. 2016, pp. 1128–1132.

[96] Annamaria Mesaros et al. "Acoustic event detection in real life recordings". In: *2010 18th European Signal Processing Conference*. IEEE. 2010, pp. 1267–1271.

[97] Annamaria Mesaros et al. "DCASE 2017 challenge setup: Tasks, datasets and baseline system". In: 2017.

[98] Annamaria Mesaros et al. "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.2 (2017), pp. 379–393.

[99] Annamaria Mesaros et al. "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 151–155.

[100] Tomáš Mikolov et al. "Recurrent neural network based language model". In: *Eleventh annual conference of the international speech communication association*. 2010.

[101] Jesus Monge-Alvarez et al. "Audio-cough event detection based on moment theory". In: *Applied Acoustics* 135 (2018), pp. 124–135.

[102] Todd K Moon. "The expectation-maximization algorithm". In: *IEEE Signal processing magazine* 13.6 (1996), pp. 47–60.

[103] Nicolas Morizet et al. “Classification of acoustic emission signals using wavelets and Random Forests: Application to localized corrosion”. In: *Mechanical Systems and Signal Processing* 70 (2016), pp. 1026–1037.

[104] Gregory N Morscher. “Modal acoustic emission of damage accummulation in woven SiC/SiC at elevated temperatures”. In: *Review of progress in quantitative nondestructive evaluation*. Springer, 1999, pp. 419–426.

[105] Elizabeth Murray et al. “Differential diagnosis of children with suspected childhood apraxia of speech”. In: *Journal of Speech, Language, and Hearing Research* 58.1 (2015), pp. 43–60.

[106] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[107] Alireza Nasiri et al. “AudioMask: Robust Sound Event Detection Using Mask R-CNN and Frame-Level Classifier”. In: *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2019, pp. 485–492.

[108] Alireza Nasiri et al. “Online Damage Monitoring of SiC f-SiC m Composite Materials Using Acoustic Emission and Deep Learning”. In: *IEEE Access* 7 (2019), pp. 140534–140541.

[109] Takashi Nozawa, Kazumi Ozawa, and Hiroyasu Tanigawa. “Re-defining failure envelopes for silicon carbide composites based on damage process analysis by acoustic emission”. In: *Fusion Engineering and Design* 88.9-10 (2013), pp. 2543–2546.

[110] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).

[111] Yelakan Berenger Ouattara et al. “KNN and SVM Classification for Chainsaw Identification in the Forest Areas”. In: *International journal of advanced computer science and applications (IJACSA)* 10.12 (2019).

[112] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. “Rethinking CNN Models for Audio Classification”. In: *arXiv preprint arXiv:2007.11154* (2020).

[113] Dimitri Palaz, Ronan Collobert, and Mathew Magimai Doss. “Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks”. In: *arXiv preprint arXiv:1304.1018* (2013).

[114] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. "Recurrent neural networks for polyphonic sound event detection in real life recordings". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 6440–6444.

[115] Ya-Ti Peng et al. "Healthcare audio event classification using hidden Markov models and hierarchical hidden Markov models". In: *2009 IEEE International conference on multimedia and expo*. IEEE. 2009, pp. 1218–1221.

[116] Huy Phan et al. "Audio scene classification with deep recurrent neural networks". In: *arXiv preprint arXiv:1703.04770* (2017).

[117] Huy Phan et al. "DNN and CNN with weighted and multi-task loss functions for audio event detection". In: *arXiv preprint arXiv:1708.03211* (2017).

[118] Huy Phan et al. "Improved audio scene classification based on label-tree embeddings and convolutional neural networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.6 (2017), pp. 1278–1290.

[119] Huy Phan et al. "Random regression forests for acoustic event detection and classification". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.1 (2014), pp. 20–31.

[120] Huy Phan et al. "What makes audio event detection harder than classification?" In: *2017 25th European signal processing conference (EUSIPCO)*. IEEE. 2017, pp. 2739–2743.

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

[122] Karol J Piczak. "ESC: Dataset for environmental sound classification". In: *Proceedings of the 23rd ACM international conference on Multimedia*. 2015, pp. 1015–1018.

[123] Axel Plinge, Rene Grzeszick, and Gernot A Fink. "A bag-of-features approach to acoustic event detection". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 3704–3708.

[124] Boris T Polyak. "Some methods of speeding up the convergence of iteration methods". In: *Ussr computational mathematics and mathematical physics* 4.5 (1964), pp. 1–17.

[125] Jose Portelo et al. “Non-speech audio event detection”. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE. 2009, pp. 1973–1976.

[126] Hendrik Purwins et al. “Deep learning for audio signal processing”. In: *IEEE Journal of Selected Topics in Signal Processing* 13.2 (2019), pp. 206–219.

[127] Regunathan Radhakrishnan, Ajay Divakaran, and A Smaragdis. “Audio analysis for surveillance applications”. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.* IEEE. 2005, pp. 158–161.

[128] Alain Rakotomamonjy and Gilles Gasso. “Histogram of gradients of time–frequency representations for audio scene classification”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.1 (2014), pp. 142–153.

[129] Bharath Ramsundar and Reza Bosagh Zadeh. *TensorFlow for deep learning: from linear regression to reinforcement learning.* " O’Reilly Media, Inc.", 2018.

[130] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems.* 2015, pp. 91–99.

[131] Douglas A Reynolds. “Speaker identification and verification using Gaussian mixture speaker models”. In: *Speech communication* 17.1-2 (1995), pp. 91–108.

[132] Douglas A Reynolds and Richard C Rose. “Robust text-independent speaker identification using Gaussian mixture speaker models”. In: *IEEE transactions on speech and audio processing* 3.1 (1995), pp. 72–83.

[133] Kevin Rhodes et al. “Understanding the degradation of silicon electrodes for lithium-ion batteries using acoustic emission”. In: *Journal of the Electrochemical Society* 157.12 (2010), A1354–A1360.

[134] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386.

[135] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[136] L Sadowski. “Non-destructive investigation of corrosion current density in steel reinforced concrete by artificial neural networks”. In: *Archives of Civil and Mechanical Engineering* 13.1 (2013), pp. 104–111.

[137] Ruslan Salakhutdinov and Geoff Hinton. “Learning a nonlinear embedding by preserving class neighbourhood structure”. In: *Artificial Intelligence and Statistics*. 2007, pp. 412–419.

[138] Justin Salamon and Juan Pablo Bello. “Deep convolutional neural networks and data augmentation for environmental sound classification”. In: *IEEE Signal Processing Letters* 24.3 (2017), pp. 279–283.

[139] Justin Salamon and Juan Pablo Bello. “Unsupervised feature learning for urban sound classification”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 171–175.

[140] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. “A dataset and taxonomy for urban sound research”. In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, pp. 1041–1044.

[141] Dominik Scherer, Andreas Müller, and Sven Behnke. “Evaluation of pooling operations in convolutional architectures for object recognition”. In: *International conference on artificial neural networks*. Springer. 2010, pp. 92–101.

[142] Mike Schuster and Kuldip K Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[143] Benjamin Schuster-Böckler and Alex Bateman. “An introduction to hidden Markov models”. In: *Current protocols in bioinformatics* 18.1 (2007), A–3A.

[144] Ervin Sejdić, Igor Djurović, and Jin Jiang. “Time–frequency feature representation using energy concentration: An overview of recent advances”. In: *Digital signal processing* 19.1 (2009), pp. 153–183.

[145] M Sharda and NC Singh. “Auditory perception of natural sound categories–an fMRI study”. In: *Neuroscience* 214 (2012), pp. 49–58.

[146] Urmila Shrawankar and Vilas Thakare. “Noise estimation and noise removal techniques for speech recognition in adverse environment”. In: *International Conference on Intelligent Information Processing*. Springer. 2010, pp. 336–342.

[147] Bruno da Silva et al. “Evaluation of Classical Machine Learning Techniques towards Urban Sound Recognition on Embedded Systems”. In: *Applied Sciences* 9.18 (2019), p. 3885.

[148] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[149] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[150] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. “A scale for the measurement of the psychological magnitude pitch”. In: *The journal of the acoustical society of america* 8.3 (1937), pp. 185–190.

[151] JG Stone et al. “Stress analysis and probabilistic assessment of multi-layer SiC-based accident tolerant nuclear fuel cladding”. In: *Journal of Nuclear Materials* 466 (2015), pp. 682–697.

[152] Bob L Sturm. “The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use”. In: *arXiv preprint arXiv:1306.1461* (2013).

[153] Woubishet Zewdu Taffese and Esko Sistonen. “Machine learning for durability and service-life assessment of reinforced concrete structures: Recent advances and future directions”. In: *Automation in Construction* 77 (2017), pp. 1–14.

[154] Daiki Takeuchi et al. “Effects of word-frequency based pre-and post-processings for audio captioning”. In: *arXiv preprint arXiv:2009.11436* (2020).

[155] Andrey Temko and Climent Nadeu. “Acoustic event detection in meeting-room environments”. In: *Pattern Recognition Letters* 30.14 (2009), pp. 1281–1288.

[156] Andrey Temko and Climent Nadeu. “Classification of acoustic events using SVM-based clustering schemes”. In: *Pattern Recognition* 39.4 (2006), pp. 682–694.

[157] Yuji Tokozume and Tatsuya Harada. “Learning environmental sounds with end-to-end convolutional neural network”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 2721–2725.

[158] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. “Learning from Between-class Examples for Deep Sound Recognition”. In: *International Conference on Learning Representations*. 2018.

[159] Barbara Tomasino et al. “Identifying environmental sounds: a multimodal mapping study”. In: *Frontiers in human neuroscience* 9 (2015), p. 567.

[160] Joel A Tropp. “Greed is good: Algorithmic results for sparse approximation”. In: *IEEE Transactions on Information theory* 50.10 (2004), pp. 2231–2242.

[161] George Tzanetakis and Perry Cook. “Musical genre classification of audio signals”. In: *IEEE Transactions on speech and audio processing* 10.5 (2002), pp. 293–302.

[162] Burak Uzkent, Buket D Barkana, and Hakan Cevikalp. “Non-speech environmental sound classification using SVMs with a new set of features”. In: *International Journal of Innovative Computing, Information and Control* 8.5 (2012), pp. 3511–3524.

[163] Michele Valenti et al. “A neural network approach for sound event detection in real life audio”. In: *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE. 2017, pp. 2754–2758.

[164] Xavier Valero and Francesc Alias. “Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification”. In: *IEEE Transactions on Multimedia* 14.6 (2012), pp. 1684–1689.

[165] Emmanuel Vincent et al. “From blind to guided audio source separation: How models and side information can improve the separation of sound”. In: *IEEE Signal Processing Magazine* 31.3 (2014), pp. 107–115.

[166] Tuomas Virtanen. “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria”. In: *IEEE transactions on audio, speech, and language processing* 15.3 (2007), pp. 1066–1074.

[167] Tuomas Virtanen. “Sound source separation using sparse coding with temporal continuity objective”. In: *ICMC*. 2003, pp. 231–234.

[168] Feng Wang and Huaping Liu. “Understanding the Behaviour of Contrastive Loss”. In: *arXiv preprint arXiv:2012.09740* (2020).

[169] Kaiwu Wang, Liping Yang, and Bin Yang. “Audio Event Detection and classification using extended R-FCN Approach”. In: *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. 2017, pp. 128–132.

[170] Yun Wang, Leonardo Neves, and Florian Metze. “Audio-based multimedia event detection using deep recurrent neural networks”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 2742–2746.

[171] Yusong Wu et al. *Audio captioning based on transformer and pre-training for 2020 DCASE audio captioning challenge*. Tech. rep. DCASE2020 Challenge, Tech. Rep, 2020.

[172] Lonce Wyse. “Audio spectrogram representations for processing with convolutional neural networks”. In: *arXiv preprint arXiv:1706.09559* (2017).

[173] Xianjun Xia et al. “A survey: neural network-based deep learning for acoustic event detection”. In: *Circuits, Systems, and Signal Processing* 38.8 (2019), pp. 3433–3453.

[174] Dingjun Yu et al. “Mixed pooling for convolutional neural networks”. In: *International conference on rough sets and knowledge technology*. Springer. 2014, pp. 364–375.

[175] Dong Yu and Li Deng. *AUTOMATIC SPEECH RECOGNITION.* Springer, 2016.

[176] Md Zaigham Zaheer et al. “A preliminary study on deep-learning based screaming sound detection”. In: *2015 5th International Conference on IT Convergence and Security (ICITCS)*. IEEE. 2015, pp. 1–4.

[177] Yan Zhang and Dan-jv LV. “Selected features for classifying environmental audio data with random forest”. In: *The Open Automation and Control Systems Journal* 7.1 (2015).

[178] Zhichao Zhang et al. “Deep convolutional neural network with mixup for environmental sound classification”. In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer. 2018, pp. 356–367.

[179] Zhichao Zhang et al. “Learning attentive representations for environmental sound classification”. In: *IEEE Access* 7 (2019), pp. 130327–130339.

[180] Qing Zhou and Zuren Feng. “Robust sound event detection through noise estimation and source separation using NMF”. In: *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), Munich, Germany*. 2017, pp. 16–17.

[181] Xi Zhou et al. “HMM-based acoustic event detection with AdaBoost feature selection”. In: *Multimodal technologies for perception of humans*. Springer, 2007, pp. 345–353.

[182] Boqing Zhu et al. “Learning environmental sounds with multi-scale convolutional neural network”. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[183] Xiaodan Zhuang et al. “Real-world acoustic event detection”. In: *Pattern Recognition Letters* 31.12 (2010), pp. 1543–1551.