HAND CRAFTED VERSUS LEARNED REPRESENTATIONS FOR AUDIO EVENT RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SELVER EZGİ KÜÇÜKBAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JUNE 2023

Approval of the thesis:

HAND CRAFTED VERSUS LEARNED REPRESENTATIONS FOR AUDIO EVENT RECOGNITION

submitted by **SELVER EZGİ KÜÇÜKBAY** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Prof. Dr. Sinan Kalkan
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Adnan Yazıcı
Computer Science, Nazarbayev University _____

Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Assist. Prof. Dr. Emre Akbaş
Computer Engineering, METU _____

Assoc. Prof. Dr. Mustafa Sert
Computer Engineering, Başkent University _____

Assoc. Prof. Dr. Hacer Yalım Keleş
Computer Engineering, Hacettepe University _____

Date:23.06.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Selver Ezgi Küçükbay

Signature :

ABSTRACT

HAND CRAFTED VERSUS LEARNED REPRESENTATIONS FOR AUDIO EVENT RECOGNITION

Küçükbay, Selver Ezgi
Ph.D., Department of Computer Engineering
Supervisor: Prof. Dr. Sinan Kalkan

June 2023, 66 pages

Audio Event Recognition (AER) pertains to identifying the types of events in audio signals. AER is essential for applications requiring decisions based on audio signals, which can be critical, for example, for health, surveillance, and security applications.

Despite the proven benefits of deep learning in obtaining the best representation for solving a problem, AER studies still generally employ hand-crafted representations even when deep learning is used for solving the AER task. Intrigued by this, we investigate whether or not hand-crafted representations (i.e., spectrogram, mel spectrogram, log mel spectrogram, and mel frequency cepstral coefficients) are better than a representation learned using a Convolutional Autoencoder (CAE) or representation learned using a pre-trained network. To the best of our knowledge, our study is the first to ask this question and thoroughly compare feature representations for AER. To this end, we first find the best hop size and window size for each hand-crafted representation and compare the optimized hand-crafted representations with CAE-learned representations. Our extensive analyses on a subset of the AudioSet dataset confirm the common practice in that hand-crafted representations do perform better than learned features by a large margin ($\sim$30 AP). We additionally conduct a comparative

analysis between hand-crafted features and learned representations generated from pre-trained networks. Our findings indicate that the log mel spectrogram continues to exhibit superior performance ($\sim$8 AP).

Moreover, we investigate whether the aforementioned audio features are complementary to each other by exploring different fusion strategies. We show that significant improvements ($\sim$6 AP) can be obtained with an early fusion scheme compared to the best hand-crafted feature.

Keywords: Audio Event Recognition, Audio Event Classification, Deep Learning, Log Mel Spectrogram, Mel Spectrogram, Spectrogram, MFCC, cross-modal attention, fusion

ÖZ

SES OLAYLARI TANIMI İÇİN ELLE ÇIKARILMIŞ VE ÖĞRENİLMİŞ ÖZNİTELİKLERİN KARŞILAŞTIRILMASI

Küçükbay, Selver Ezgi
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Sinan Kalkan

Haziran 2023, 66 sayfa

Ses Olay Tanımı (SOT), ses sinyallerindeki olay türlerinin tanımlanması ile ilgili yapılan çalışmalar bütünüdür. SOT, ses sinyallerine dayalı olarak alınacak kararların etkili olduğu bazı uygulamalarda çok önemli bir rol oynar. Bu uygulamalar; sağlık, gözetlem ve güvenlik uygulamaları olabilir.

Derin öğrenme yöntemlerinin bir problemin çözümünde en iyi çözümleri ürettiği kanıtlanmış olmasına rağmen, SOT çalışmalarında problemin çözümünde derin öğrenme kullanılmasına rağmen, hala, bir uzmanın tasarladığı algoritma çıktısı olan el yapımı öznitelikler kullanılmaktadır. Bununla ilgili olarak, bu tez çalışmasında yaygın olarak literatürde kullanılan el yapımı özniteliklerin (spectrogram, mel spectrogram, logmel spectrogram ve mel frekans sepstral katsayıları) evrişimsel otokodlayıcılar kullanılarak çıkarılmış öznitelikler ve daha önce öğrenilmiş bir ağ ile çıkarılmış öznitelikler ile karşılaştırması yapılmıştır. Bilindiği kadarıyla, bu soruyu soran ve SOT için öznitelik temsillerini bu şekilde kapsamlı bir şekilde ele alan ilk çalışmadır. Bu amaçla, el yapımı öznitleik çıkarımları için en iyi pencere ve atlama boyutu bulunmuş ve optimize edilmiş el yapımı öznitelikler otokodlayıcının çıktısı

olan öznitelikler ile karşılaştırılmıştır. Kapsamlı analizlerimiz neticesinde, AudioSet veri kümesinden oluşturulmuş bir alt veri kümesinde le yapımı özniteliklerin, öğrenilmiş özniteliklerden daha iyi performans ($\sim$30 AP) gösterdiği söylenebilir. Ek olarak el yapımı özniteliklerle önceden eğitilmiş ağlardan elde edilen öğrenilmiş temsiller arasında karşılaştırmalı bir analiz de gerçekleştiriyoruz. Bulgularımız, logmel spektrogramının üstün performansını sürdürdüğünü göstermektedir ($\sim$8 AP).

Ayrıca bahsi geçen ses özniteliklerinin birbirleri ile tamamlayıcı olup olmadığını araştırmak için farklı birleştirme yöntemleri araştırılmıştır. En iyi sonucu elde ettiğimiz el yapımı öznitelikle kıyasladığımızda erken birleştirme düzeniyle önemli geliştirmelerin elde edilebileceğini gösterilmiştir ($\sim$6 AP) .

Anahtar Kelimeler: Ses Olay Tespiti, Ses Olay Sınıflandırması, Derin Öğrenme, Spectrogram, Mel Spectrogram, MFCC, çapraz model etkisi, füzyon

This thesis is dedicated to my precious, sun and stars:

Melek Ada Küçükbay

on behalf of all daughters in Turkey.

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor, Prof. Dr. Sinan Kalkan, for all his patience, guidance, knowledge, help, and support over the years. I have learned many things from him.

I cannot thank Prof. Adnan Yazıcı enough for his kind and helpful style while pushing me to do my best in the thesis project or even in my motherhood journey.

I am incredibly grateful to my advisors for our friendly chats during the meetings and for making me laugh while my stress level was on top. I would only ever finish this thesis with you.

I would also like to thank my doctoral thesis monitoring committee member - not only a member; he provided me a Ph.D. opportunity-, Assoc. Prof. Dr. Mustafa Sert for his valuable feedback and discussions with me.

I would like to thank my thesis committee, Dr. Emre Akbaş and Dr. Hacer Yalım Keleş for their insightful comments and encouragement.

My most sincere thanks go out to my family and my friends, who remind me of my energy whenever I have difficulties and encourage me for doing my best every time.

My baby, Melek Ada, you are the best thing in my life. I passed the doctoral qualification exam while you are kicking me inside. I read literature while waiting for you to wake up from your short naps. I ran the codes while you are spraying your food on my face. And I wrote while you sing in front of me with a beautiful harmony. The journey was too long and hard, but we grew up together. Thank you, my sunshine, my stars, the queen of my dreams, my love, my everything.

Last but not least, to Serkan, my dear, we shared every moment of life, and you believed in me more than I did. We laugh, cry, hope, fail, and succeed all together. Now, we are at the end of this hard and long way, let's move on to the next journey.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xx

CHAPTERS

1 INTRODUCTION 1

1.1 Problem Definition: Audio Event Recognition 2

1.2 Scope and Contributions of the Thesis 3

1.3 The Outline of the Thesis 4

2 RELATED WORK 5

2.1 Studies on Audio Event Analysis 6

2.2 Studies Analyzing Features for Audio Event Analysis 8

2.3 Studies about Fusion Schemes for Audio Event Recognition 9

2.4 A Summary of the Literature and Our Contributions 10

3 AEDNET: MODEL AND METHODS FOR AUDIO EVENT RECOGNITION . . . 15

3.1 Problem Formulation . . . 15

3.2 The CNN Architecture and Objective . . . 15

3.3 AER with Hand-crafted Representations . . . 18

3.3.1 Spectrogram - $\phi_{spec}(\mathbf{x})$ . . . 18

3.3.2 Mel spectrogram features - $\phi_{mel}(\mathbf{x})$ . . . 20

3.3.3 Log mel spectrogram features - $\phi_{logmel}(\mathbf{x})$ . . . 21

3.3.4 Mel Frequency Cepstral Coefficients - $\phi_{MFCC}(\mathbf{x})$ . . . 21

3.4 AER with Raw Audio Waveform . . . 22

3.5 AER with Learned Features from a Convolutional Autoencoder . . . 22

3.6 AER with Features from a Pre-trained Network . . . 24

4 ANALYSIS OF HAND-CRAFTED AND LEARNED REPRESENTATIONS FOR AUDIO EVENT RECOGNITION . . . 27

4.1 Evaluation Measure . . . 27

4.2 Utilized Dataset . . . 27

4.3 Experimental Results . . . 29

4.3.1 Analysis 1: Analyzing Features and their Hyper-parameters (RQ1, RQ2) . . . 29

4.3.1.1 Analyzing the effect of window and hop sizes . . . 30

4.3.1.2 Analyzing the effect of the number of mels . . . 30

4.3.1.3 Analyzing the effect of the number of coefficients in MFCC features . . . 30

4.3.1.4 Comparison among Hand-crafted Features . . . 30

4.3.2 Analysis 2: Learning-based Representation (RQ2) . . . 33

4.3.2.1 Tuning the Autoencoder Hyper-parameters 33
4.3.2.2 Analyzing the effect of normalization 34
4.3.2.3 Analyzing the learned representations from a pre-trained network 35
4.3.3 Analysis 3: Comparing All Representations (RQ3) 35
4.3.3.1 Overall Comparison 36
4.3.3.2 Robustness against the amount of training samples . . . 37
4.3.3.3 Robustness against the noise (RQ3) 38
5 MULTI MODAL FUSION FOR AUDIO EVENT RECOGNITION 41
5.1 Implementation Details of Fusion Schemes 42
5.1.1 Early Fusion Scheme 43
5.1.1.1 Concatenation-based Early Fusion 44
5.1.1.2 Attention-based Early Fusion 44
5.1.2 Late Fusion Scheme 46
5.1.2.1 Concatenation-based Late Fusion 47
5.1.2.2 Attention-based Late Fusion 48
5.2 Experimental Results of Fusion Schemes 48
5.2.1 Experiment 1: Early Fusion Schemes 48
5.2.2 Experiment 2: Late Fusion Schemes 51
6 CONCLUSION AND DISCUSSION 53
6.1 Answers to the Research Questions (RQ1-RQ4) 53
6.2 Limitations and Future Work 56
REFERENCES 59

LIST OF TABLES

TABLES

Table 2.1 A comparison of AED and AER studies in terms of the feature representations they use. 12

Table 3.1 AEDNet architecture details. 17

Table 3.2 The convolutional autoencoder we use for learning features. 23

Table 4.1 The AudioSet subset we used in our experiments. The number of samples is listed. 28

Table 4.2 Individual class performances for log mel spectrogram features. . . . 33

Table 4.3 The effect of kernel sizes and strides on convolutional autoencoder. See Table 3.2 for the layer numbering.'*' denotes cases where training diverged. 34

Table 4.4 A comparison of the best settings of hand-crafted features. Average signal preprocessing time refers to the average time in milliseconds for processing a signal i.e. reading the raw signal, extracting the feature from the signal, and appending it to the data frame before feeding it to the architecture. *The input audio signal is processed without windows. . . . 36

Table 5.1 A comparison of fusion studies in terms of the schemes and methods they use for audio signal domain 42

Table 5.2 A comparison of concatenation-based early fusion of hand-crafted features and their baseline performance results on the AEDNet. 50

Table 5.3 A comparison of attention-based early fusion of hand-crafted features and their baseline performance results on the AEDNet 50

Table 5.4 A comparison of late fusion of hand-crafted features and their baseline performance results 51

LIST OF FIGURES

FIGURES

Figure 1.1 The general framework of an AER system. Various types of sound events are collected as raw audio signals and from these signals, features are extracted. After selecting the different features, a classification model is applied to the input features and results are obtained. . . 2

Figure 2.1 The proposed method in [18] is illustrated in the conceptual diagram. The analysis pipeline involves two distinct steps: Firstly, handcrafted audio features are computed on the raw signal. Secondly, a supervised CNN, which has been trained to differentiate between different context classes, is utilized as a feature extractor. [Figure source: [18]] 7

Figure 2.2 Early fusion mechanism and overall schema in [22]. Log mel spectrogram and MFCC features are extracted and the temporal frequency attention mechanism is applied to the features as early fusion. [Figure source: [22]] 9

Figure 2.3 The Multi-level Attention Fusion network (MAFnet) is taken from the original paper [7]. Firstly, a video is divided into T non-overlapping clips. Then, pre-trained models are employed to extract audio and visual information from each clip. These clip features are subsequently passed through the modality and temporal attention module to generate a global feature representation that encompasses multimodal and temporal information. This global feature is utilized to predict the label of the video. [Figure Source: [7]] 11

Figure 3.1 WAL-Net from Shah et al. [51] consists of 6 convolutional layers followed by a max pooling layer and 2 convolutional layers followed by a global average pooling layer. [Figure source: [51]] 16

Figure 3.2 AEDNet network architecture (modified from Shah et al. [51]) is composed of many convolution and pooling layers. The shapes of the convolution layer and pooling layers change according to input type; i.e. raw audio waveform ($\mathbf{x}$) or extracted/learned features ($\phi(\mathbf{x})$). 18

Figure 3.3 The diagram illustrates the commonly employed hand-crafted features. The arrows indicate the process of utilizing mel-scale filter banks to transform a spectrogram into a mel-spectrogram, followed by the computation of the logarithm of the mel-spectrogram to generate a log mel spectrogram. 19

Figure 3.4 A block diagram listing the steps of spectrogram feature extraction for audio event detection. 20

Figure 3.5 Wav2Vec2 model network architecture (from Baevski et al. [5]). The framework demonstrates the simultaneous acquisition of contextualized speech representations and a repository of discretized speech units. [Figure source: [5]] 24

Figure 4.1 Top two main classes of 527 audio classes that are collected from 2,084,320 YouTube videos. All audio classes may have multiple classes. The labels are given by human annotators. To nominate segments for annotation, YouTube metadata and content-based search is used. [Figure Source: [17]] 29

Figure 4.2 The effect of window and hop size on spectrogram-based features. 31

Figure 4.3 The effect of the number of mels for the log mel spectrogram feature (with 1024 as the window size and 512 as the hop size). 32

Figure 4.4 Effects of different window and hop size options in log mel spectrogram, mel spectrogram, and spectrogram features. 32

Figure 4.5 Comparison of the AEDNet performance for learned representation, hand-crafted features, and raw audio waveform with and without normalization. 35

Figure 4.6 In order to analyze the effect of pre-trained network outputs on learned representations, we used the Wav2Vec2 model [5]. 36

Figure 4.7 To see the effect of features on size, different methods of feature extraction on datasets of different sizes are performed. The values on the y-axis represent the ratios taken from the original dataset. 37

Figure 4.8 Additive white noise is applied to the dataset to see the robustness of features. 38

Figure 5.1 Three main different fusion strategies, namely early-fusion, joint (mid-fusion), and late-fusion. Early- fusion concatenates raw input or extracted features. Mid-fusion joins features at the input level, but the loss is propagated. Late-fusion aggregates predictions. [Figure Source: [21]] 43

Figure 5.2 The general process for early fusion using concatenation is shown in the diagram. Hand-crafted features are extracted and then combined vertically through concatenation. Any combination of features can be used in this process. Once the concatenated features are obtained, they are fed into AEDNet as a new feature representation. 44

Figure 5.3 The general process for attention-based early fusion is shown in the diagram. The inputs for the attention module can be any combination of hand-crafted features, as long as they have the same dimensions. 45

Figure 5.4 The Temporal Attention (TA) mechanism [11]. The query matrix is used to calculate the covariance between each sample in the feature at the first channel and the related samples in the dependency scope from the feature at the second channel by operating on the key matrix. [Figure Source: [11]] 46

Figure 5.5 The diagram shows the design of the concatenation-based late fusion. Hand-crafted features are input into AEDNet without the last two layers of the original model, and the outputs are concatenated. Finally, a classifier produces a final output 47

Figure 5.6 The diagram shows the design of the cross-modal attention module. Hand-crafted features are input into AEDNet without the last two layers of the original model, and the outputs are then processed by the attention module. Finally, a classifier produces an output 49

LIST OF ABBREVIATIONS

AED	Audio Event Detection
AER	Audio Event Recognition
MER	Multimedia Event Recognition
AEA	Audio Event Analysis
CAE	Convolutional Autoencoder
MFCC	Mel Frequency Cepstral Coefficients
AP	Average Precision
mAP	Mean Average Precision
MSE	Mean Squared Error

CHAPTER 1

INTRODUCTION

Advances in sensory hardware, signal processing, and machine learning have boosted applications where identifying various forms of events is beneficial. Example applications include recognizing intruders at a border between countries using seismic sensors, monitoring wildlife for endangered animals from audio sensors, classifying different types of blood cells from high-resolution microscopic images, and detecting oil spillage in oceans from airborne video cameras.

Multimedia Event Recognition (MER) is the problem of identifying such events from different forms of signals captured from the environment. These signal forms commonly include audio, image, and video data, which carry complementary information about different aspects of the events.

Existing MER studies and applications have often used visual data such as images and videos for identifying events since visual data is easier to process and interpret [1]. However, MER from visual data is limited to the recognition of only events with a clear line of sight. Events that are not visible require using other modalities, e.g., seismic, magnetic, and audio signals.

As events generally produce sound, the use of audio signals for event recognition, i.e., Audio Event Recognition (AER), is a common problem in MER. Compared to visual data, AER does not require a line of sight. Moreover, audio sensors are considerably cheaper compared to cameras. Finally, audio signals are lower dimensional, requiring lower space and computational complexities.

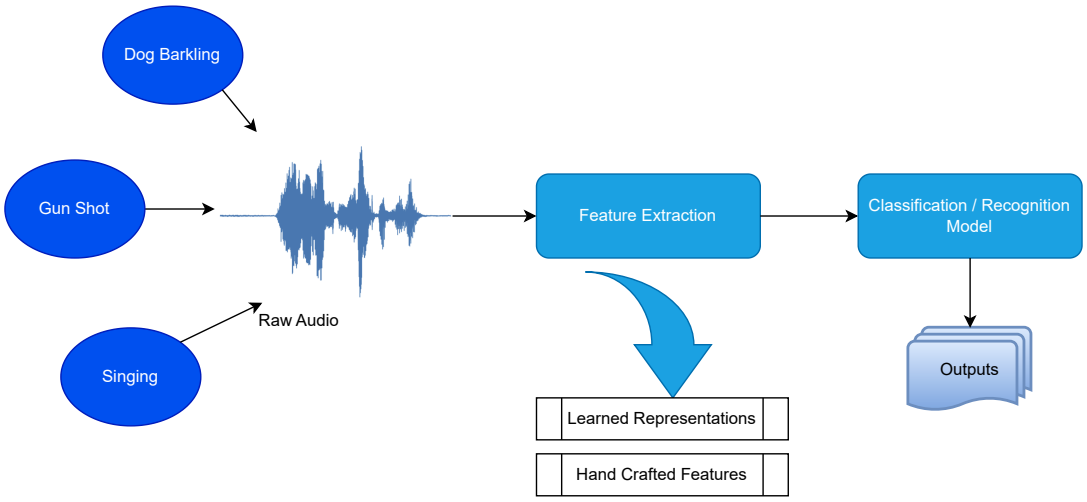


Figure 1.1: The general framework of an AER system. Various types of sound events are collected as raw audio signals and from these signals, features are extracted. After selecting the different features, a classification model is applied to the input features and results are obtained.

1.1 Problem Definition: Audio Event Recognition[1]

In this thesis, we focus on the Audio Event Recognition (AER) problem. AER is a single-label or multi-label classification problem wherein we are interested in estimating the labels of the events, such as singing, car horn, and talking, in audio signals.

AER is generally performed in two steps (Figure 1.1 – see also [4]): (1) Feature extraction, and (2) event classification/recognition. The first step, feature extraction, pertains to obtaining crucial information in the audio signals provided as input whereas the second step tries to find a mapping from the rich feature representations to the target event labels.

Before the rise of deep learning, these two steps used to be handled separately and independently. The feature extraction step was addressed using hand-crafted features: The mainstream approach was to first preprocess the audio signal and extract the low-level descriptive features [33, 36, 43, 59]. Thanks to deep learning, it is now possible to learn representations that are more suitable to the tasks to be solved and provide

[1] Some text in this section is borrowed from our MTAP'22 paper [29].

better performance than hand-crafted features [24, 38].

Although deep learning is known to obtain the most suitable representation for a problem, the learning-based AER studies still employ hand-crafted features and use them as input data for deep networks instead of adopting an end-to-end approach [24, 27, 38, 55, 57, 66]. This is peculiar when compared to the findings and common practice in, e.g. computer vision [28, 49]. The literature lacks a systematic investigation that can shed more light on this point. Therefore, a comparison of hand-crafted and learned representations for audio event recognition is necessary (**Limitation 1**).

Furthermore, different audio features represent different aspects of audio signals and therefore, AER can benefit from combining the complementary aspects of audio features, whether hand-crafted or learned (**Limitation 2**).

1.2 Scope and Contributions of the Thesis

In this thesis, we are interested in mitigating the two aforementioned limitations. For the first limitation, we investigate whether or not hand-crafted features are better than learned features in a deep-learning approach. For the second, we explore various strategies for fusing different feature representations to see whether the complementary nature of audio features can be beneficial for AER.

To be specific, we ask and answer the following research questions:

RQ1. Are hand-crafted representations robust against hyper-parameters such as window size and hop size? Is there a specific hand-crafted representation method that is considered the best for AER?

RQ2. Do learned representations perform better than hand-crafted representations?

RQ3. Do learned and hand-crafted representations compare well under low data regimes or under noise?

RQ4. Are different audio features complementary? Could we increase the performance of AER by fusing different audio features?

1.3 The Outline of the Thesis

Chapter 2 explains Audio Event Analysis studies and presents the previous works related to AEA studies with different perspectives: studies on AEA and studies on feature extraction methods in AEA.

In Chapter 3, we describe our method and explain the details of the architecture. We briefly present the designed system, the dataset used, the feature extraction methods, and the details of the classifier.

In Chapter 4, we describe our setup and present our results. In Chapter 5 cross-modal fusion schemes are given and the results are presented.

Finally, Chapter 6 concludes the paper by summarizing our findings regarding the research questions (RQ1-RQ4) and the literature.

CHAPTER 2

RELATED WORK[1]

The primary goal of an AER system is to accurately identify and classify specific sound events or activities within a long audio stream. The objective is to detect and label individual events of interest, such as identifying different types of musical instruments, recognizing environmental sounds, detecting speech, or identifying specific actions or events in audio data. The aim is to develop algorithms and systems that can automatically analyze and understand the content of audio recordings.

In the domain of audio event analysis, feature extraction plays a pivotal role as a crucial step in the assessment and characterization of audio signals, enabling the extraction of meaningful and discriminative representations that capture the inherent properties and distinguishing attributes of the audio content under investigation. These extracted features serve as the foundation for subsequent stages in audio signal processing, including classification, recognition, event detection, and other tasks essential for audio analysis and understanding.

In this chapter, we first review approaches to AER and then discuss studies that analyze the features of AER. Finally, we present studies that use fusion techniques in the feature phase for AER.

In many of the examined studies, the term AED (Audio Event Detection) is used. However, these studies have not been specifically utilized as a comprehensive approach to the AED problem. Since the literature often employs the AED and AER (Audio Event Recognition) concepts interchangeably, our focus in this chapter has been on feature extraction. As both AER and AED problems utilize common fea-

[1] Some text in this chapter is borrowed from our MTAP'22 paper [29].

tures, studies addressing both problems have been collectively analyzed in order to better explore the shared aspects and examine their interrelationships.

2.1 Studies on Audio Event Analysis

With advances in deep learning, AER and AED studies have relied heavily on deep learning methods for specific tasks. Whether deep learning is used or not, AER studies rely on feature representations extracted from the audio signal [17, 41, 45, 46, 54]. The methods used for obtaining feature representations can be broadly grouped into three: (i) Hand-crafted feature extraction methods. (ii) Learned feature representations. (iii) Hybrid methods.

Hand-crafted methods generally transform the audio signal into the frequency domain and obtain rough coefficients in this domain. For example, using the spectrogram of the audio signal [6, 10, 20, 27], its representation in the mel domain [15, 48, 57, 66] or the log mel domain [9, 20, 30, 47, 51, 55, 61] are quite common. The visualizations and detailed explanations of these, frequently used three features are provided in Chapter 3

The learning-based methods, on the other hand, aim to obtain the features that are most suitable for the AER task at hand. This can be done in different ways: either by (i) using unsupervised or self-supervised learning strategies such as auto-encoders [10, 20, 65] or contrastive learning [16, 50], to train a feature representation extraction network, then training another network for the AER task or (ii) directly providing raw audio signals to a single network designed for the AER task and performing end-to-end learning [12, 31].

Despite the success of learning-based representations in other classification problems such as object recognition, AED studies, regardless of deep learning, still prefer hand-crafted features since the hand-crafted features provide a very compact representation that summarizes the audio signal. Moreover, these spectrum-based representations can be viewed and inspected to provide insight into the event or audio signal.

Hybrid methods take advantage of both types of methods and combine hand-crafted

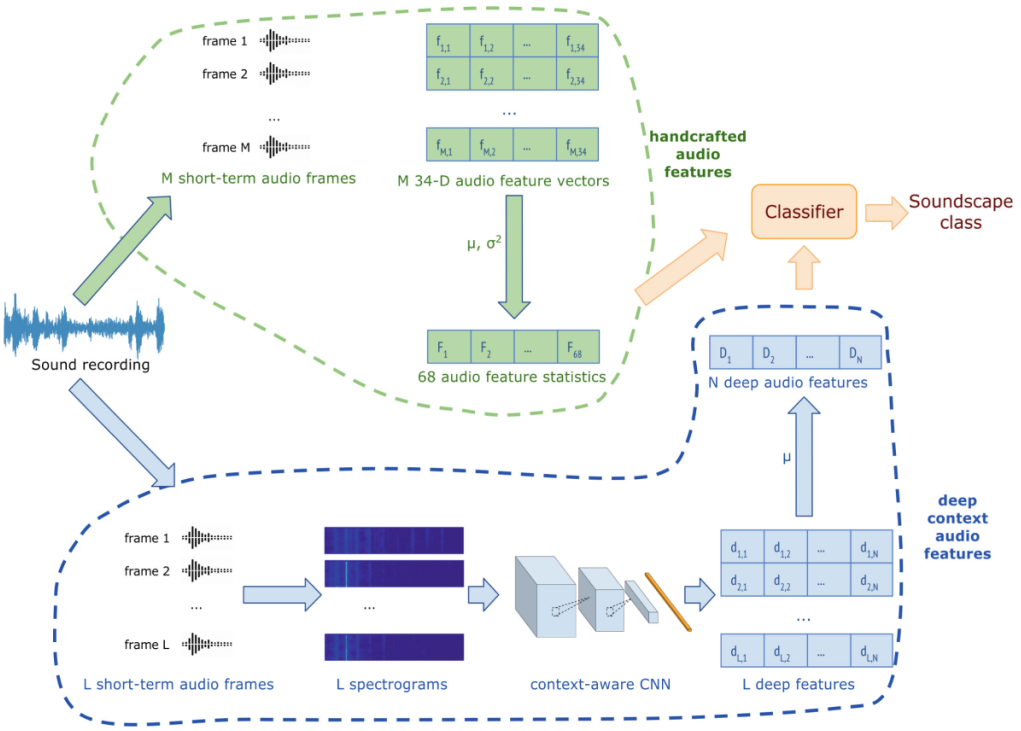


Figure 2.1: The proposed method in [18] is illustrated in the conceptual diagram. The analysis pipeline involves two distinct steps: Firstly, hand-crafted audio features are computed on the raw signal. Secondly, a supervised CNN, which has been trained to differentiate between different context classes, is utilized as a feature extractor. [Figure source: [18]]

features with learning-based features [18]. As shown in Figure 2.1, while hand-crafted features are computed on one side, learned features are computed on the other side, and then they are fed into separate channels to the classifier for urban sound recognition purposes. The learned features are context-aware features since they are derived from the whole dataset by a training procedure to accomplish a certain task; on the other hand, hand-crafted features are extracted from an algorithm and are based on expert knowledge. Since learned features are difficult to interpret and hand-crafted features are narrow in scope, it is better to use early or late fusion in the feature extraction phase. The classification accuracy is improved significantly when the learned feature representations are combined with the typical hand-crafted features.

2.2 Studies Analyzing Features for Audio Event Analysis

In the field of audio signal processing, the extraction of features plays a vital role in audio event analysis. Audio features are essential representations derived from the raw audio signal that capture relevant characteristics related to the underlying audio events. The desired properties of these features include easy adaptability, robustness against noise, straightforward implementation, and the presence of appropriate smoothing characteristics [58].

Despite its importance for the AER problem, there is limited analysis of the feature representation methods for the AER problem and such analyses generally pertain to only a small subset of the commonly used feature extraction methods used in the literature. Moreover, the existing studies do not provide an extensive analysis of the (hyper) parameters of the methods (see Table 2.1).

For example, Cakir et al. [9] utilized log mel spectrogram features with 40 ms window size and 20 ms hop size and 40 mel bands. They also used MFCC features with 40 ms window size and 20 ms hop size and 40 mel bands. Cakir and Virtanen [10] carried out their study based on feature extraction over the raw waveform. They also extracted log mel spectrograms with different sampling rates in audio samples, so they changed their feature dimensions by using different sampling rates. They took 40 ms as window size and 20 ms as hop size in log mel spectrogram feature representation. Li et al. [35] used different features such as 23 mel band MFCC with 20 ms window size and 10 ms hop size, MFCC, and log mel spectrogram features with 60 and 200 mel bands separately. Piczak et al. [47] used 60 mel band log mel spectrogram features with 25 ms window size and 10 ms hop size. Xu et al. [65] took 20 ms as window size and 10 ms as hop size and 24 mel bands for MFCC feature. Kong et al. [26] used learned representations and fusion of log mel spectrogram features to tag the audio files in the AudioSet dataset. They analyzed different hop sizes and different coefficients only for the log mel spectrogram.

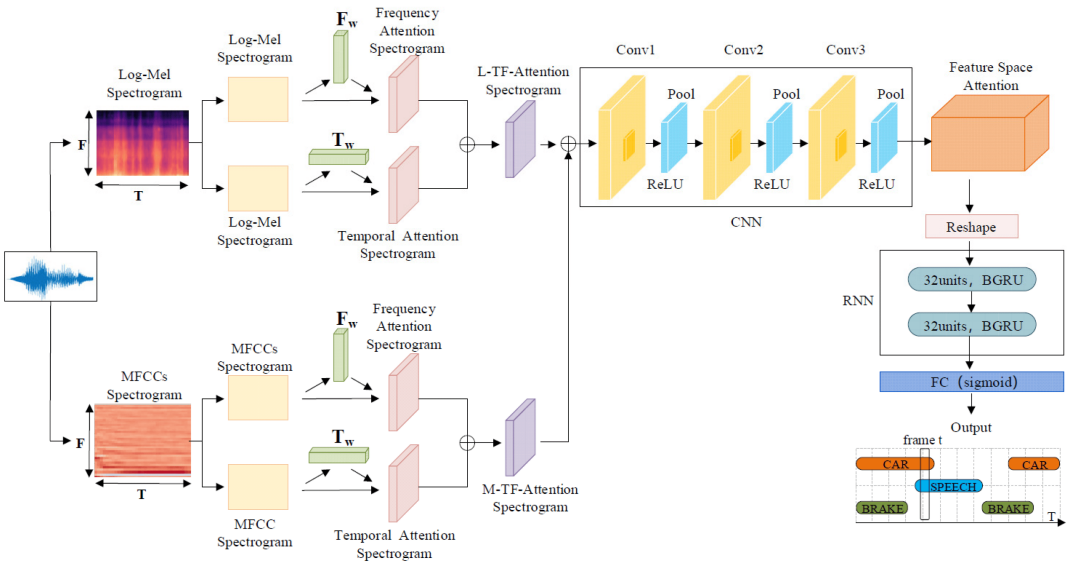


Figure 2.2: Early fusion mechanism and overall schema in [22]. Log mel spectrogram and MFCC features are extracted and the temporal frequency attention mechanism is applied to the features as early fusion. [Figure source: [22]]

2.3 Studies about Fusion Schemes for Audio Event Recognition

It is important to use fusion schemes based on basic statistics or cross-modal attention in the feature extraction phase. In some studies, traditional single acoustic features are unable to adequately capture the key feature information of audio events, leading to poor performance in classification models. Jin et al. [22] used temporal and feature space attention mechanism as depicted in Figure 2.2 as early fusion and improved the baseline system 12.4% and 25.2%. Jo et al. [23] use concatenation-based early fusion in their study in order to fuse the spatial features and confirmed that their combination and proposed feature scheme yielded better performance than state-of-the-art methods. McLoughlin et al. [40] studied the use of three different features: spectrograms, cochleograms, and constant-Q transform-based images. They also explored feature fusion by using CNN for each pair of features and found that the fusion of spectrograms and cochleograms was beneficial. Lee et al. [32] propose a method for addressing the problem of polyphonic sound event localization and detection. In the paper, the dual cross-modal attention method is utilized for specified tasks. The first cross-modal attention is used to capture the relationship between audio and visual

features. They used log mel spectrogram features in their study and reported that they achieve competitive results and outperforms other state-of-the-art approaches in terms of accuracy. In [44], a novel approach that utilized attention mechanisms to improve the fusion of information from different modalities is proposed. Attention bottlenecks are introduced and they enable the network selectively attend to the most information from each modality and enhance the fusion process. One of the multi-modal tasks that they evaluate their approach is audio-visual speech recognition. They use AudioSet in their study, experiment with different fusion techniques, and work with spectrogram features. They improve performance by using the cross-attention technique at lower computational cost and achieved state-of-the-art results on benchmarks. Tan [56] proposed a multi-modal network with cross-modal attention by processing information in both audio and visual modalities. The method incorporates audio-visual fusion features, a heterogeneous graph-like approach for the aggregation of features. They use learned representation for the feature extraction phase and fuse and finally classify them. Experimental results indicate favorably over existing methods in audio-event localization tasks.

The study [7] introduces the Multi-level Attention Fusion network (MAFnet), a novel architecture designed to dynamically fuse visual and audio information for event recognition. The researchers integrate both modalities at various levels within the visual and audio pathways that are depicted in Figure 2.3. Moreover, the network dynamically emphasizes a specific modality during a relevant time window for accurate event classification. Experimental evaluations on the AVE (Audio-Visual Event), UCF51, and Kinetics-Sounds datasets demonstrate the effectiveness of our approach in enhancing the accuracy of audio-visual event classification.

2.4 A Summary of the Literature and Our Contributions

Previous research has primarily concentrated on extracting specific features with predefined parameters. These studies typically choose an extraction method and evaluate its performance based on the selected architecture, as shown in Table 2.1. However, the table indicates that there is a lack of comprehensive analyses that consider well-established feature extraction methods with varying parameter settings on the same

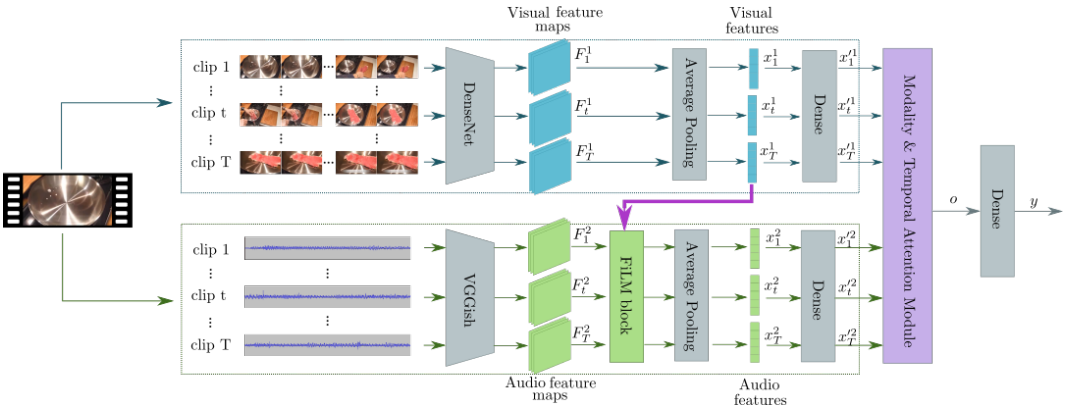


Figure 2.3: The Multi-level Attention Fusion network (MAFnet) is taken from the original paper [7]. Firstly, a video is divided into T non-overlapping clips. Then, pre-trained models are employed to extract audio and visual information from each clip. These clip features are subsequently passed through the modality and temporal attention module to generate a global feature representation that encompasses multi-modal and temporal information. This global feature is utilized to predict the label of the video. [Figure Source: [7]]

dataset

In this thesis, we perform an in-depth representation analysis for AER using a subset of the AudioSet dataset [17] to seek answers to the research questions (RQ1-RQ4) outlined in Chapter 1. To be precise:

- We compare the performance of a deep convolutional network for AER when its input is (i) the widely used hand-crafted features namely spectrogram, mel spectrogram, log mel spectrogram, or MFCC features, (ii) raw audio waveform or (iii) learned representations using convolutional autoencoder (CAE) and (iv) learned representation using the pre-trained network.

- We conduct a systematic analysis of various settings for the methods, including spectrogram, mel spectrogram, and log mel spectrogram features with different window and hop sizes, as well as MFCC features with different coefficients. Through this analysis, we identify the optimal setting for each method, determining the configuration that yields the best performance.

Table 2.1: A comparison of AED and AER studies in terms of the feature representations they use.

Paper	**Ref.**	**Spect.**	**Mel Spect.**	**Log Mel Spect.**	**MFCC**	**Learned Repres.**
Zhuang et al., 2008	[67]	×	×	✓	✓	×
Piczak, 2015	[47]	×	×	✓	×	×
Stowell et al., 2015	[54]	×	×	×	✓	×
Hershey et al., 2016	[20]	✓	×	×	×	×
Piczak, 2016	[48]	×	✓	×	×	×
Xu et al., 2016	[65]	×	×	×	✓	✓
Mesaros et al., 2016	[41]	×	×	×	✓	×
Cakir et al., 2017	[9]	×	×	✓	×	×
Li et al., 2017	[35]	×	×	✓	✓	×
Cakir and Virtanen, 2018	[10]	✓	×	✓	×	✓
Shah et al., 2018	[51]	×	×	✓	×	×
Becker et al., 2018	[6]	✓	×	×	×	×
Wang et al., 2019	[14]	×	×	×	✓	×
Kothinti et al., 2019	[27]	✓	×	×	×	×
Zhang et al., 2019	[66]	×	✓	×	×	×
Turpault and Serizel, 2020	[57]	×	✓	×	×	×
Sun and Ghaffarzadegan, 2020	[55]	×	×	✓	×	×
Kong et al., 2020	[26]	×	×	✓	×	✓
Kwag and Chunk, 2020	[30]	×	×	✓	×	×
Maria and Jeyaseelan, 2021	[39]	×	×	×	✓	×
Eutizi and Benedetto, 2021	[15]	×	✓	×	×	×
Wang et al., 2021	[61]	×	×	✓	×	×
Nagrani et al., 2021	[44]	✓	×	×	×	×
Lee et al., 2022	[32]	×	×	✓	×	×
Tan, 2022	[56]	×	×	×	×	✓
Our work		✓	✓	✓	✓	✓

- In our evaluation, we assess the performance of the considered methods not only in terms of their AER performance but also in their ability to maintain

robustness when exposed to various types of noise present in the input signal.

- We try to fuse the features to gather better accuracy. We utilize early and late fusion for different features to feed the classifier with the output of cross-modal attention.

CHAPTER 3

AEDNET: MODEL AND METHODS FOR AUDIO EVENT RECOGNITION[1]

In this chapter, we first introduce a convolutional network for AER, and present the alternative audio feature representations (hand-crafted or learned) with which the network can be used.

3.1 Problem Formulation

In AER, given an audio signal $\mathbf{x} \in \mathbb{R}^{W \times T}$, we are interested in finding the label $\mathbf{y} \in \mathcal{Y}$ of the audio event in $\mathbf{x}$, where $\mathcal{Y}$ is a set of discrete labels such as 'singing', 'water', 'vehicle', 'speech', and 'guitar'. In AER approaches, generally a hand-crafted or learned feature extractor $\phi(\mathbf{x}) \in \mathbb{R}^{N \times C}$ is used to extract a representation for $\mathbf{x}$. Depending on the feature-extraction method, N and the length C can take on different values according to the parameter settings of the method (e.g. window size, stride, etc.). In our dataset, audio recordings can have multiple classes; on average, there are 2.7 labels per recording.

3.2 The CNN Architecture and Objective

In this thesis, we first construct our CNN-based architecture similar to the model of Shah et al. [51]. Shah et al. designed a fully convolutional network in order to handle audio signals of variable length – see Figure 3.1[2]. Their main motivation was localizing the events in the AudioSet dataset [17]. We chose the model of Shah et al.

[1] Some text in this chapter is borrowed from our MTAP'22 paper [29].

[2] In our paper [29], we referred to it as AEDNet, and to maintain consistency and avoid any confusion, we continued to use the same name throughout the entire thesis.

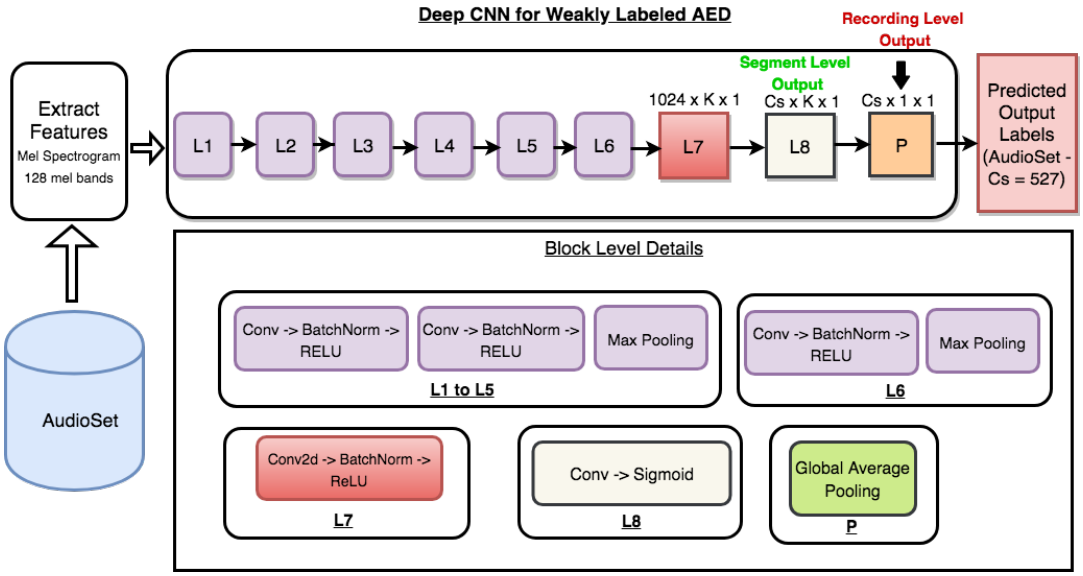


Figure 3.1: WAL-Net from Shah et al. [51] consists of 6 convolutional layers followed by a max pooling layer and 2 convolutional layers followed by a global average pooling layer. [Figure source: [51]]

[51] since it is a deep CNN model, it is easy to modify, and it provides reasonable results on the challenging AudioSet dataset. In our study, we use the same dataset and adapt the network in order to be able to handle smaller feature dimensions and name the architecture as AEDNet – see Figure 3.2 for an overall view of the network and Table 3.1 for the details of all layers.

The AEDNet architecture employs max-pooling after each convolutional layer to reduce the dimensionality. In addition, each layer is followed by a batch normalization layer for regularization. In total, the network has ten convolutional layers, which, as shown by Shah et al. yield promising results for the AER problem. Deeper networks could have been designed and tested; however, for the purposes of comparing the different representations, the AEDNet architecture in Figure 3.2 is sufficient.

The AEDNet is trained with the binary cross-entropy loss, irrespective of the type of feature used (hand-crafted or learned):

$$\mathcal{L}_{AER} = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} [-y_c \times \log p_c - (1 - y_c) \times \log(1 - p_c)] \,, \tag{3.1}$$

where $y_c \in \{0, 1\}$ is the target output for the class $c \in \mathcal{Y}$, and $p_c \in [0, 1]$ is the

Table 3.1: AEDNet architecture details.

Layer	Filter Size	# of Filters	Activation
Convolution 2D	3 x 3	16	
BatchNormalization	-	-	
Activation	-	-	ReLU
MaxPooling	2 x 2	-	-
Convolution 2D	3 x 3	32	
BatchNormalization	-	-	
Activation	-	-	ReLU
MaxPooling	2 x 2	-	-
Convolution 2D	3 x 3	64	
BatchNormalization	-	-	
Activation	-	-	ReLU
Convolution 2D	3 x 3	64	
BatchNormalization	-	-	
Activation	-	-	ReLU
MaxPooling	2 x 2	-	-
Convolution 2D	3 x 3	128	
BatchNormalization	-	-	
Activation	-	-	ReLU
Convolution 2D	3 x 3	128	
BatchNormalization	-	-	
Activation	-	-	ReLU
MaxPooling	2 x 2	-	-
Convolution 2D	3 x 3	256	
BatchNormalization	-	-	
Activation	-	-	ReLU
Convolution 2D	3 x 3	256	
BatchNormalization	-	-	
Activation	-	-	ReLU
MaxPooling	2 x 2	-	-
Convolution 2D	3 x 3	512	
BatchNormalization	-	-	
Activation	-	-	ReLU
MaxPooling	2 x 2	-	-
Convolution 2D	2 x 2	1,024	
BatchNormalization	-	-	
Activation	-	-	ReLU
Convolution 2D	1 x 1	16	
Activation	-	-	sigmoid
GlobalAveragePooling	-	-	-

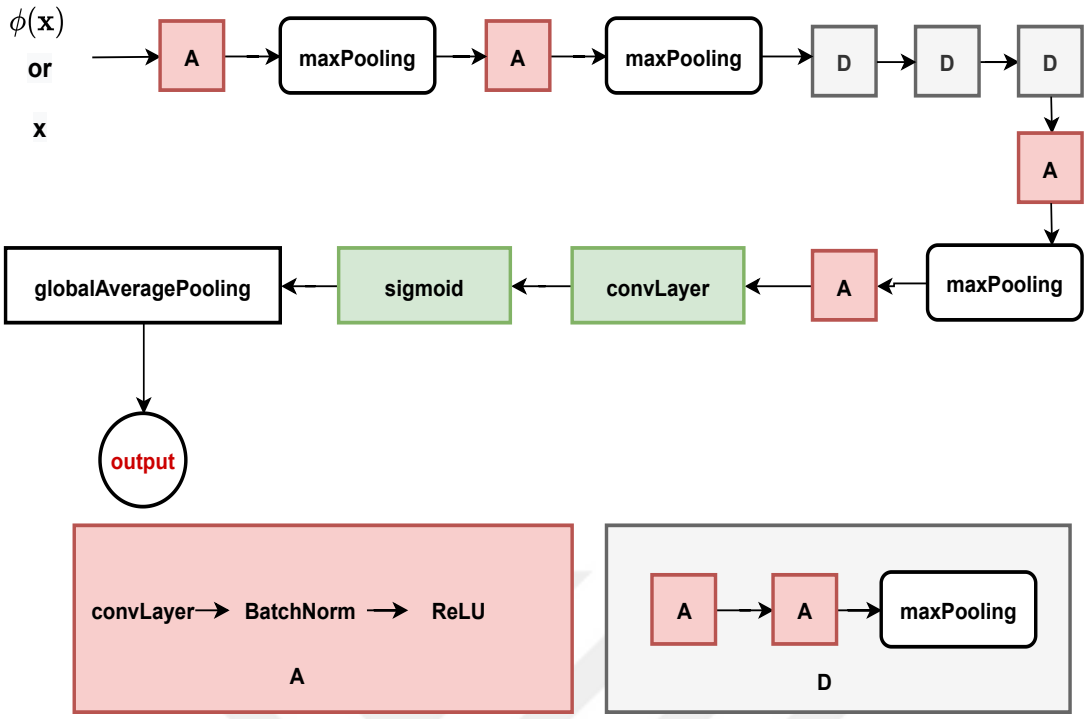


Figure 3.2: AEDNet network architecture (modified from Shah et al. [51]) is composed of many convolution and pooling layers. The shapes of the convolution layer and pooling layers change according to input type; i.e. raw audio waveform ($\mathbf{x}$) or extracted/learned features ($\phi(\mathbf{x})$).

prediction probability of AEDNet for class c.

3.3 AER with Hand-crafted Representations

As hand-crafted features, we consider the widely used representations: Namely, spectrogram, mel spectrogram, log mel spectrogram, and MFCC. See Figure 3.3 for an illustration of these features for a sample audio signal.

3.3.1 Spectrogram - $\phi_{spec}(\mathbf{x})$

A spectrogram is the time-frequency representation of an audio signal ($\mathbf{x}$). The main steps of extracting a spectrogram representation are given in Figure 3.4. When calcu-

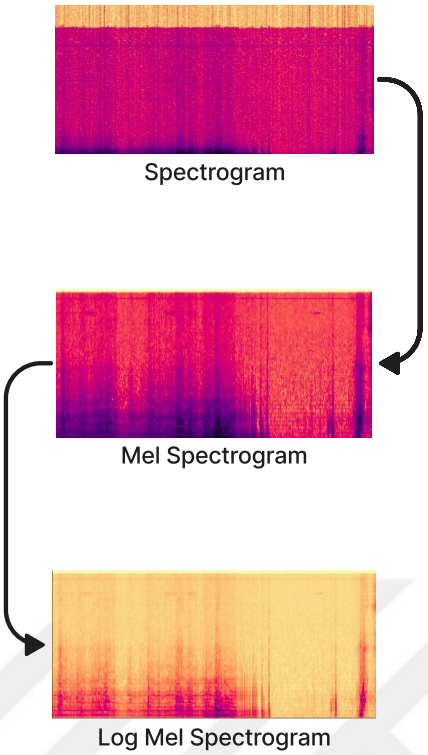


Figure 3.3: The diagram illustrates the commonly employed hand-crafted features. The arrows indicate the process of utilizing mel-scale filter banks to transform a spectrogram into a mel-spectrogram, followed by the computation of the logarithm of the mel-spectrogram to generate a log mel spectrogram.

lating a spectrogram $\phi_{spec}(\mathbf{x})$, since the frequencies in a signal change over time and in most cases the application of Fourier Transform to the entire signal makes it more difficult to recognize events with short durations, the signal $\mathbf{x}$ is first split into small segments $\{\mathbf{s}_i\}_i$. After slicing the signal into segments, in order to reduce the "spectral leakage", a windowing function $w(n)$ is applied to each segment $\mathbf{s}_i$ as follows:

$$\hat{\mathbf{s}}_i = \sum_{n=0}^{\mathcal{W}} s_{in} \times w(n), \tag{3.2}$$

where $\hat{\mathbf{s}}_i$ is a windowed segment of the signal, $\mathcal{W} = |\mathbf{s}_i|$ is the number of samples in the segment and $w(\cdot)$ can be e.g. the Hamming function:

$$w(n) = 0.54 - 0.45 \times \cos\left(\frac{2 \times \pi \times n}{\mathcal{W}}\right). \tag{3.3}$$

After the windowing function, a short-time Fourier Transform is applied to the windowed segments to obtain the Fourier coefficients, as $\mathcal{F}_i = \mathcal{FFT}(\hat{\mathbf{s}}_i)$. With these

steps, the spectrogram $\phi_{spec}(\mathbf{x})$ for an audio clip ($\mathbf{x}$) is calculated as a matrix:

$$\phi_{spec}(\mathbf{x}) = [\mathcal{F}_{jk}]_{jk}. \tag{3.4}$$

with $j \in [0, N-1]$ and $k \in [0, C-1]$. N, the row size, and C, the column size, are calculated as:

$$N = \frac{L \times F_s}{S}\,, \tag{3.5}$$

$$C = 1 + \frac{W_s}{2}\,, \tag{3.6}$$

where L is the audio clip duration in seconds; F_s is the sampling rate; W_s is the window size; and S is the stride, or the hop size, indicating the overlap among the windows.

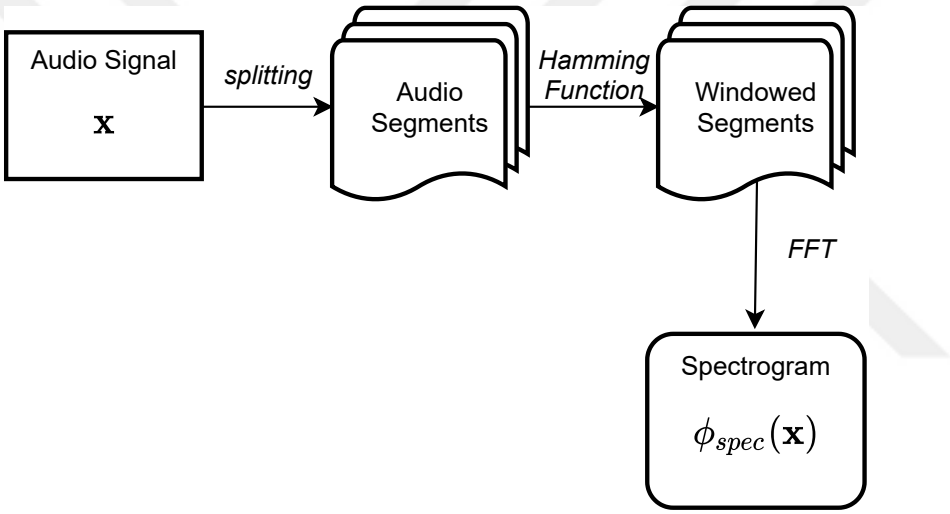


Figure 3.4: A block diagram listing the steps of spectrogram feature extraction for audio event detection.

3.3.2 Mel spectrogram features - $\phi_{mel}(\mathbf{x})$

A mel is a pair of sounds perceptually equidistant in pitch. The name mel comes from the word melody and indicates that the scale is based on pitch comparisons. Humans do not perceive frequencies on a linear scale: The difference between 500Hz and 1000Hz is easy to perceive, in contrast to the difference between 10000Hz and 10500Hz even though the magnitude of the difference is the same. The mel-scale [53]

aims to mimic the non-linear human ear perception of sound, being more discriminative at lower frequencies and less discriminative at higher frequencies. The mel scale is divided into units called mels and the reference frequency of 1000Hz, 40dB above a listener's threshold, is assigned to 1000 mels. Beginning at a frequency around 500Hz, the perceived interval between two pitches decreases as frequency increases. Above 500Hz, the number of mels between pitches is perceived as evenly spaced. Below 500Hz, the number of mels associated with a tone approximately corresponds to its frequency. Mel spectrograms are spectrograms that represent signals on the Mel scale:

$$\phi_{mel}(\mathbf{x})_i = mel(\phi_{spec}(\mathbf{x})_i), \tag{3.7}$$

where $mel(\cdot)$ transforms a signal from the Hertz scale into the Mel scale:

$$mel(f) = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right). \tag{3.8}$$

The mel spectrogram for an audio signal has size $N \times M$, where N is defined in Equation 3.5 and M, the number of columns, is the number of mels.

3.3.3 Log mel spectrogram features - $\phi_{logmel}(\mathbf{x})$

The log mel spectrogram is simply obtained by taking the logarithm of mel representation:

$$\phi_{logmel}(\mathbf{x})_i = \log(\phi_{mel}(\mathbf{x})_i). \tag{3.9}$$

Taking the logarithm compresses the dynamic range of values and makes frequency estimates less sensitive to slight variations in input. The size of the log mel spectrogram for an audio signal is the same as the mel spectrogram of that signal.

3.3.4 Mel Frequency Cepstral Coefficients - $\phi_{MFCC}(\mathbf{x})$

Cepstral coefficients obtained from the Mel spectrogram are referred to as Mel Frequency Cepstral Coefficients (MFCC). To get the MFCC, Discrete Cosine Transform (DCT) is applied on the log mel spectrogram:

$$\phi_{MFCC}(\mathbf{x}) = \sum_{k=1}^{K} \log D_k \times \cos\left[m \times \left(k - \frac{1}{2}\right) \times \frac{\pi}{K}\right], \tag{3.10}$$

where $m = 1, 2, ..., K$ is the number of coefficients and D_k is the mel power spectrum coefficients. The MFCC feature for an audio clip has a size (N, m), where N is as defined in Equation 3.5 and m is the number of coefficients. MFCC was very popular in traditional machine learning algorithms since they provide a very compressed representation with a few coefficients. Unlike numerous mel bands, MFCC can have as few dimensions as 13-20. Despite its compactness, MFCC can provide some advantages: DCT produces highly uncorrelated features, and deep neural networks can learn from these features much more easily than the correlated ones. In deep learning techniques, it is still unclear whether MFCCs perform better, as deep neural networks are less sensitive to highly correlated inputs than traditional machine learning algorithms.

3.4 AER with Raw Audio Waveform

In order to be able to better analyze AEDNet's performance with hand-crafted or learned features, we evaluate the model using the raw audio waveform, $\mathbf{x}$, as input. In this case, we use AEDNet with 1D convolutional layers with the same kernel size and the same number of filters and activation functions shown in Table 3.1. In the dataset, all audio clips are sampled with 44100 hertz with two channels. Moreover, the total duration of the audio clip is 10 seconds; therefore, the input $\mathbf{x}$ has the size of $(441000, 2)$.

3.5 AER with Learned Features from a Convolutional Autoencoder

Hand-crafted features are designed by humans and therefore may be sub-optimal. An alternative method is to learn to encode the raw audio signal using an autoencoder. To see if learned features have any advantages, in this work we compare the features learned with an autoencoder with the popular hand-crafted features we introduced in Section 3.3. We implement a convolutional autoencoder and use its 'bottleneck' features from the encoder. We do this by adapting the autoencoder architecture from [3], which was used to extract a learned representation of the raw audio signals, and modify the kernel sizes and the strides as needed - the architecture is listed in Table

3.2. The autoencoder is trained with an MSE loss defined as follows:

$$\mathcal{L}_{AE} = \frac{1}{W \times T} \sum_{i=1}^{W \times T} (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 , \tag{3.11}$$

where $\mathbf{x} \in \mathbb{R}^{W \times T}$ is the original input and $\hat{\mathbf{x}} \in \mathbb{R}^{W \times T}$ is the reconstructed input.

Table 3.2: The convolutional autoencoder we use for learning features.

	Encoder	**Decoder**
Layer1	conv1D Filters:16 Kernel Size:2 Stride:2	conv1DTrans Filters:16 Kernel Size:2 Stride:2
Layer2	maxPooling Size:8	conv1DTrans Filters:32 Kernel Size:2 Stride:1
Layer3	conv1D Filters:32 Kernel Size:2 Stride:2	upSampling Size:8
Layer4	maxPooling Size:8	conv1DTrans Filters:16 Kernel Size:16 Stride:2
Layer5	conv1D Filters:64 Kernel Size:2 Stride:1	upSampling Size:8
Layer6	conv1D Filters:128 Kernel Size:2 Stride:1	conv1DTrans Filters:1 Kernel Size:2 Stride:2
Layer7	Dense(128)	Cropping

3.6 AER with Features from a Pre-trained Network

Pre-trained networks for feature extraction have gained significant popularity in the field of audio event detection tasks[2, 13, 25, 62, 64], allowing researchers to utilize the representations to extract meaningful features from audio signals. These pre-trained models are trained on large-scale datasets and are capable of capturing high-level features in audio that can be used as inputs for downstream tasks like event detection, sound classification, speech recognition, and more. In the case of utilizing pre-trained networks for feature extraction, it is common practice to remove the final classification layer and extract features from the preceding layers. These features, known as bottleneck features, can then be employed as inputs to another model for additional processing and analysis. By leveraging pre-trained networks for feature extraction, researchers can benefit from the knowledge and representations learned by these models, saving substantial computational resources and achieving competitive performance on various audio-related tasks.

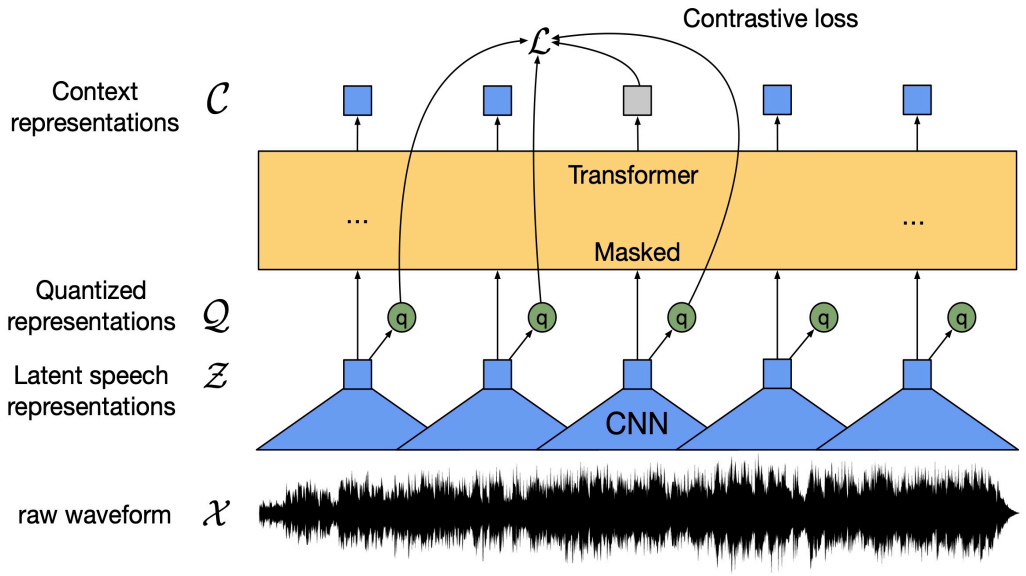


Figure 3.5: Wav2Vec2 model network architecture (from Baevski et al. [5]). The framework demonstrates the simultaneous acquisition of contextualized speech representations and a repository of discretized speech units. [Figure source: [5]]

In this thesis, we utilized the Wav2Vec2 model [5] as a pre-trained network for feature extraction. The model is primarily focused on addressing automatic speech recog-

nition tasks. The network structure is given in Figure 3.5 The fundamental concept behind this model is self-supervised learning. It employs a contrastive learning framework to pre-train the model using unlabeled speech data. With the help of this step, the model learns to extract meaningful representations from raw audio. By predicting future audio samples from the past, the model becomes proficient at capturing intricate patterns and phonetic information from the signals.

The core of the Wav2Vec2 architecture model is a stacked transformer encoder. The encoder in this model consists of several layers, each composed of multi-head self-attention mechanisms and position-wise-feed-forward neural networks. These layers allow the model to capture long-range dependencies and learn high-level contextual representations from the audio input. Following the pre-training phase, the Wav2Vec2 model undergoes a fine-tuning process using labeled data that includes corresponding transcriptions. During fine-tuning, the model's parameters are adjusted to optimize its performance on the specific task at hand, incorporating the provided transcriptions for supervision and improving its ability to accurately transcribe speech. This process aligns the learned acoustic features with textual data, enabling the model to associate acoustic features with linguistic units. By integrating all of the aforementioned steps, the Wav2Vec2 model becomes capable of capturing robust representations from raw audio signals, enabling it to effectively learn and process the underlying audio information.

The selection of the Wav2Vec2 model for the AER task in this thesis is motivated by its ability to extract high-quality features. The model has been pre-trained on a substantial amount of data, consisting of six different corpora. Moreover, it is a relatively recent study (from 2021). In addition, Wav2Vec2 is trained with raw audio signals and therefore, very suitable for our analyses with raw audio signals. We believe that the inclusion of Wav2Vec2 in our study is beneficial because it has been designed to capture general patterns and features that are typically present in audio signals. This inherent generalization ability of Wav2Vec2 enables it to extract high-level features that are particularly relevant for the AER task. By employing the pre-trained Wav2Vec2 model, we expect to leverage its capacity to learn meaningful representations from audio data, ultimately enhancing the effectiveness of our AER analyses.

CHAPTER 4

ANALYSIS OF HAND-CRAFTED AND LEARNED REPRESENTATIONS FOR AUDIO EVENT RECOGNITION[1]

In this chapter, we present the findings obtained through experimental evaluations and analysis. We aim to provide quantitative analysis to answer research questions that we follow in the Chapter 1

4.1 Evaluation Measure

We use mean average precision (mAP) as an evaluation measure following the literature [51]. The mAP metric is calculated by taking the mean of average precision (AP) over all classes in the dataset. For this, we first calculate AP for each event class using:

$$AP = \frac{1}{N} \sum_{n} (R_n - R_{n-1}) \times P_n \,, \tag{4.1}$$

where R_n and P_n are the precision and recall at the n^{th} threshold, and N is the number of such thresholds.

4.2 Utilized Dataset

We make use of the AudioSet [17] in our experiments. The original AudioSet dataset consists of approximately 2.1M samples with 527 classes and audio events organized by a hierarchy. Classes are organized with a maximum depth of 6 level labels. In Figure 4.1, the top two level labels are presented.

[1] Some text in this chapter is borrowed from our MTAP'22 paper [29].

Table 4.1: The AudioSet subset we used in our experiments. The number of samples is listed.

Class Name	**Training Samples**	**Test Samples**
Music	898	804
Speech	648	574
Animal	404	420
Vehicle	401	310
Musical Instrument	410	305
Domestic Animals, Pets	328	352
Guitar	309	286
Singing	229	214
Boat, Water Vehicle	210	201
Dog	260	252
Plucked String Instrument	295	232
Brass Instrument	211	200
Water	211	201
Car	264	227
Percussion	214	201
Wind, Wood Instrument	212	203
Total	**5504**	**4982**

Each instance is tagged with one or more labels. If one sample has a child class, it takes all parent classes as the class label. We use a balanced subset of AudioSet which included 16 classes. The selected classes and the number of samples that belong to these classes are given in Table 4.1. We choose samples randomly for each class, and the classes are constructed in a balanced way.



Figure 4.1: Top two main classes of 527 audio classes that are collected from 2,084,320 YouTube videos. All audio classes may have multiple classes. The labels are given by human annotators. To nominate segments for annotation, YouTube metadata and content-based search is used. [Figure Source: [17]]

4.3 Experimental Results

In this section, we analyze the methods following the research questions listed in Chapter 1.

4.3.1 Analysis 1: Analyzing Features and their Hyper-parameters (RQ1, RQ2)

We now analyze different window sizes and hop sizes to see the effect of different parameters for the hand-crafted features that are commonly used in the literature: spectrogram, mel spectrogram, log mel spectrogram and MFCC. We also consider different numbers of coefficients for the log mel spectrogram and MFCC feature. We evaluate the best parameters based on mAP, time complexity, GPU usage, and

memory usage. We train AEDNet for each method and setting.

4.3.1.1 Analyzing the effect of window and hop sizes

We first analyze the effect of window and hop sizes for the hand-crafted features by fixing the number of mel bands to 128, providing the best performance (see the next experiment). As shown in Figure 4.2, the methods perform best with different window and hop sizes. Namely, the best mAP is achieved with the parameters window size as 2048 and the hop size as 1024 for spectrogram features; 1024 and 256 for the mel spectrogram; and 1024 and 512 for log mel spectrogram.

4.3.1.2 Analyzing the effect of the number of mels

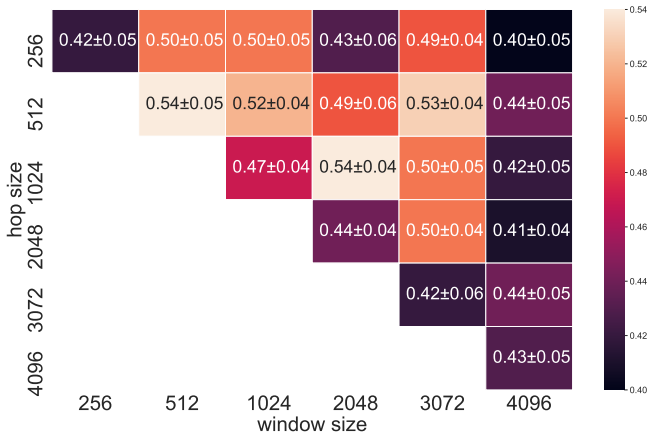
We analyze the effect of the number of mels. As the log mel spectrogram provided the best results in the previous experiment, we only focus on the log mel spectrogram with a window size of 1024 and a hop size of 512. The results in Figure 4.3 suggest that 128 mels is indeed the best setting, although the difference from other mels is not significant. Therefore, if resource efficiency has a higher priority, a lower number of mel bands may be preferred with a slight decrease in performance.

4.3.1.3 Analyzing the effect of the number of coefficients in MFCC features

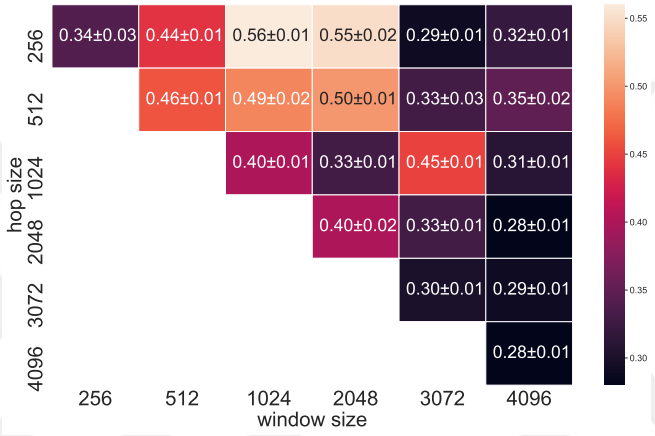
The number of coefficients in MFCC features is generally very small, which makes it difficult to utilize a deep AEDNet that is comparable in depth to the AEDNet with other hand-designed features. We designed two different coefficient settings: 40 coefficients, yielding an mAP of 0.60 and 20 coefficients, yielding an mAP of 0.56.

4.3.1.4 Comparison among Hand-crafted Features

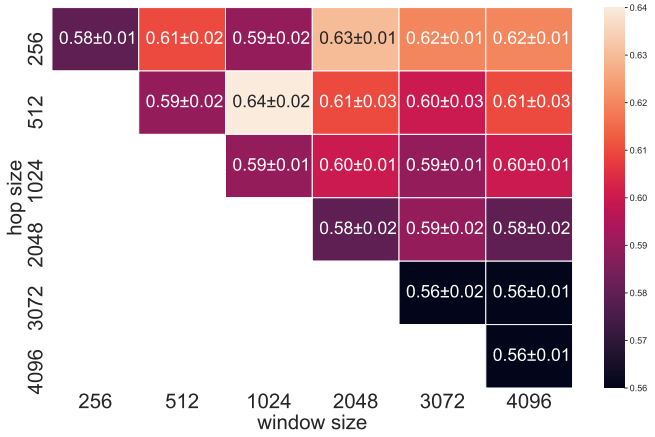
We now compare the best settings of all representations in Figure 4.4. We see that log mel spectrogram consistently outperforms the alternatives across different window & hop size combinations. In contrast, mel spectrogram features generally perform



(a) spectrogram (ϕ_{spec})



(b) mel spectrogram features (ϕ_{mel})



(c) log mel spectrogram (ϕ_{logmel})

Figure 4.2: The effect of window and hop size on spectrogram-based features.

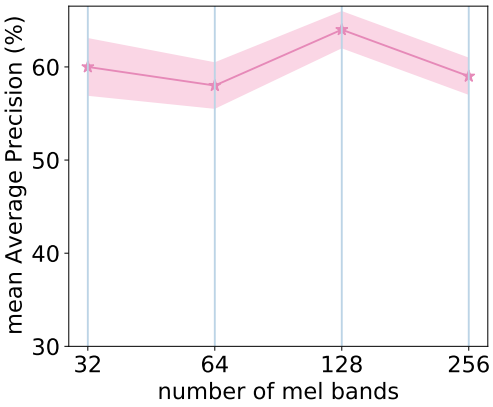


Figure 4.3: The effect of the number of mels for the log mel spectrogram feature (with 1024 as the window size and 512 as the hop size).

the worst, except for certain window & hop sizes (e.g. $\mathcal{W}$=1024, $\mathcal{S}$=256). In addition, MFCC coefficients provide better mAP than both the mel spectrogram and spectrogram but are still lower than the log mel spectrogram. The individual class performances for the log mel spectrogram features are given in Table 4.2, where we see that some classes (e.g., 'Music') are easier to recognize than others (e.g., 'Boat, Water Vehicle').

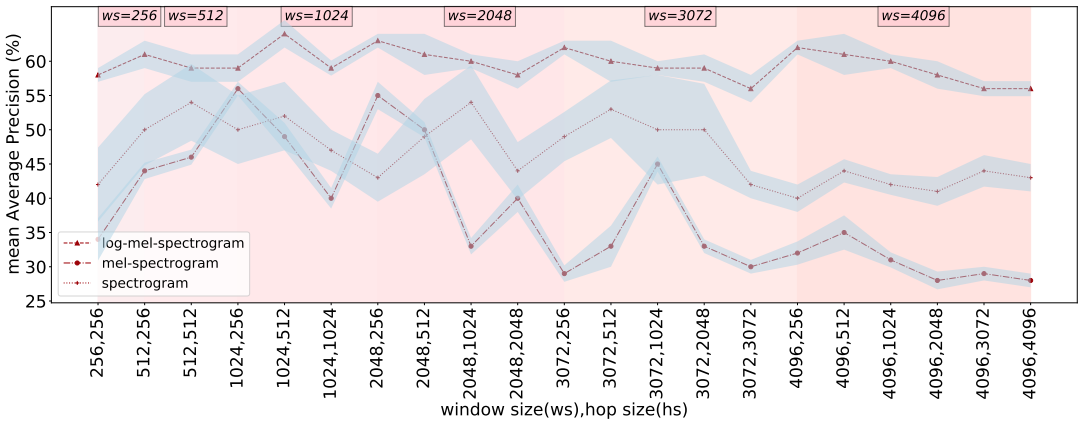


Figure 4.4: Effects of different window and hop size options in log mel spectrogram, mel spectrogram, and spectrogram features.

Table 4.2: Individual class performances for log mel spectrogram features.

Class Name	**mean Average Precision (mAP)**	**Standard Deviation**
Music	80 %	±0.02
Speech	60 %	±0.02
Animal	77 %	±0.02
Vehicle	68 %	±0.04
Musical Instrument	52 %	±0.02
Domestic Animals, Pets	75 %	±0.03
Guitar	71 %	±0.01
Singing	53 %	±0.03
Boat, Water Vehicle	49 %	±0.04
Dog	66 %	±0.02
Plucked String Instrument	58 %	±0.01
Brass Instrument	75 %	±0.03
Water	62 %	±0.03
Car	56 %	±0.03
Percussion	67 %	±0.03
Wind, Wood Instrument	68 %	±0.02
Average	**64%**	±0.01

4.3.2 Analysis 2: Learning-based Representation (RQ2)

4.3.2.1 Tuning the Autoencoder Hyper-parameters

We implement a convolutional autoencoder to extract learned representation from the raw audio waveform. While designing the architecture, we experiment with different kernel sizes, and the stride values, as shown in Table 4.3. We select these two parameters to show the effectiveness of the dimensions in the size of the feature. We see from the table that the best results are obtained using 2 as the kernel size in all

convolutional layers.

Table 4.3: The effect of kernel sizes and strides on convolutional autoencoder. See Table 3.2 for the layer numbering.'*' denotes cases where training diverged.

		Kernel Size Values in Layers		
		Layer1 = 8 Layer3 = 32 Layer5 = 16 Layer6 = 8	Layer1 = 4 Layer3 = 4 Layer5 = 4 Layer6 = 4	Layer1 = 2 Layer3 = 2 Layer5 = 2 Layer6 = 2
Stride Values in Layers	Layer1 = 2 Layer3 = 2 Layer5 = 1 Layer6 = 1	0.31 $\pm$ 0.07	0.33 $\pm$0.08	**0.38** $\pm$0.07
	Layer1 = 2 Layer3 = 2 Layer5 = 2 Layer6 = 1	0.33$\pm$ 0.07	*	0.31 $\pm$ 0.04
	Layer1 = 2 Layer3 = 2 Layer5 = 2 Layer6 = 2	0.27 $\pm$ 0.06	*	0.27 $\pm$ 0.03

4.3.2.2 Analyzing the effect of normalization

It is known that normalization can significantly affect performance in deep learning. For this reason, we analyze the different normalization techniques such as L1, L2, MinMax, and instance normalization. Since other normalization methods did not yield promising results, we only continue analyzing instance normalization, L_∞:

$$X_{scaled} = \frac{X}{\max_i |X_i|}, \tag{4.2}$$

where the X_i is the raw audio signal of the i^{th} window.

As shown in Figure 4.5, while the performances of hand-crafted features are better without normalization, the learned representation performs better with the normalized

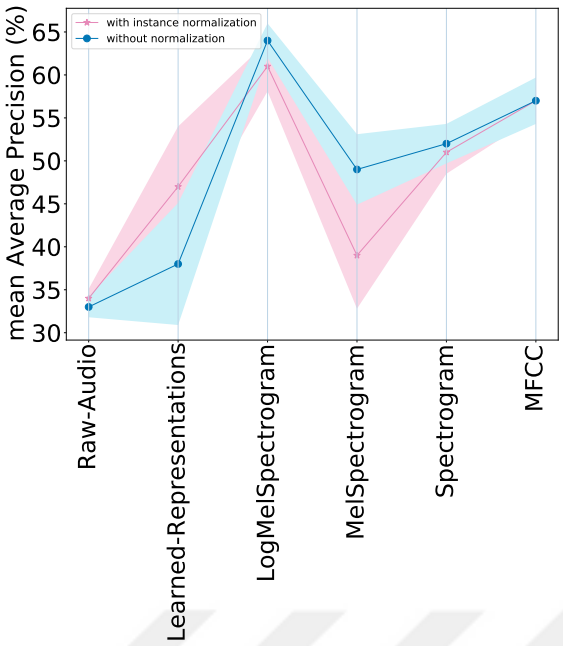


Figure 4.5: Comparison of the AEDNet performance for learned representation, hand-crafted features, and raw audio waveform with and without normalization.

signals. In particular, the performances of the raw audio waveform and the MFCC features are nearly the same with and without normalization.

4.3.2.3 Analyzing the learned representations from a pre-trained network

We conducted an analysis that utilizes the Wav2Vec2 model [5] for feature extraction. The general steps are given in Figure 4.6. To be able to use the Wav2Vec2 model, we needed to resample the audio clips to 16kHz while using this pre-trained network.

The results obtained from these experiments demonstrate that when we feed the learned representations obtained as the output of this model into AEDnet with parameter tuning, we achieve a mAP value of 56.

4.3.3 Analysis 3: Comparing All Representations (RQ3)

We now compare all the methods for different aspects in order to see their effects.

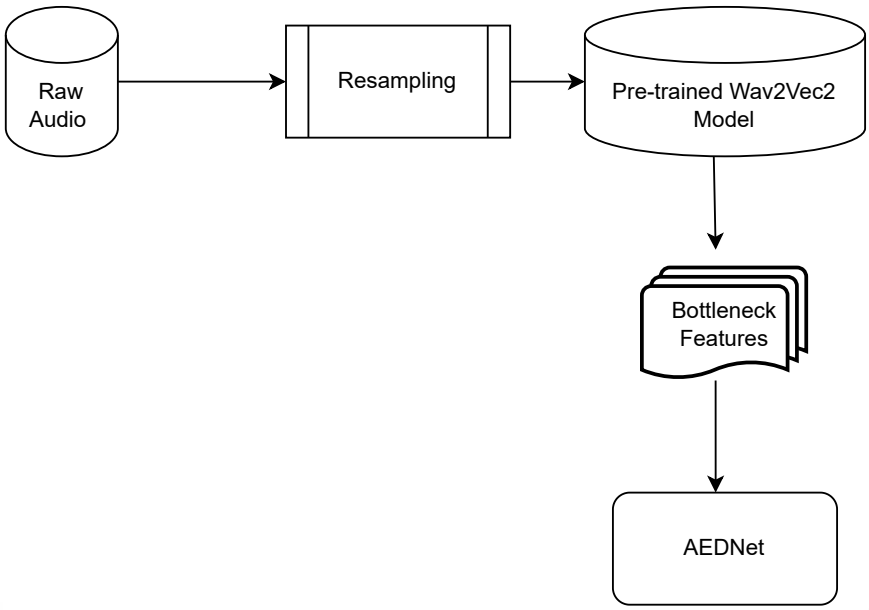


Figure 4.6: In order to analyze the effect of pre-trained network outputs on learned representations, we used the Wav2Vec2 model [5].

4.3.3.1 Overall Comparison

Table 4.4: A comparison of the best settings of hand-crafted features. Average signal preprocessing time refers to the average time in milliseconds for processing a signal i.e. reading the raw signal, extracting the feature from the signal, and appending it to the data frame before feeding it to the architecture. *The input audio signal is processed without windows.

Method	$\mathcal{W}$, S	**mAP**	**Feature Size**	**Number of parameters**	**Avg. preprocessing time of a signal**
ϕ_{spec}	2048, 1024	52%	221,616 (432X513)	4,471,568	30.6 ms
ϕ_{logmel}	1024, 512	**64%**	110,592 (864X128)	4,471,568	36.3 ms
ϕ_{mel}	1024, 256	56%	110,592 (864X128)	4,471,568	36.1 ms
ϕ_{MFCC}	1024, 512	60%	**34,560 (864X40)**	4,471,568	50.7 ms
ϕ_{raw}	*	33%	882,000 (441.000X2)	**1,859,248**	**13.0 ms**
$\phi_{learned_{pre-trained}}$	*	56%	510,916 (499X1024)	4,471,568	76.0 ms
$\phi_{learned_{CAE}}$	*	38%	220,416 (1722X128)	4,471,568	28.9 ms

In Table 4.4, we contrast the best settings of the methods in terms of mAP and several resource-related aspects as these can be important in practice. We see that log

mel spectrogram provides the best results among hand-crafted, learned (both for CAE and Wav2Vec2), and raw waveform representations. Moreover, we see that the methods significantly provide different levels of resource use and therefore, comparing the methods in terms of these aspects is important. For example, we notice that MFCC provides 60% mAP with only $\sim$1/3 of the feature size of log mel spectrogram and $\sim$1/30 of the raw waveform. Moreover, the number of parameters required for learning from all the representations is the same (except for the raw waveform) because of the convolutional & pooling based design of our AEDNet.

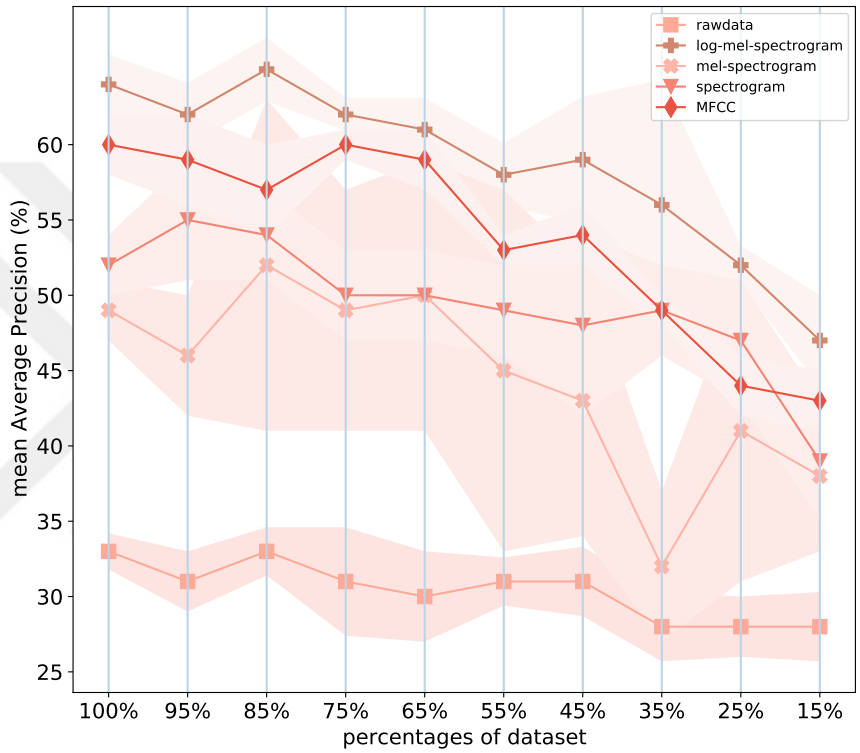


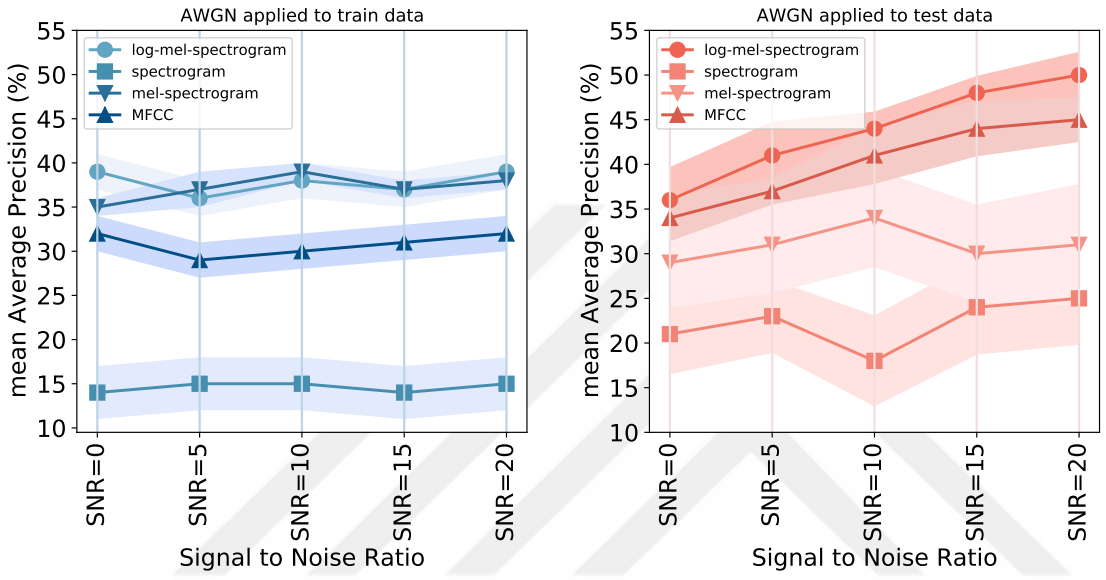
Figure 4.7: To see the effect of features on size, different methods of feature extraction on datasets of different sizes are performed. The values on the y-axis represent the ratios taken from the original dataset.

4.3.3.2 Robustness against the amount of training samples

In this part, we analyze the effect of the quantity of the dataset. We do this by reducing the size of our dataset by different ratios. As shown in Figure 4.7, we see that reducing the size of the dataset significantly degrades the performance of all methods. The

results suggest that selecting the feature extraction method based on the size of the dataset is an essential step for the subsequent steps. In the same architecture, the dataset prepared by taking 25% of the original dataset gives better results with the spectrogram features. However, the log mel spectrogram still gives higher results.

4.3.3.3 Robustness against the noise (RQ3)



(a) added noise to the train data

(b) added noise to the test data

Figure 4.8: Additive white noise is applied to the dataset to see the robustness of features.

In this part, we add additive white Gaussian noise (AWGN) to the training data to see the effect of the noise on different features. We add noise with five different levels of signal-to-noise ratio (SNR), which can be expressed as:

$$SNR = 10 \times \log\left(\frac{RMS^2_{signal}}{RMS^2_{noise}}\right), \tag{4.3}$$

where RMS is the root mean squared value of the signal and the noise. AWGN contains all frequency components.

We perform two experiments (Figure 4.8): (i) Both training and test sets are corrupted with noise, which assesses the suitability of all methods if we are to make them more

robust against the noise by adding noise at the training stage. (ii) To assess the robustness of the models against unforeseen noise conditions, only the test set is subjected to the addition of noise. This allows for the evaluation of the model's performance and its ability to handle noise types that were not included in the training data. The results in Figure 4.8 suggest that all methods may very well benefit from noise in the training data and perform well if such noise is also present in the test data. However, Figure 4.8(b) shows that all methods are affected by noise that is only supplied to test samples, although mel spectrogram and spectrogram provide more robust behavior.

CHAPTER 5

MULTI MODAL FUSION FOR AUDIO EVENT RECOGNITION

Multi-modal fusion strategies in audio features refer to techniques used to combine or integrate multiple audio features or representations into a unified representation. By fusing information from different sources, the resulting representation can provide a more comprehensive and robust understanding of the underlying data.

There are various fusion strategies commonly used in audio signal processing. Some of the popular techniques include early-fusion, mid-fusion, or late-fusion which can be depicted in Figure 5.1. Most common fusion strategies use methods like concatenation and decision-level or multi-modal networks. Cross-modal attention methods have also been increasing in the fusion in recent studies [37, 64, 60]. The comparison of the fusion studies in terms of the schemes and methods they use in audio domain is also is given in Table 5.1. Cross-modal attention presents a novel and intuitive fusion technique, involving the utilization of attention masks derived from one modal, to highlight the extracted features in another modal. Cross-modal attention is different from self-attention since in the self-attention, attention masks from one modal are employed to emphasize its own features, cross-attention is a distinct approach that utilizes attention masks from one modality to highlight the extracted features in another modality.

The choice of fusion strategy depends on the specific task, available audio features, computational requirements, and desired performance. In this study, we focused on early and late fusion techniques.

Spectrogram features were excluded from the fusion phase due to their comparatively inferior performance when compared to other hand-crafted features like log

Table 5.1: A comparison of fusion studies in terms of the schemes and methods they use for audio signal domain

Paper	Ref.	Fusion Scheme Type	Fusion Method
McLoughlin et al., 2020	[40]	Early	Concatenation
Brousmiche et al., 2021	[7]	Mid	Sef-Attention
Cai et al., 2021	[8]	Early	Cross-Modal Attention
Nagrani, 2021	[44]	Early, Mid or Late	Self, Cross Modal Attention
Xu et al., 2021	[64]	Mid	Transfer Learning
Jin et al., 2022	[22]	Early	Self-Attention
Jo et al., 2022	[23]	Early	Concetanation
Lee et al., 2022	[32]	Early	Cross-Modal Attention
Tan, 2022	[56]	Mid	Cross-Modal Attention
Xiong et al., 2022	[63]	Early	Cross-Modal Attention

mel spectrogram and MFCC. Furthermore, incorporating spectrogram features in the fusion phase would lead to increased computational complexity due to their large dimensions. Hence, they were not included in the fusion process.

5.1 Implementation Details of Fusion Schemes

In this section, we describe the early and late fusion schemes employed in this study. Early fusion involves merging the feature representations from different modalities or sources at an early stage of processing before any further analysis or modeling takes place. On the other hand, late fusion techniques involve combining the outputs or decisions of multiple models or systems at a later stage, after individual processing and analysis have been conducted.

The combination of different feature representations can be achieved through the use of basic statistics or concatenation techniques. While these methods are simple to

implement, they may not always be the most effective choice. In recent years, researchers have been developing new fusion techniques that utilize attention-based mechanisms [22, 23, 32, 40, 44, 56]. These fusion methods provide a uniform and generic structure and allow for the fusion of features at various scales, such as bottleneck features that can better represent raw input.

5.1.1 Early Fusion Scheme

As described in [52], early fusion is a technique that combines uni-modal features before learning concepts. Early fusion creates a new feature representation from the start. In this thesis, we extract hand-crafted features and combine them through concatenation or by using the attention mechanism, feeding the resulting feature representation into AEDNet in order to obtain new results.

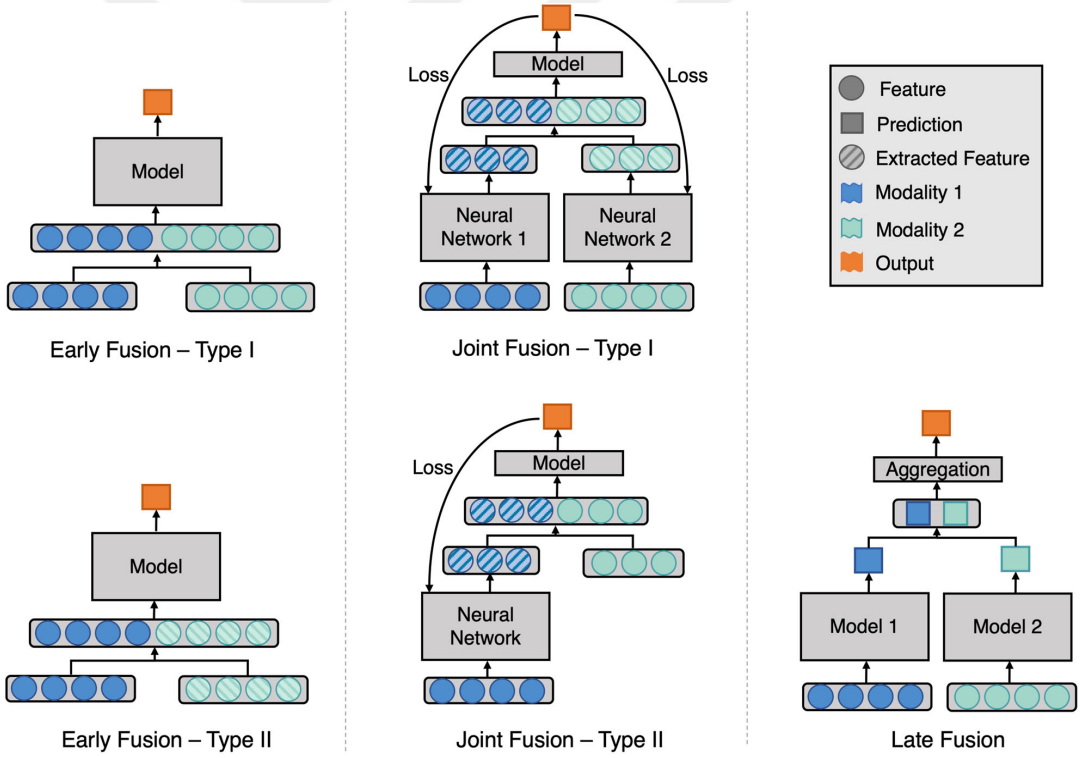


Figure 5.1: Three main different fusion strategies, namely early-fusion, joint (mid-fusion), and late-fusion. Early- fusion concatenates raw input or extracted features. Mid-fusion joins features at the input level, but the loss is propagated. Late-fusion aggregates predictions. [Figure Source: [21]]

5.1.1.1 Concatenation-based Early Fusion

The general scheme for concatenation-based early fusion on the features is given in Figure 5.2.

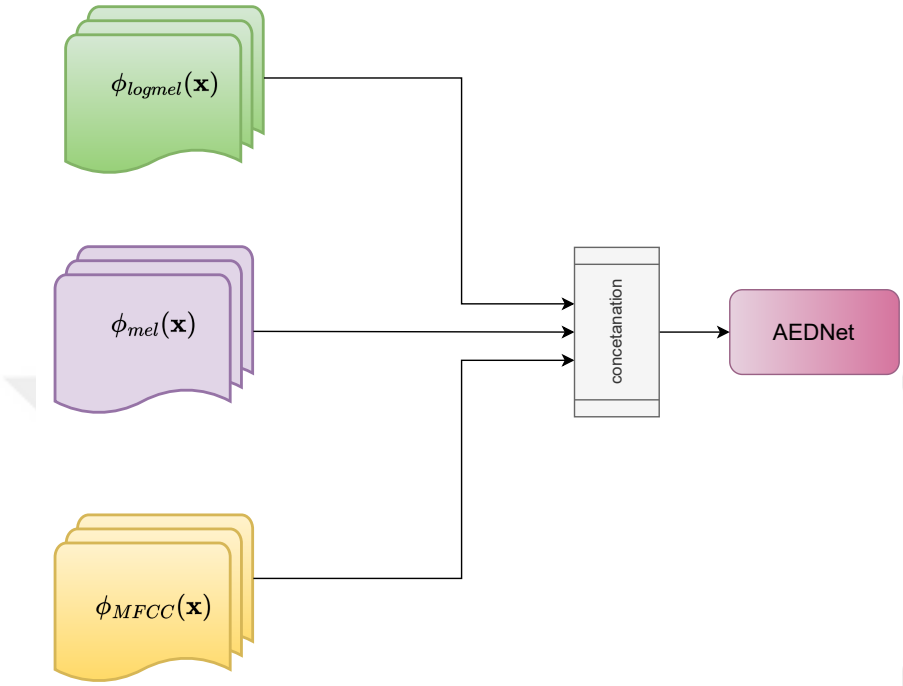


Figure 5.2: The general process for early fusion using concatenation is shown in the diagram. Hand-crafted features are extracted and then combined vertically through concatenation. Any combination of features can be used in this process. Once the concatenated features are obtained, they are fed into AEDNet as a new feature representation.

In the initial experiment, we focused on the concatenation of the log mel spectrogram and MFCC features. These two features were selected based on their superior performance in our main study, where we fine-tuned their hyperparameters. Once the features were extracted, we vertically combined them.

5.1.1.2 Attention-based Early Fusion

We propose a novel approach for combining features using an attention mechanism in this part. The module accepts hand-crafted features as input and generates attention-

based features as output. These attention-based features are then fed into AEDNet for evaluation as a new feature representation. This method of combining features using an attention mechanism is referred to as attention-based early fusion. The general steps are given in Figure 5.3.

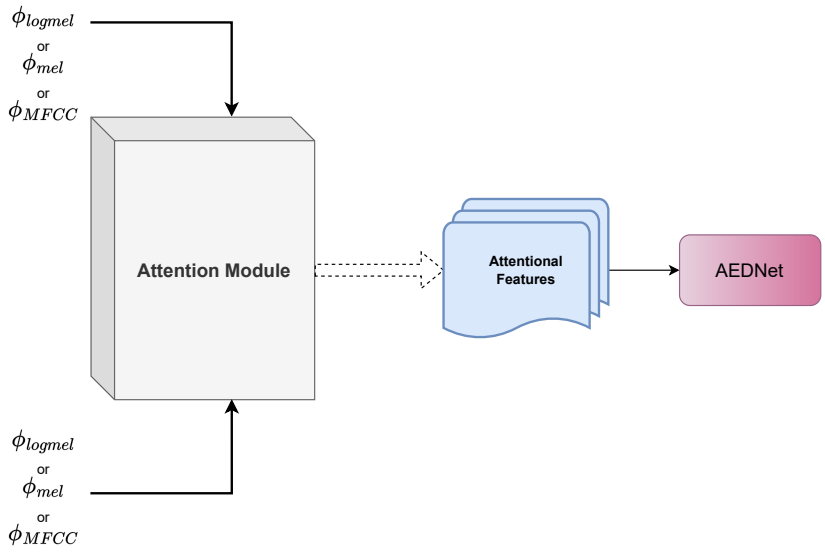


Figure 5.3: The general process for attention-based early fusion is shown in the diagram. The inputs for the attention module can be any combination of hand-crafted features, as long as they have the same dimensions.

We used the implementation from [11] in our thesis but modified the dynamic temporal attention module with a multi-layer implementation to a single layer. The attention-based features were then extracted from the attention module. The details of the attention module are shown in Figure 5.4.

This attention module is used in [11] and the goal was to detect scene changes between two images, but we used this module with some adaptations to extract attention-based features from hand-crafted audio features. Our attention module consists of a single layer, which takes two same-sized feature vectors from two channels as input and produces an attention-based feature representation as output. After obtaining attention-based features from the module, we input them into AEDNet and evaluate the new results.

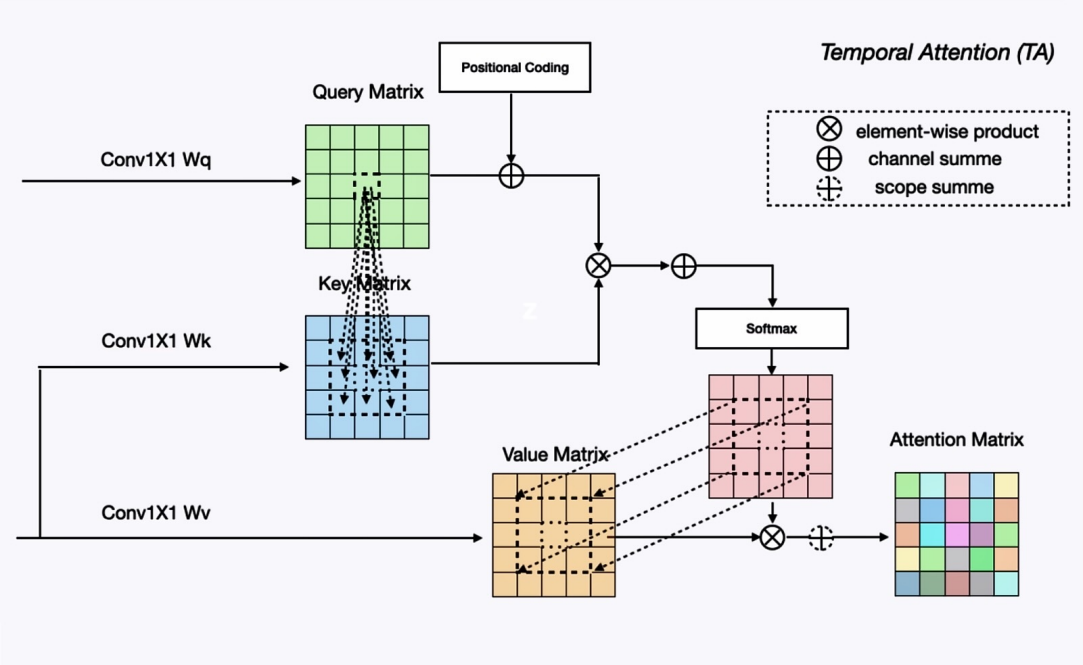


Figure 5.4: The Temporal Attention (TA) mechanism [11]. The query matrix is used to calculate the covariance between each sample in the feature at the first channel and the related samples in the dependency scope from the feature at the second channel by operating on the key matrix. [Figure Source: [11]]

5.1.2 Late Fusion Scheme

According to the definition in [52], late fusion involves reducing uni-modal features to separately learned concept scores and then integrating these scores to learn concepts. Late fusion emphasizes the individual strengths of different modalities. Instead of combining features, it merges uni-modal concept detection scores into a multi-modal semantic representation. Thus far, all of the work has been conducted in Section 5.1.1 using early fusion techniques. In Section 5.1.2 of the thesis, we attempted to implement late fusion by trying out concatenation-based and attention-based methods for fusing features.

5.1.2.1 Concatenation-based Late Fusion

In this technique, we fed the log mel spectrogram and mel spectrogram features as input to two different AEDNet, where we dropped the last 2 layers. We concatenated the outputs obtained from each AEDNet. This concatenated feature vector is then fed into a classifier for the desired task. This technique allows for the combination of different information sources, leveraging their complementary strengths and we expect to improve overall performance. The general steps are depicted in Figure 5.5.

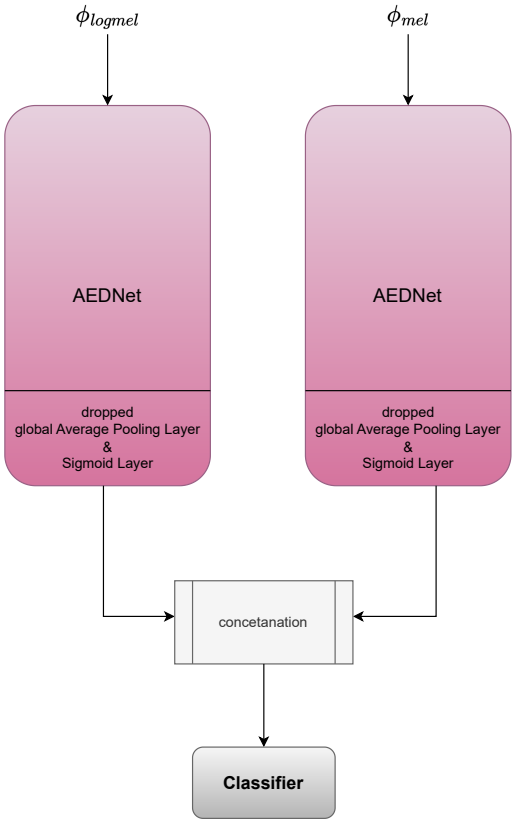


Figure 5.5: The diagram shows the design of the concatenation-based late fusion. Hand-crafted features are input into AEDNet without the last two layers of the original model, and the outputs are concatenated. Finally, a classifier produces a final output

5.1.2.2 Attention-based Late Fusion

In the attention-based late fusion method, we use log mel spectrogram and mel spectrogram as the features. These features are input into AEDNet with the last two layers removed, and the outputs are then processed by the multi-head attention module as separate channels. The module performs the following steps:

1. First, the inputs are processed through self-attention, where each position attends to other positions in the sequence. The attention weights are calculated based on the inputs.

2. The attention weights are scaled by the square root of the feature dimension and then masked. Masking is useful to prevent attending future positions.

3. The attention outputs are concatenated across the heads and processed through a linear layer to obtain the final output.

The attention-based features are generated in the multi-head attention module and passed to a basic classifier to evaluate the results. The general steps are shown in Figure 5.6.

5.2 Experimental Results of Fusion Schemes

In this section, we present the experimental results of the fusion schemes employed in our study. We evaluate the performance of early fusion and late fusion techniques.

5.2.1 Experiment 1: Early Fusion Schemes

We first analyze the results obtained from early fusion techniques. Our experiments demonstrate that early fusion consistently outperformed individual hand-crafted features. The concatenation-based early fusion technique, specifically combining log mel spectrogram and MFCC features, yielded a remarkable improvement, achieving an average precision (AP) score of 67%. This surpassed the performance of the log

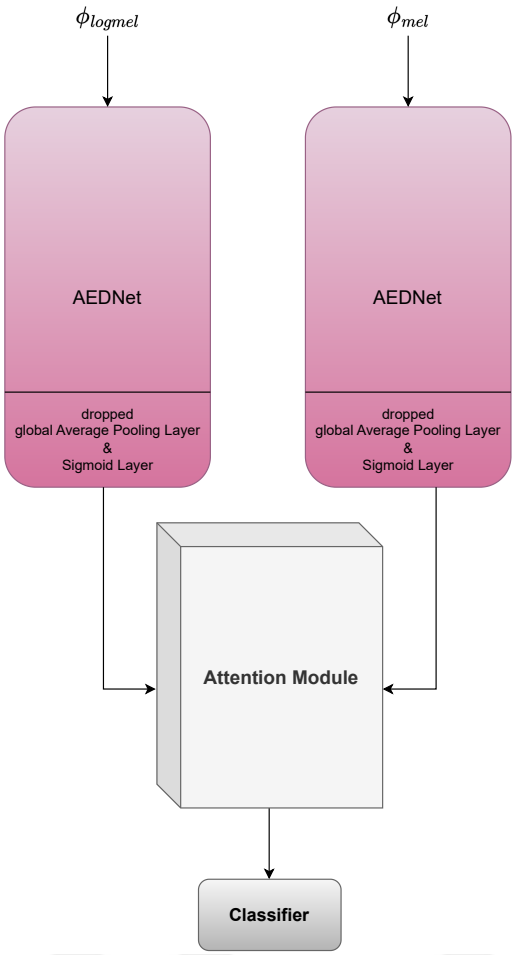


Figure 5.6: The diagram shows the design of the cross-modal attention module. Hand-crafted features are input into AEDNet without the last two layers of the original model, and the outputs are then processed by the attention module. Finally, a classifier produces an output

mel spectrogram feature alone. In the early fusion scheme, we used two different strategies: concatenation-based early fusion and attention-based early fusion.

The results shown in Table 5.2 indicate that early fusion consistently outperformed individual hand-crafted feature representations. By concatenating the log mel spectrogram and MFCC features, we were able to achieve a 67% mAP, surpassing the score obtained with the log mel spectrogram alone. However, when we attempted to include the mel spectrogram as a third feature alongside this pair, its performance was not as promising. Despite the similar mAP scores, we ultimately chose the concatenation of log mel spectrogram and MFCC features due to their smaller feature size

and lower time and memory complexity

Table 5.2: A comparison of concatenation-based early fusion of hand-crafted features and their baseline performance results on the AEDNet.

Features	**mAP**	**# of Parameters in Model**	**Fusion**
ϕ_{logmel}	64%	4,466,608	-
ϕ_{MFCC}	60%	4,466,608	-
ϕ_{mel}	56%	4,466,608	-
ϕ_{logmel} + ϕ_{MFCC}	**67%**	4,466,608	Concatenation-based Early
ϕ_{logmel} + ϕ_{MFCC} + ϕ_{mel}	66%	4,466,608	Concatenation-based Early

The attention-based early fusion results are presented in Table 5.3. Since the attention module requires feature vectors of the same size, we experimented with different combinations of hand-crafted features with equal dimensions. Based on the success of early fusion, we tried to log mel spectrogram, mel spectrogram, and MFCC features. As shown in the table, we obtained the best results with the log mel spectrogram and mel spectrogram attention map features, improving our baseline scores by 70% in terms of mAP.

Table 5.3: A comparison of attention-based early fusion of hand-crafted features and their baseline performance results on the AEDNet

Features	**mAP**	**# of Parameters in Model**	**Fusion**
ϕ_{logmel}	64%	4,466,608	-
ϕ_{MFCC}	60%	4,466,608	-
ϕ_{mel}	56%	4,466,608	-
ϕ_{logmel} + ϕ_{mel}	**70%**	4,471,600	Attention-based Early
ϕ_{logmel} + ϕ_{MFCC}	61%	4,471,600	Attention-based Early

5.2.2 Experiment 2: Late Fusion Schemes

The results of the different late fusion schemes are given in Table 5.4. The initial mAP score obtained with a concatenation-based late fusion approach was 37%. Despite tuning the hyperparameters and changing the classifier, we improve our result by 10% and raised the score to 47%. However, in the attention-based late fusion, we obtained an mAP score of 62% by tuning the parameters.

Table 5.4: A comparison of late fusion of hand-crafted features and their baseline performance results

Features	**mAP**	**# of Parameters in Network**	**Fusion**
ϕ_{logmel}	64%	4,466,608	-
ϕ_{MFCC}	60%	4,466,608	-
ϕ_{mel}	56%	4,466,608	-
ϕ_{logmel} + ϕ_{mel}	**62%**	13,149,872	Attention-based Late
ϕ_{logmel} + ϕ_{mel}	47%	9,255,136	Concatenation-based Late

According to the results of the late fusion scheme, we could not achieve better results than hand-crafted features. We obtained a worse result compared to the concatenation-based early fusion or log mel spectrogram on AEDNet. We believe that we encountered two important limitations in this study. Firstly, the small size of the training dataset, where we think a model trained with more data could have better avoided overfitting. Secondly, the number of parameters in the network, as we worked with nearly twice the number of parameters compared to AEDNet. For such a deep network, a larger dataset might have been required.

CHAPTER 6

CONCLUSION AND DISCUSSION[1]

In this thesis, we provided a comparative analysis of the different feature representation methods for the AED problem. To be specific, we were intrigued by the use of hand-crafted features for AED in the literature, despite the end-to-end use of deep learning methods in other domains, and we compared commonly used hand-crafted features with learned representations. Our analysis considered hand-crafted features such as spectrograms and MFCC, raw waveform as well as representation learned using an autoencoder and evaluated the methods on a subset of the AudioSet dataset in terms of their detection quality, resource efficiency, and robustness against various aspects.

6.1 Answers to the Research Questions (RQ1-RQ4)

Our analysis focused on seeking answers to the three research questions (RQ1-RQ3) listed in the Introduction section:

RQ1. *Are hand-crafted representations robust against hyper-parameters such as window size and hop size? Is there a specific hand-crafted representation method that is considered the best for AED?*

Our findings: Our analysis suggests that hand-crafted representations are very sensitive to window size and hop size. In fact, there can be a difference of 28AP between the best setting and the worst (e.g. for the mel spectrogram features in Figure 4.2). This gap is, however, less for spectrogram ($\sim$13AP) and

[1] Some text in this chapter is borrowed from our MTAP'22 paper [29].

log mel spectrogram (∼18 AP). As for the second part of the question: the log mel spectrogram offers the best detection performance (at 64AP versus 54AP of spectrogram and 56AP of mel spectrogram) while it is not the fastest among the hand-crafted features. Log mel spectrogram takes 36.3 ms whereas spectrogram does 30.6ms.

Comparison with respect to the literature: In the literature [20, 47, 48, 51], researchers generally set the hyper-parameters at specific values and extract the features based on these settings. In audio processing, generally, window sizes range from 20ms to 100ms with 50% overlap between consecutive windows for feature extraction and recommended settings are 40ms for window size which provides a good balance between having sufficient details and not being too high-dimensional [42]. However, our findings suggest 23ms to be the best setting for the window size for log mel spectrogram and mel spectrogram features, and 46ms or 12ms for spectrogram features. In contrast, our results confirm that a hop size corresponding to 50% overlap provides good results in general.

Overall, our results suggest that window size and hop size should be tuned for AED and the recommended settings may not provide the ideal setting. To the best of our knowledge, there are no studies that experiment with different window and hop sizes or coefficients in hand-crafted feature extraction methods.

RQ2. *Do learned representations perform better than hand-crafted representations?*

Our findings: Upon comparing hand-crafted features with features learned by a convolutional autoencoder, we observe that all hand-crafted feature representations surpass the learned representations in performance. This holds true even when training an AED network with the raw waveform as input. However, when utilizing a pre-trained network, we achieve improved results for certain hand-crafted features, specifically the spectrogram and mel spectrogram, yet the log mel spectrogram continues to exhibit the highest performance.

Comparison with respect to the literature: As we discussed in Introduction, the literature mainly use deep learning with hand-crafted feature representations.

Although existing studies do not compare hand-crafted features with learned representations, our results provide a justification for this trend.

RQ3. *Do learned and hand-crafted representations compare well under low data regimes or under noise?*

Our findings: According to our experiments, feature representations are affected by the amount of the dataset and performance dropped drastically with the reduction in dataset amount. For example, the performance of the log mel spectrogram dropped by 6AP, when 55% of the dataset is used, and by 11AP when 25% is available. However, despite being affected by the reduction in data, the log mel spectrogram still provides the best performance across different amounts of data. Our analysis for robustness to noise shows that, although log mel spectrogram features provide the best results, spectrogram, and mel spectrogram features are more robust against noise: Log mel spectrogram features admit $\sim$14AP drop under noise whereas mel spectrogram's performance is almost unaffected with $\sim$2AP change in performance. It is also worth mentioning that, when noise is added only to the test data without introducing additional noise to the training data, the obtained results in terms of mAP are not significantly low. This might be because the original dataset already contains noise, which allows the training data to perform well even when tested with noise-added data. This suggests that a model trained on real-world data containing inherent noise can achieve robust and accurate results.

Comparison with respect to the literature: We are not aware of any work that compared the robustness of methods against a reduction in data or increase in noise.

RQ4. *Are different audio features complementary? Could we increase the performance of AER by fusing different audio features?*

Our findings: Our experiments have shown that different fusion techniques can affect the performance of the system. By using early fusion methods, we were

able to improve upon our baseline and enhance the performance of the detection system. Even with the simple concatenation-based fusion, we saw an improvement in results to 67% mAP. The attention-based module further enhanced the system, leading to an mAP score of 70%. However, we did not obtain promising results with late fusion approaches. For example, we expect to achieve better results with the help of the concatenation-based late fusion technique, however, even trying different classifiers with tuned hyperparameters, we did not achieve better scores than our baseline

Comparison with respect to the literature: We are not aware of any work that uses any comprehensive analysis fusion schemes by using attention-based or concatenation-based.

6.2 Limitations and Future Work

Although the analysis we provide in this paper is very valuable, it is performed with a balanced subset of the AudioSet dataset in order to be able to perform in-depth experiments. An analysis with a larger dataset would be valuable and might reveal e.g. that deep representation learning requires significantly larger datasets than the ones we used.

Another important aspect is the deep model that was used for the comparative analysis. We chose a CNN-based model that could be considered to be representative of existing CNN-based solutions. However, it would be interesting to study how different types of architectures (e.g. networks using attention, transformers etc.) would work with different representations.

In this thesis, besides Wav2Vec2, alternative approaches can be explored for extracting features from audio signals. Two widely used state-of-the-art models, namely AST (Audio Spectrogram Transformer) [19] and BART (Bidirectional and Auto-Regressive Transformer) [34], can be considered as potential feature extractors. AST is a pre-trained network that leverages the transformer architecture. It takes spectrograms as input and trained with the AudioSet dataset. BART, initially developed for natural

language processing tasks, has shown promise in various domains, including audio processing. It can be utilized by using spectrogram-based inputs, similar to AST. However, note that AST and BART take a spectogram representation as input while Wav2Vec2 works with raw audio signals.

Another promising research direction is to study how the different representations can be fused in order to obtain better performance. In the study, we tried to use early and late fusion schemes, however, mid-level fusion can improve the performance. Also, the fusion mechanism can be dynamic, i.e., depending on the requirements of the task, different representations can be utilized. For example, if the input seems to be easy, the fastest representation can be chosen to obtain a reliable response in the shortest time possible.

The size of the dataset may have an impact on attention-based late fusion. As the architecture changes in terms of the number of parameters, the utilized dataset may be modified by adding new samples or classes, increasing the number of samples, and allowing the entire network to learn more effectively.

REFERENCES

[1] K. Ahmad and N. Conci. How deep features have improved event recognition in multimedia: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.*, 15(2), jun 2019.

[2] M. Al-Malla, A. Jafar, and N. Ghneim. Pre-trained cnns as feature-extraction modules for image captioning: An experimental study. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 21:1–16, 05 2022.

[3] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, 2016.

[4] E. Babaee, N. B. Anuar, A. W. A. Wahab, S. Shamshirband, and A. T. Chronopoulos. An overview of audio event detection methods from feature extraction to classification. *Applied Artificial Intelligence*, 31(9-10):661–714, 2017.

[5] A. Baevski, H. Zhou, A. Mohamed, and M. Auli. Wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[6] S. Becker, M. Ackermann, S. Lapuschkin, K. Müller, and W. Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, 2018.

[7] M. Brousmiche, J. Rouat, and S. Dupont. Multi-level attention fusion network for audio-visual event recognition. *CoRR*, abs/2106.06736, 2021.

[8] S. Cai, P. Li, E. Su, and L. Xie. Auditory attention detection via cross-modal attention. *Frontiers in Neuroscience*, 15, 2021.

[9] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. *CoRR*, 2017.

[10] E. Çakir and T. Virtanen. End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input. *CoRR*, 2018.

[11] S. Chen, K. Yang, and R. Stiefelhagen. Dr-tanet: Dynamic receptive temporal attention network for street scene change detection. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, page 502–509. IEEE Press, 2021.

[12] W. Dai, C. Dai, S. Qu, J. Li, and S. Das. Very deep convolutional neural networks for raw waveforms, 2016.

[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[14] H. Dinkel, Y. Qian, and K. P. Yu. A hybrid asr model approach on weakly labeled scene classification. Technical report, Shanghai Jiao Tong University, Computer Science Dept., Shanghai, China, 2018.

[15] C. Eutizi and F. Benedetto. On the performance improvements of deep learning methods for audio event detection and classification. In *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*, pages 141–145, 2021.

[16] E. Fonseca, D. Ortego, K. McGuinness, N. E. O'Connor, and X. Serra. Unsupervised contrastive learning of sound event representations, 2020.

[17] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled

dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

[18] T. Giannakopoulos, E. Spyrou, and S. Perantonis. Recognition of urban sound events using deep context-aware feature extractors and handcrafted features. *Int. Conf. on Artificial Intelligence Applications and Innovations*, 2019.

[19] Y. Gong, Y.-A. Chung, and J. Glass. AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, pages 571–575, 2021.

[20] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. W. Wilson. CNN architectures for large-scale audio classification. *CoRR*, 2016.

[21] S.-C. Huang, A. Pareek, S. Seyyedi, I. Banerjee, and M. Lungren. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *npj Digital Medicine*, 3, 12 2020.

[22] Y. Jin, M. Wang, L. Luo, D. Zhao, and Z. Liu. Polyphonic sound event detection using temporal-frequency attention and feature space attention. *Sensors*, 55:593–625, 09 2022.

[23] S.-H. Jo, C. Y. Jeong, and M. Kim. Sound event localization and detection using spatial feature fusion. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1849–1851, 2022.

[24] M. Kayser and V. Zhong. Denoising convolutional autoencoders for noisy speech recognition. Technical report, CS231 Standford Reports, March 2015.

[25] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito. A transformer-based audio captioning model with keyword estimation. pages 1977–1981, 10 2020.

[26] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition, 2020.

[27] S. Kothinti, G. Sell, S. Watanabe, and M. Elhilali. Integrated bottom-up and top-down inference for sound event detection. Technical report, Department

of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, USA, June 2019.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[29] S. E. Küçükbay, A. Yazıcı, and S. Kalkan. Hand-crafted versus learned representations for audio event detection. *Multimedia Tools Appl.*, 81(21):30911–30930, sep 2022.

[30] J.-Y. Kwak and Y.-J. Chung. Sound event detection using derivative features in deep neural networks. *Applied Sciences*, 10:4911, 07 2020.

[31] J. Lee, T. Kim, J. Park, and J. Nam. Raw waveform-based audio classification using sample-level cnn architectures. *31st Conf. on Neural Information Processing Systems (NIPS)*, 2017.

[32] S.-H. Lee, J.-W. Hwang, M.-H. Song, and H.-M. Park. A method based on dual cross-modal attention and parameter sharing for polyphonic sound event localization and detection. *Applied Sciences*, 12(10), 2022.

[33] S. Lefèvre and N. Vincent. A two level strategy for audio segmentation. *Digital Signal Processing*, 21(2):270 – 277, 2011.

[34] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. rahman Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*, 2019.

[35] J. Li, W. Dai, F. Metze, S. Qu, and S. Das. A comparison of deep learning methods for environmental sound. *CoRR*, 2017.

[36] H. Liu and S. Zhang. Noisy data elimination using mutual k-nearest neighbor for classification mining. *Journal of Systems and Software*, 85(5):1067 – 1074, 2012.

[37] Y. Liu, O. Yoshie, and H. Watanabe. Application of multi-modal fusion attention mechanism in semantic segmentation. In L. Wang, J. Gall, T.-J. Chin,

I. Sato, and R. Chellappa, editors, *Computer Vision – ACCV 2022*, pages 378–397, Cham, 2023. Springer Nature Switzerland.

[38] A. Maas, Q. Le, T. neil, O. Vinyals, P. Nguyen, and A. Ng. Recurrent neural networks for noise reduction in robust asr. *13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012*, 1, 2012.

[39] A. Maria and A. S. Jeyaseelan. Development of optimal feature selection and deep learning toward hungry stomach detection using audio signals. *Journal of Control, Automation and Electrical Systems*, 32, 08 2021.

[40] I. Mcloughlin, Z. Xie, Y. Song, H. Phan, and R. Palaniappan. Time–frequency feature fusion for noise robust audio event classification. *Circuits, Systems, and Signal Processing*, 39, 03 2020.

[41] A. Mesaros, T. Heittola, and T. Virtanen. Tut database for acoustic scene classification and sound event detection. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1128–1132, 2016.

[42] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley. Sound event detection: A tutorial. *IEEE Signal Processing Magazine*, 38(5):67–83, Sep 2021.

[43] G. Muhammad and M. Melhem. Pathological voice detection and binary classification using mpeg-7 audio features. *Biomedical Signal Processing and Control*, 11:1 – 9, 2014.

[44] A. Nagrani, S. Yang, A. Arnab, A. Jansen, C. Schmid, and C. Sun. Attention bottlenecks for multimodal fusion. *CoRR*, abs/2107.00135, 2021.

[45] S. Ntalampiras, I. Potamitis, and N. Fakotakis. On acoustic surveillance of hazardous situations. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 165–168, 2009.

[46] S. Ntalampiras, I. Potamitis, and N. Fakotakis. A portable system for robust acoustic detection of atypical situations. In *17th European Signal Processing Conference*, pages 1121–1125, 2009.

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

[48] K. J. Piczak. Recognizing bird species in audio recordings using deep convolutional neural networks. In *CLEF*, 2016.

[49] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[50] A. Saeed, D. Grangier, and N. Zeghidour. Contrastive learning of general-purpose audio representations, 2020.

[51] A. Shah, A. Kumar, A. G. Hauptmann, and B. Raj. A closer look at weak label learning for audio events. *arXiv preprint arXiv:1804.09288*, 2018.

[52] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, page 399–402, New York, NY, USA, 2005. Association for Computing Machinery.

[53] S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. *Journal of the Acoustical Society of America*, 8:185–190, 1937.

[54] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746, 2015.

[55] Y. Sun and S. Ghaffarzadegan. An ontology-aware framework for audio event classification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2020*, 2020.

[56] Q. Tan. Multimodal network with cross-modal attention for audio-visual event localization. In *Proceedings of the 3rd International Workshop on Human-Centric Multimedia Analysis*, HCMA '22, page 71–78, New York, NY, USA, 2022. Association for Computing Machinery.

[57] N. Turpault and R. Serizel. Training sound event detection on a heterogeneous dataset, 2020.

[58] A. Uncini. Audio signal processing by neural networks. *Neurocomputing*, 55:593–625, 10 2003.

[59] M. Vasilakis and Y. Stylianou. Spectral jitter modeling and estimation. *Biomedical Signal Processing and Control*, 4(3):183 – 193, 2009.

[60] Y. Wang, Y. Xie, J. Zeng, H. Wang, L. Fan, and Y. Song. Cross-modal fusion for multi-label image classification with attention mechanism. *Computers and Electrical Engineering*, 101:108002, 2022.

[61] Z. Wang, J. Casebeer, A. Clemmitt, E. Tzinis, and P. Smaragdis. Sound event detection with adaptive frequency selection. *ArXiv*, abs/2105.07596, 2021.

[62] H. Won, B. Kim, I.-Y. Kwak, and C. Lim. Using various pre-trained models for audio feature extraction in automated audio captioning. *Expert Systems with Applications*, page 120664, 2023.

[63] J. Xiong, P. Zhang, L. Xie, W. Huang, Y. Zha, and Y. Zhang. Audio-visual speech separation based on joint feature representation with cross-modal attention. *ArXiv*, abs/2203.02655, 2022.

[64] X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. Yu. Investigating local and global information for automated audio captioning with transfer learning. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 905–909, 2021.

[65] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. B. Jackson, and M. D. Plumbley. Fully deep neural networks incorporating unsupervised feature learning for audio tagging. *CoRR*, abs/1607.03681, 2016.

[66] Z. Zang, M. Yang, and L. Liu. An improved system for dcase 2019 challenge task4. Technical report, University of Electronic Science and Technology of China School of Information and Communication Engineering, June 2019.

[67] X. Zhuang, X. Zhou, T. S. Huang, and M. Hasegawa-Johnson. Feature analysis

and selection for acoustic event detection. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 17–20, 2008.