UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Emotion Based Music and Audio Understanding

Permalink

https://escholarship.org/uc/item/3765h8k8

Author

Koh, Eunjeong

Publication Date

2022

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA SAN DIEGO

Emotion Based Music and Audio Understanding

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Music

by

Eunjeong Koh

Committee in charge:

Professor Shlomo Dubnov, Chair
Professor Garrison W. Cottrell
Professor Virginia de Sa
Professor Miller Puckette
Professor Shahrokh Yadegari

2022

Copyright
Eunjeong Koh, 2022
All rights reserved.

The dissertation of Eunjeong Koh is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my family. Thank you for your love and support.

EPIGRAPH

Music comes to me

more readily than words.

—Ludwig van Beethoven

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Professor Shlomo Dubnov, for his guidance and support over the years throughout my PhD studies. Shlomo introduced me to machine learning research areas and he always shared his creative ideas for thinking about the future of Artificial Intelligence for Arts. During my studies, Shlomo supported my desire to explore interdisciplinary research fields. I appreciate his advice and I will miss our research discussion time that can result in this thesis and my future career. I would like to thank Professor Miller Puckette for his unconditional guidance and support. I remember listening to his lectures before joining UCSD and getting excited about audio synthesis and sound design to study them. I have learned some lessons from him in every research meeting. Additional thanks to Professor Shahrokh Yadegari who guided me with the work of my first computer music research paper. I really enjoyed learning and discussing our interface design and Calit2 exhibition work with him. I would like to also thank my other committee members, Professor Garrison Cottrell, and Professor Virginia de Sa. All their feedback and comments were essential and helpful in my research progress. In addition, I would like to thank my research collaborators, Fatemeh Saki, Yinyi Guo, Cheng-yu Hung, Erik Visser, and Dustin Wright for their supportive feedback through the process of publishing my publications.

I would like to express my sincere thank you to my family for their love and support. My parents, Youngmi Seo and Moonsang Ko always encourage me to challenge something new. In my childhood, they tried their best to raise their kids explore a new adventure, and travel abroad together for making us experience the wide world. I always appreciate their incredible support in my journey all the time. Thanks to my siblings, Hyunjeong, Heejeong, and Yeajin for their continued love and encouragement. I’ve never felt alone because we are together in this world. My deepest thanks to my husband, Jason Koh for his love, support, and research guidance. I have been lucky for having him as my spouse. Lastly, thanks to my daughter, Piper. She makes me realize how life is beautiful.

Thanks to Timothy Langlois at Adobe Research, who worked with me for audio projects at the computer vision team in Adobe Research. Thanks to Yinyi Guo at Qualcomm who shared his interests in various audio research projects. Thanks to Bob Coover at Apple Music for his support and research guidance. Thanks to John Shimmin at Amazon Music who guided me with patience during my first internship in the music technology field.

Thank you to my friends in San Diego who have shared some joy and difficulties throughout this process: Jean Choi, Min-ju Kim, Sanghyun Lee, Blenda Yu, Eunice Kim, Mingi Chung, Kyungeun Yang, Lauren Jones, Mari Kawamura, Nakul Tiruviluamala, Jacob Sundstrom, David Defilippo, Chris Donahue, Gabriel Zalles-ballivian, Andrés Gutiérrez Martínez, Shehzeen Hussain, Paarth Neekhara, Vignesh Gokul.

Chapter 3 contains materials found in the following paper. ”Incremental Learning Algorithm for Sound Event Detection”. Koh, Eunjeong, Saki, Fatemeh, Guo, Yinyi, Hung, Cheng-Yu, and Visser, Erik. IEEE International Conference on Multimedia and Expo, 2020. The dissertation/thesis author was the primary researcher and author of the paper.

Chapter 4 contains materials found in the following paper. ”Rethinking recurrent latent variable model for music composition”. Koh, Eunjeong, Dubnov, Shlomo, and Wright, Dustin. IEEE International Workshop on Multimedia Signal Processing, 2018. The dissertation/thesis author was the primary researcher and author of the paper.

Chapter 5 contains materials found in the following paper. ”Information Dynamics in Machine Generated Music”, Koh, Eunjeong and Dubnov, Shlomo. SoCal Machine Learning Symposium, 2019. The dissertation/thesis author was the primary researcher and author of the paper.

Chapter 6 contains materials found in the following paper. ”Mugeetion: Musical Interface Using Facial Gesture and Emotion”, Koh, Eunjeong and Yadegari, Shahrokh. International Computer Music Conference, 2018. The dissertation/thesis author was the primary researcher and author of the paper.

Chapter 7 contains materials found in the following papers. ”Using Deep Audio Embeddings for Music Emotion Recognition”, Koh, Eunjeong and Dubnov, Shlomo. Association for the Advancement of Artificial Intelligence Workshop On Affective Content Analysis, 2021. ”Understanding Affective Aspects of Music Using Deep Audio Embeddings”, Koh, Eunjeong and Dubnov, Shlomo. International Conference on Music Perception and Cognition, 2021. The dissertation/thesis author was the primary researcher and author of the paper.

Chapter 8 contains materials found in the following paper. ”The Role of Musical Structure in Shaping Listener’s Preference”, Koh, Eunjeong and Kim, Min-ju. Society for Music Perception and Cognition, 2017. The dissertation/thesis author was the primary researcher and author of the paper.

VITA

2012	B. S. in Computer Science and Engineering, Ewha Womans University, South Korea
2014	M. S. in Engineering, Seoul National University, South Korea
2022	Ph. D. in Music, University of California San Diego

PUBLICATIONS

Eunjeong Koh, Shlomo Dubnov. “Understanding Affective Aspects of Music Using Deep Audio Embeddings”, *International Conference on Music Perception and Cognition*, 2021.

Eunjeong Koh, Shlomo Dubnov. “Comparison and Analysis of Deep Audio Embeddings for Music Emotion Recognition”, *Association for the Advancement of Artificial Intelligence Workshop On Affective Content Analysis*, 2021.

Eunjeong Koh, Fatemeh Saki, Yinyi Guo, Cheng-Yu Hung, Erik Visser. “Incremental Learning Algorithm for Sound Event Detection”, *IEEE International Conference on Multimedia and Expo*, 2020.

Fatemeh Saki, Yinyi Guo, Cheng-Yu Hung, Lae-Hoon Kim, Manyu Deshpande, Sunkuk Moon, Eunjeong Koh, Erik Visser. “Open-set evolving acoustic scene classification system”, *Detection and Classification of Acoustic Scenes and Events*, 2019.

Eunjeong Koh, Shlomo Dubnov. ”Information Dynamics in Machine Generated Music”, *SoCal Machine Learning Symposium*, 2019.

Eunjeong Koh, Shahrokh Yadegari. “Mugeetion: Musical interface using facial gesture and emotion”, *International Computer Music Conference*, 2018.

Eunjeong Koh, Shlomo Dubnov, Dustin Wright. “Rethinking recurrent latent variable model for music composition”, *IEEE International Workshop on Multimedia Signal Processing*, 2018.

Eunjeong Koh, Min-ju Kim. “The Role of Musical Structure in Shaping Listener’s Preference”, *Society for Music Perception and Cognition*, 2017.

ABSTRACT OF THE DISSERTATION

Emotion Based Music and Audio Understanding

by

Eunjeong Koh

Doctor of Philosophy in Music

University of California San Diego, 2022

Professor Shlomo Dubnov, Chair

Machine learning is a methodology of data analysis that allows software to learn about data, identify patterns, and make predictions without human intervention. Using machine learning, researchers can automatically generate high-quality images and write a novel. Creating art with machine learning is a state-of-the-art technique, which can bridge several different research areas, such as computer science, cognitive science, psychology, and music. In this dissertation, the research objectives are (i) to find effective machine learning practices for music and audio understanding based on emotional context, which improves the knowledge of affective computing, (ii) to design a music generative model and investigate melodic anticipation/expectation and information dynamics in machine generated music.

Music and emotion are strongly linked, and listeners can feel different emotions directly or indirectly through music. Engaging emotion as a component of a musical interface has great potential for composing creative music and expressing messages in an effective way. However, emotions are not tangible objects that can be exploited in music information retrieval as they are difficult to capture and quantify in algorithms. In this dissertation, we efficiently combine machine learning techniques for understanding and extracting the emotional context in music and audio data. Several machine learning models are implemented and tested in order to understand the connection between music and emotion in a way that wasn't possible before.

First, we look at the technology of music and audio understanding and design an algorithm for improving the real case scenario for the application of sound understanding. Next, we introduce a generative machine learning model for automatic music composition which uses emotional aspects in musical dynamics for the machine generated music. In order to classify music according to emotions, the deep audio embeddings method was tested, and we show its efficacy for the automatic music emotion recognition task. Lastly, we propose an interactive audio interface that sonifies emotion. The idea is to use human facial gesture data to detect emotions and categorize these into several emotional states for sonification. Rather than simply detecting facial gesture data, it automatically extracts emotional states and produces sound output transition. This dissertation makes the technology more accessible for creative purposes so people can analyze the emotions of music by using machine learning methods and generate machine learning applications using emotional attributes of music.

Part I.

Introduction

Chapter 1

Introduction

Advances in Artificial Intelligence (AI) and machine learning have opened up new possibilities for music and audio understanding. Various information is implied in music and audio, and the information can be combined with machine learning technology to develop multimedia entertainment systems. Specifically, emotion-based machine learning technology can play a role in creating new types of sounds with context information in real-time. Understanding the relationship between music and emotion can inspire users to adapt it to create art and music in new ways. In this dissertation, we argue that the ability to understand the emotional aspects of music and audio, and the application of that understanding to AI can allow for computational creativity.

We propose several deep-learning methods that understand the context-based emotional information and affective capabilities of AI through visual and musical aspects from the data. In this thesis, we look at how to process music and audio data and extract the desired information from it. Specifically, we investigate machine learning techniques from the Sound Event Detection (SED) research area to the Music Emotion Recognition (MER) research area for understanding sentiment information in music. SED research targets the detection of sound events. The goal of the study is to provide both the event label and the event time boundaries while multiple events can be existing in audio data. MER research aims to search and organize music information based on its relevance to specific emotion queries. These two research areas have similarities in distinguishing between different sound events or different emotion classes. In MER, we are interested in organizing large collections of music according to their emotion class. For effective emotion labeling, we looked at the results of recent SED studies and decided to apply one of the recent methodologies, deep audio embeddings. The following are the advantages of adapting deep audio embeddings to MER.

Emotion labeling mainly focuses on human featured engineering, so it is hard to generalize the emotion labeling method to various datasets. In the SED study, there is an advantage of automating event detection using deep audio embeddings. With the rapid development of deep learning, we can automatically extract vector representations from audio data, and such vector representations of audio, deep audio embeddings, can be used for classification or verification tasks. We optimize this deep audio embeddings technique to music that can improve emotion labeling and also guide AI systems to recognize emotion in real-time aligned with musical dynamics. We also show how we understand the connection between music and emotion from visual aspects of data such as human facial expressions and how it can lead to enhanced communication with music. We implement and present a demo for the application which can interact between music and emotion in real-time.

When using machine learning models for designing applications, one of the limitations you may encounter is related to the training data. Not all data can be used as training data, and the system could forget previous information while modifying the pre-trained model. To improve this, we propose a new machine learning model based on an incremental learning algorithm. Using the incremental learning algorithm, the system can expand its knowledge based on existing ones. For example, there is a SED system which pre-trained with three sound events for detecting different sounds. By applying the algorithm, the system can also detect a new additional sound that has not been trained before. For doing this algorithm design, we propose and implement a Neural

Adapter structure that can effectively support this information gap in the learning process. The details of the system structure have been introduced in the main chapter.

This dissertation is about AI applications that can allow for computational creativity. One of the interesting abilities of machine learning technology is to create something new. We design a model for automatic music composition within emotional context. We show that optimizing such models using machine learning achieves computational creativity in predicting emotional anticipation. The connection between music, emotion, and the music generative model, can be related to some views of melodic anticipation and expectation. In the cognitive perception process, the view of expectation can be associated with both biology and culture [Hur08]. When it comes to music, we can think about this melodic anticipation or expectation with a machine learning generative model. During the composition process, the generative model tries to learn to understand musical structure based on training data and use the knowledge to predict the next level of musical notes. This process is related to finding the relationship between musical expectation and the mechanism of the generative machine learning model. Musical outputs are tested with Information Rate which is a measure of understanding the aspects of information dynamics in music.

The primary goal of the dissertation is to advance music content usage for designing music related applications based on machine learning. In this dissertation, we create these main intellectual contributions: (i) for music emotion recognition, we use the music semantic representation of deep audio embeddings, to prove and test a method of understanding different sound events that can work for automatic music emotion recognition, (ii) we design an incremental learning algorithm for real case scenario of audio understanding application and introduce a new learning strategy of minimal training data and maintaining the performance of existing knowledge, and (iii) we present a generative model for automatic music composition and test generated outputs in the context of information dynamics.

1.1 Dissertation Organization

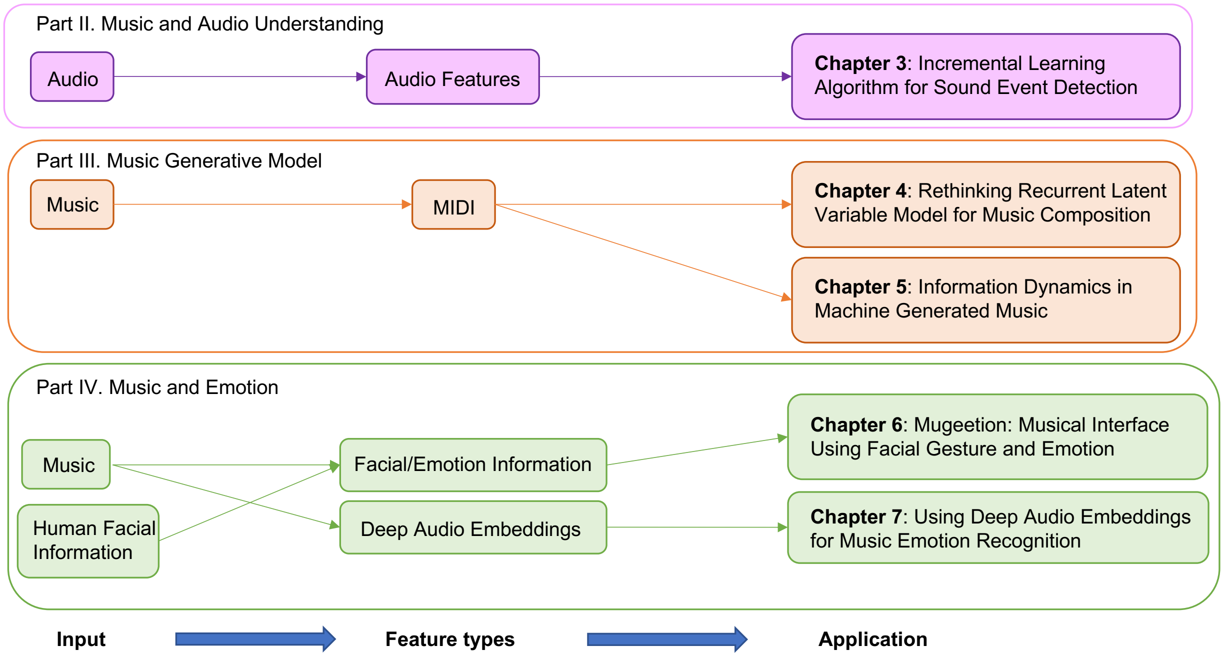

Part II. Music and Audio Understanding
Audio
Audio Features
Chapter 3: Incremental Learning Algorithm for Sound Event Detection
Part III. Music Generative Model
Music
MIDI
Chapter 4: Rethinking Recurrent Latent Variable Model for Music Composition
Chapter 5: Information Dynamics in Machine Generated Music
Part IV. Music and Emotion
Music
Human Facial Information
Facial/Emotion Information
Deep Audio Embeddings
Chapter 6: Mugeetion: Musical Interface Using Facial Gesture and Emotion
Chapter 7: Using Deep Audio Embeddings for Music Emotion Recognition
Input
Feature types
Application

Figure 1.1: An Overview of the Dissertation Organization

We have three main parts in this thesis: Music and Audio Understanding, Music Generative Model, and Music and Emotion (see Figure 1.1). This dissertation starts with a discussion in the preliminaries of music and audio feature processing. Chapter 3 in Part II contains the research on building an incremental learning algorithm for SED with a focus on detecting both known/unknown sound events without forgetting the learned knowledge during training. In Part III, Chapter 4 presents our research on designing a variational autoencoder model for generating a novel sequence of music. Chapter 5 presents the usage of Information Rate as a measure for understanding the information dynamics in machine generated music.

Part IV focuses on finding a methodology for the connection between music and emotion. Chapter 6 shares a background of understanding human facial information in real-time and shows its application of musical interface using human facial expression. Chapter 7 will present the usage of deep audio embeddings for automatic emotion recognition. Finally, in Chapter 8, we review the study of understanding listener’s preference in the context of musical structure.

This dissertation is about how we understand emotional aspects in music and how we connect music and emotion with different technologies. Not just understanding the connection between music and emotion, this dissertation is also about how we could be creative based on understanding those connections. With understanding of emotional context and information dynamics in musical structure, we can be creative composers using machine learning models. Based on the findings of the dissertation, we could, therefore, capture and generate emotion-based musical creativity, leading to engaging and meaningful musical experiences from different angles.

Part II.

Music and Audio Understanding

Chapter 2

Audio Feature Processing Preliminaries

In this chapter, we look at the ways for analysis of music and audio data. The methodology of audio analysis can be varied based on its research purpose. For example, for automatic music composition, we need to understand previous musical structures as well as the characteristics of each sound feature for the synthesis process. Also, it's important to understand the frequency spectrum and envelope from audio signals when discussing timbre in the audio. Depending on which components are used, the music generation output can be different and the generative model could be also changed. In the study of aiming to create a new sequence of music, spectrogram has been mainly used after Short-Time Fourier Transform (STFT) which is one of most representative sound characteristics in order to analyze existing musical features.

2.1 Related Work

[GFG18] demonstrated how to learn object sounds by watching the unlabeled video. For segregation of the sounds, they performed non-negative matrix factorization (NMF) on each video's audio channel. For single-channel audio source separation, the mixture of time-discrete signals was transformed into a magnitude or power spectrum. Because the purpose of this work mainly focused on audio source separation, it performed NMF independently on its audio magnitude spectrogram to process its spectral patterns. [OIM$^+$16] showed the methodology of sound synthesis from silent videos. The proposed algorithm utilized a Recurrent Neural Network (RNN) to generate next-level sound features and then created a waveform of the sound with an example-based synthesis procedure. For learning sound characteristics, it approached speech synthesis methods that used RNN to predict sound features. By decomposing the waveform into sub-band envelopes, the study got a simple representation obtained by filtering the input waveform and applying a nonlinearity. The study used a bank of 40 bandpass filters on an Equivalent Rectangular Bandwidth (ERB) scale and processed Hilbert transform. In this process, they learned about sound properties from sub-band envelopes on a sample of white noise for generating a waveform. Another work from the same research group, [OE18] examined three different methodologies for three applications, sound source localization, e.g., visualizing the source of sound in a video, audiovisual action recognition, and on/off screen audio source separation. The study targeted taking a spectrogram for the mixed audio as input and a reconstructed spectrogram for the two mixture components. For the sampling, they sampled 2 sound clips from 5 second long clips, normalized each waveform's mean squared amplitude, and utilized spectrograms with a 64 ms frame length and 16 ms step size for producing 128x1025 spectrogram. [ZGR$^+$18] introduced a model for enhancing the speech of desired speakers in a video. In the system, the model focused the audio on specific speakers in a scene and improved the speech separation quality. To pre-process before training, they computed the STFT of 3-second audio segments. Then, features were fed into the model to learn an audio representation using Convolutional Neural Network (CNN). [ZWF$^+$18] introduced a method of generating sounds from videos in wild. They sampled audio directly from the video dataset at 16kHz. For the frame-based, they set a step size to 1024, and 159744 time steps per 10 seconds. [SZL$^+$18] presented a conditional video generation network that can generate the talking face video with accurate lip synchronization. They extracted the audio feature and image identity feature using two convolutional encode networks. The audio features, Mel-Frequency Cepstral Coefficients

(MFCC), were extracted and fed into a convolutional encode network. They chose the MFCC feature due to its effectiveness in the speech recognition task.

2.2 Learning Audio-Visual Correspondence

In this section, we review different encoding methodologies for audiovisual signals and address some examples of prior work for finding a correlation between audiovisual features. Based on existing audiovisual studies, we look at which audiovisual features have trained together, and we also review the purpose of its training. In this section, we think about the question ”Are there situations in which adding audio feature detection could enhance human expectations beyond what an image can offer?” and how the system can be improved with audio features.

Several studies focused on audiovisual models to find a joint component between sound and image. Cross-modal audiovisual perception is a challenging research field as a discovery of strong correlations in human perception of auditory and visual stimuli. Most studies presented a model to find a solution for sound separation and localization. For example, [EML$^+$18, OE18] showed methodologies for learning audiovisual correspondence which uses audio to supervise visual representations. [ZGR$^+$18] also introduced a study to utilize visual features for improving speech separation quality. It has been noted that models have been varied based on the goal of their system. Such visual features are usually high dimensional and they perform Principal Component Analysis (PCA) on the extracted features of the training set to reduce dimensionality. The PCA dimensionality is chosen by cross-validation on a validation set separately for each trait. The PCA weights are saved and further used in fine-tuning the neural networks model. [SZL$^+$18] achieved a synthesis of talking face video by Conditional Recurrent Adversarial Network. For processing hybrid features, they applied recurrent units on the hybrid code of audiovisual features which can generate improved video quality. [SSKS17] presented a photorealistic video of Obama speaking with accurate lip-sync. For doing this, the system connected audio features with mouth gestures for understanding the connection between the two. They used RNNs for synthesizing a high-quality video of Obama speaking with accurate lip-sync. For photo-realistic mouth texture, they extracted audio features as input to RNNs that generate output a sparse mouth shape. They processed the audio using standard MFCC and the mouth shape with 18 lip points reduced by a PCA basis. Using MFCC audio features, the study improved the quality of video generation and shared the possibility of engaging audio features for the improvement both in computer vision and pattern recognition studies.

In this chapter we review previous methods of sound feature processing and audio-visual correspondence, which is targeting for cross modal audio visual perception algorithm. While this chapter focuses on the preprocessing of data before training, the next chapter more discusses audio processing for sound event detection systems.

Chapter 3

Incremental Learning Algorithm for Sound Event Detection

This chapter presents a new learning strategy for the Sound Event Detection (SED) system to tackle the issues of i) knowledge migration from a pre-trained model to a new target model and ii) learning new sound events without forgetting the previously learned ones without re-training from scratch. In order to migrate the previously learned knowledge from the source model to the target one, a neural adapter is employed on the top of the source model. The source model and the target model are merged via this neural adapter layer. The neural adapter layer facilitates the target model to learn new sound events with minimal training data and maintaining the performance of the previously learned sound events similar to the source model. Our extensive analysis on the **DCASE16** and **US-SED** dataset reveals the effectiveness of the proposed method in transferring knowledge between source and target models without introducing any performance degradation on the previously learned sound events while obtaining a competitive detection performance on the newly learned sound events.

3.1 Introduction

Sound Event Detection (SED) is a rapidly growing research area that aims to analyze and recognize a variety of sound events in a continuous audio signal. Neural Networks based methods such as Convolutional Neural Networks (CNNs) have recently been used for SED systems to advance the performance of these systems [MHV18, MHV16b]. In the Detection and Classification of Acoustic Scenes and Events (DCASE) Task 4 [DCA], the state-of-the-art SED systems have been tested using real data that is either weakly labeled or unlabeled and simulated strongly labeled data with the onset and offset times of sound events.

Although SED problem has been attracting many researchers, a vast majority of the state-of-the-art systems are focused on advancing the performance of the SED systems by utilizing weakly labeled data [DCA]. To our knowledge, one of the important and non-investigated challenges of the current SED models is their closed-set nature, where a fixed and limited number of known classes are used during the training. It is difficult to collect exhaustive training samples or to properly annotate all the training data to train the classifiers. Hence in the closed-set classifiers, only a limited number of classes are considered for training, with the assumption that during test time, the test data is drawn from the same set of classes as the training data. However, the SED systems in nature are open-set problems, in other words, the test data could include samples associated with unknown sound events as well. Therefore, it is always desired to have a flexible model that can learn new classes, once new training data including new sound events becomes available. Then again, it is required to still remember the previously learned classes after adopting the new classes, and learning capability is referred to as continuous learning or incremental learning.

One of the main challenges associated with these types of continuous learning algorithms is catastrophic forgetting [MC89]. That is if the information about the previously learned categories is unavailable when a new task is added, it overwrites the previously learned information.

Hence, it leads to the performance degradation of past tasks. The ability of continuous/incremental learning over time represents a long-standing challenge for Machine Learning and Neural Networks [TM95]. Recently, in the areas of computer vision and natural language processing [GBC16, CM19], Transfer Learning (TL) has shown great potential to 1) identify the transferable knowledge by accommodating new knowledge and 2) retain previously learned information. Some recent works have explored TL for audio applications [CWSB19, KKF18, JPL19, SGH^{+}19], which focus on knowledge transfer between databases with various qualities, mismatch downstream tasks, and domains. However, it remains to be seen how a flexible TL model for SED task to audio knowledge transfer can be done.

In this chapter, we present an incremental learning algorithm for SED applications effectively transferring knowledge from a source model to a target model. Our method updates the target model when new sound events are available without any catastrophic forgetting. Motivated by the recent TL advances in natural language processing [CM19], we utilize a neural adapter to bridge the gap between the source model and the target model. We combine new neurons for transferring parameters from the source model and implement a neural adapter to lessen the gap between the source and the target data distribution. For testing the performance of the neural adapter, we test the basic simple TL approach, then we show the impact of our proposed neural adapter for the SED task, while testing several TL options on just one category. The results show that our method provides an effective knowledge transfer mechanism between source and target domains without any additional training examples and any performance degradation of the previously learned tasks in the source domain. Our learning algorithm helps to transmit the predictions from the source model into that of the target model.

The rest of the chapter is organized as follows. In section 2, we cover the proposed SED incremental learning algorithm, followed by the experimental results in section 3. The conclusion and the discussion are provided in section 4.

3.2 SED Incremental Learning Algorithm

3.2.1 SED Problem Formulation

Given an input audio file that includes several acoustic scenes, a standard SED predicts the corresponding labels that indicate the annotation of all the sound events in the scenes. The time-series audio input is represented by audio embedding vectors and the event label file includes the information of event specification, such as the sound event onset/offset time and sound event label.

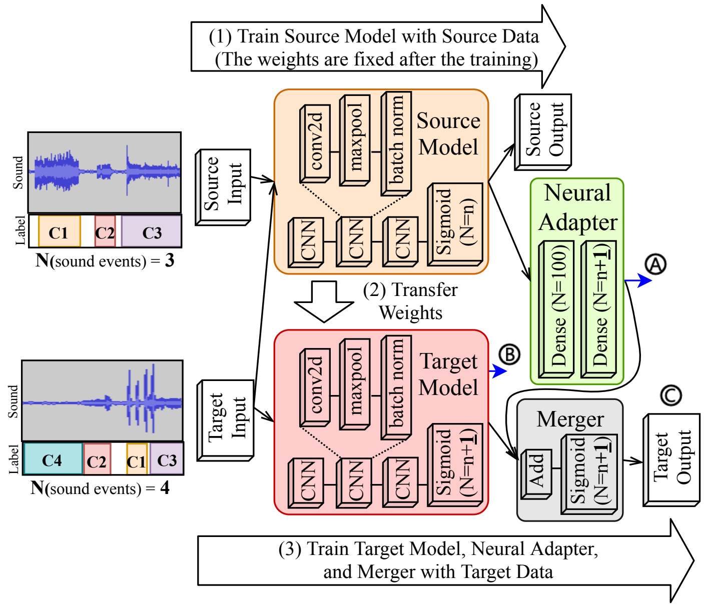


Figure 3.1: **SED incremental learning algorithm structure.** Source input consists of *N* sound events and target input consists of *N+1* sound events leaving one sound event out to be incrementally learned later with the target model. The source and target model includes three 2D convolution layers, 2D max-pooling operation, and batch normalization layers. We use a sigmoid function for output activation of the source and the target model. Source model's weights are fixed during target model training. The target model is initialized with the optimal weights of the source model. The neural adapter consists of fully-connected dense layers. Note that source and target models have different output dimensions due to the new class in the target data. Ⓐ& Ⓑ are intermediate outputs for analysis, we add Ⓐ and Ⓑ to Ⓒ using merger, and Ⓒ is the final output of our model.

3.2.2 Transfer Learning (TL) Workflow

Our TL mechanism consists of several steps (see Figure 3.1); (i) a pre-trained source model, M_S, for a certain number of categories, in the source domain, D_S; (ii) a target model, M_T, that utilizes the source model M_S parameters as a starting point to learn the target domain, D_T. In addition to the typical TL mechanism, which is (i) and (ii), we add a neural adapter (iii) a connection system between the source model M_S and the target model M_T for effectively transferring knowledge from the source domain D_S to the target domain D_T. This connection will alleviate the effect of information discrepancy and prevent any catastrophic forgetting on the previously learned information. Finally, we (iv) jointly train the target model M_T and the connection system together to effectively learn the target domain D_T information. Note that the parameters of the source model M_S are not updated during this target model M_T training process.

3.2.3 Model Architecture

SED Source Model

Figure 3.1 describes the process and the structure of our SED incremental learning methodology. We revise the Convolutional Neural Network (CNN) proposed by Salamon and Bello [SB17], which includes three 2D convolution layers, a 2D max-pooling operation, and batch normalization layers. Each layer processes 64 convolutional filters. The input to the network is a Mel spectrogram of size 128x128 that is extracted from a one-second audio file. ReLU activation functions are applied to the convolutional layers to reduce the backpropagation errors and accelerate the learning process [GBC16]. Sigmoid function are used as the output activation function with N classes. Adam optimizer [KB14] and binary cross-entropy loss function are used. The stopping criterion is set as 500 epochs with an early-stopping rule, if there is no improvement to the F1 score during last 100 learning epochs [MHV16a]. The final model has 720k parameters. This model is implemented in Keras [Cho15].

Incremental Learning using Neural Adapter

In this work, the M_S is trained for N sound events using the aforementioned CNN model. The goal is to create a M_T for *N+1* sound events without training from scratch. Note that the N sound events are common for both M_S and M_T. To this end, the M_T has the same CNN structure as the M_S and the trained parameters of the previously learned N sound events are utilized from the M_S as an initial training point for the M_T. To account for the new category in the M_T, we modify its output layer with *N+1* sigmoid activation. It is well known that learning a new task via such a simple transfer learning paradigm usually results in forgetting the previously learned classes while adding new classes. To avoid this knowledge-lost problem, we adopt the TL mechanism proposed by Chen and Moschitti [CM19]. In this method, to effectively transfer the knowledge between the M_S and the M_T, a neural adapter is utilized to bridge the two models and jointly trained with the M_T. More specifically, a neural adapter consists of two fully-connected dense layers over the last layer of M_S is used to connect the M_S to the M_T. This process is called the element-wise summation which integrates the outputs from the source and target domain and finally processes *N+1* categories. The parameters of the neural adapter and the M_T are learned simultaneously while the M_S parameters are fixed.

3.3 Experimental Results

3.3.1 Datasets

We start our implementation with building M_S using only N sound events for D_S, out of the *N+1* events, leaving one sound event out to be learned incrementally later with the M_T. We train the M_S using the D_S until the optimal parameters are achieved. These parameters and results will be saved and re-utilized for D_T. We evaluate our algorithm over three datasets; the DCASE 2016 challenge Task 2 (**DCASE16**) [MHV16b], the UrbanSound-SED (**US-SED**), and

UrbanSound-8K (**US-8K**) [SJB14] dataset.

The **DCASE16** dataset includes eleven different sound events for the SED challenges[1], *clearing throat, coughing, door knock, door slam, drawer, human laughter, keyboard, keys (put on the table), page turning, phone ringing, and speech*. In this work, we use four sound events; *"door knock", "door slam", "keyboard",* and *"phone ringing"*. To generate soundscapes from these sound files, we use the Scaper open-source library [SMC+17] for the synthesis and augmentation [2]. We create 800 soundscapes for M_S training data and 200 soundscapes for each test/validation data. The duration of each soundscape audio file is ten seconds. It is worth noting that the soundscapes are generated in such that each event appears at least once and a maximum of two times in every soundscape file. This **DCASE16** dataset denotes a clean and well-labeled dataset for our experimental setting.

The **US-SED** dataset [SJB14] includes ten different sound classes; *air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren,* and *street music*. In this work, we use the pre-generated UrbanSound soundscape audio files from the Scaper study [SMC+17]. It has 10,000 soundscape files with a duration of ten seconds. Each of the soundscape files has a minimum number of sound events as zero, and the maximum number of nine. Thus, some of the soundscapes might be empty of the sound events of interest. This situation can be interpreted as a more realistic SED framework compared to the aforementioned **DCASE16**, where each of its soundscape files contains all the sound events. For our experiment, we considered five sound events; *"car horn", "dog bark", "gun shot", "siren",* and *"street music"*. 4,995 files are used for M_S training, and 1,665 files for each of the test and verification data.

[1] `http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-synthetic-audio`

[2] We revise this open-source implementation:
`https://github.com/justinsalamon/scaper_waspaa2017`

Table 3.1: **F1-score of the M_S, Simple TL, and Neural Adapter TL over different settings on DCASE16.** Simple TL means M_T built based on a simple (or typical) TL method without the neural adapter, and Neural Adapter TL means M_T built using the neural adapter TL method in addition to the simple TL. The M_S sections shows the F1-score of M_S for different D_S domain settings. Simple TL section illustrates the F1-score of the M_T that are built via a simple TL approach. Neural Adapter TL section reports the F1-score of the proposed approach. D_S columns of the Simple TL and Neural Adapter TL indicate the F1-score of M_T on the previously learned classes. The New columns present the F1-score of the newly learned class and All columns report the overall F1. Rows in the table depict each of the test scenarios assuming that unseen label in D_S is newly introduced in D_T. DCASE16 includes four sound event classes, C1: keyboard, C2: door slam, C3: phone ringing, and C4: door knock.

DCASE16	M_S	Simple TL			Neural Adapter TL		
Labels in D_S	D_S	D_S	New	All	D_S	New	All
C_1 C_2 C_3	.9444	.8518	.666	.8055	.944	.6666	.8518
C_1 C_2 C_4	.8888	.7777	.8888	.8055	.8886	.8888	.8888
C_1 C_3 C_4	.8518	.8518	1.0	.8888	.8513	.8	.8388
C_2 C_3 C_4	.7407	.8518	1.0	.8888	.74	1.0	.8055
Overall	.8561	.8332	.8888	.8469	.8559	.8388	.8462

Table 3.2: **F1-score of the M_S, Simple TL, and Neural Adapter TL method over different settings on US-SED and US-8K.** This table is configured the same as Table 3.1 with different datasets and sound classes. US-SED and US-8K include five sound event classes, C1: street music, C2: siren, C3: gun shot, C4: dog bark, C5: car horn).

US-SED	M_S	Simple TL			Neural Adapter TL		
Labels in D_S	D_S	D_S	New	All	D_S	New	All
C_1 C_2 C_3 C_4	.5503	.5782	.6697	.5965	.5640	.6962	.5966
C_1 C_2 C_3 C_5	.5897	.5918	.4316	.5598	.589	.5270	.5789
C_1 C_2 C_4 C_5	.5985	.6040	.4493	.5731	.5947	.4232	.5666
C_1 C_3 C_4 C_5	.5850	.5826	.6151	.5892	.5826	.6188	.5929
C_2 C_3 C_4 C_5	.5891	.5822	.5572	.5573	.5875	.5977	.5938
Overall	.5825	.5877	.5445	.5791	.5836	.5725	.5780

US-8K	M_S	Simple TL			Neural Adapter TL		
Labels in D_S	D_S	D_S	New	All	D_S	New	All
C_1 C_2 C_3 C_4	.6041	.5857	.5	.5704	.6023	.5706	.5908
C_1 C_2 C_3 C_5	.5166	.4523	.5233	.5076	.5047	.5285	.5142
C_1 C_2 C_4 C_5	.6277	.5568	.5846	.5999	.6145	.6085	.6133
C_1 C_3 C_4 C_5	.6499	.5865	.5076	.5538	.6192	.5464	.5607
C_2 C_3 C_4 C_5	.6791	.6756	.6461	.6773	.6675	.6567	.6606
Overall	.6154	.5713	.5809	.5818	.6016	.5821	.5871

3.3.2 Performance of the Incremental Learning

We evaluate our algorithm in three different settings; i) evaluating the M_S trained on D_S with *N* sound events, ii) evaluating the M_T that is built via a simple TL on the D_T with *N+1* sound events, and iii) evaluating the model that is built utilizing the neural adapter, where the M_T is merged with the M_S through the neural adapter and trained on D_T. The results are reported in terms of F1-score (see Table 3.1 and 3.2). We perform experiments per every class assuming it is newly introduced in the D_T. Note that *N* indicates three sound event classes for **DCASE16** dataset and *N* indicates four sound event classes for the **US-SED** and **US-8K** dataset.

Table 3.1 and Table 3.2 present the evaluation results for the **DCASE16** and **US-SED**, respectively. These tables include three sections; M_S, Simple TL and Neural Adapter TL. The M_S section shows the F1-score of M_S models for different D_S. Section Simple TL illustrates the F1-score of the M_T that are built via a simple transfer learning approach. Section Neural Adapter TL reports the F1-score of the proposed approach. D_S columns of the Simple TL and the Neural Adapter TL indicate the F1-score of the new models on the previously learned classes after the incremental learning process. The New columns present the F1-score of the newly learned class, and finally the All columns report the overall F1-score on the *N+1* sound events. Rows in the table depict different test scenarios assuming that unseen label in D_S is newly introduced in D_T for incremental learning.

• DCASE16: By paying attention to Table 3.1 and the D_S column in the M_S section versus the D_S column in the Simple TL section, it can be seen that learning sound classes via the simple TL approach, without the neural adapter, results in performance degradation in the previously learned sound events. This drop in performance indicates a catastrophic forgetting in the D_T on the original three categories in D_S. On the other hand, the consistent F1-scores between the D_S column in the M_S section and the D_S column in the Neural Adapter TL section is an illustration that the neural adapter can properly maintain the knowledge learned from D_S while learning the new class in the D_T. This result proves that the neural adapter manages to mitigate the knowledge forgetting and enabling the model to update to the new domain.

• US-SED: The top table of Table 3.2 illustrates the results on **US-SED** dataset. Similar to **DCASE16**, we can see the neural adapter can effectively bridge the knowledge between the M_S and the M_T models while learning the new sound events. However, by comparing the D_S column in the M_S section to the D_S column in the Simple TL section, we can see the simple TL achieves some improvement over the M_S on the previously learned sound events. Unexpectedly, these results are even slightly better than the neural adapter TL approach (see the D_S column in the Neural Adapter TL section). This result does not match the results obtained for the **DCASE16** dataset. The reason for this could be the amount of presented noise during the training of the models. As it is mentioned earlier, in the **US-SED**, some of the used soundscapes in the training data are empty of the sound events of interest. Hence, these files are interpreted as noisy samples by the network. On the other hand, all the sound files used from the **DCASE16** at least contain one of the desired sound events. To assess this argument, we modify the original UrbanSound8K dataset [SJB14] to mimic the settings from the **DCASE16** in creating a more clean dataset[3] and this is the **US-8K** for our next dataset. In **US-8K**, each event appears at least once and a maximum of two times in every soundscape file which is the same setting as **DCASE16** soundscape generation.

• US-8K: This **US-8K** dataset has the same five sound events as the **US-SED**. Similar to **DCASE16**, each sound event appears in each soundscape file at least once. The evaluation results on this dataset is provided in the bottom table of Table 3.2. By looking at the D_S column in the M_S section and the D_S column in the Neural Adapter TL section, it can be seen that similar to the **DCASE16** dataset, the neural adapter method consistently maintains the performance on the previously learned sound events after learning new sound events. Also transferring the knowledge from the M_S via the neural adapter is more effective compared to the simple TL approach. It is

[3] For our experimental setting, US-8K soundscapes are generated based on the UrbanSound8K dataset which can be found on:
`https://urbansounddataset.weebly.com/urbansound8k.html`.

Table 3.3: **Comparison between individual performances from each model over different settings on DCASE16, US-SED, and US-8K.** Ⓐ is the output of the source model with a neural adapter. Ⓑ is the output of the target model. Ⓒ is a final output of our proposed model (see also Figure 3.1).

	DCASE16	US-SED	US-8K
Ⓐ	.5321	.2537	.3871
Ⓑ	.8017	.3498	.5073
Ⓒ	.8451	.4833	.5272

important to ensure that this improvement in the performance is not specific to any target event category, and it is common across different experiments denoted in various rows of the tables.

Summarizing the experimental results from the three datasets, it can be seen that simple TL method without the neural adapter has confronted the degradation of the performance with losing the previously learned knowledge from the source model training. There are also up and down in the performance from specific sound events or dataset in the simple TL method. On the other hand, in the case of the neural adapter approach, it is possible to see the inclination in which the learned knowledge is maintained consistently, and the performance sustains in a balanced manner. Therefore, we show performance consistency on the previously learned sound events through neural adapter while obtaining decent detection performance on the newly learned sound events well.

3.3.3 The Feasibility of Neural Adapter for Incremental Learning

In this section, we study the contribution of each model's outputs separately on the overall performance. In Figure 3.1, we separate three individual outputs in the neural adapter approach; (1) the optimal output of the source model with a neural adapter (Ⓐ in Figure 3.1), (2) the output of the target model (Ⓑ in Figure 3.1) and (3) the final output of our proposed TL model (Ⓒ in Figure 3.1). For this study, the target domain is used in three cases. The results of the analysis are provided in Table 3.3.

It can be seen that when using **DCASE16** dataset, the trained target model via the neural adapter reaches an optimal point to be able to detect all the *N+1(=4)* sound events without the need to have the source model (see Ⓑ&Ⓒ of the DCASE16 column). Therefore, only the target model could be stored and used as a starting point for learning new categories without the need to store the source model and the neural adapter. This result can provide a low footprint continuous learning framework for further model expansion.

In contrast, in the case of using the **US-SED** dataset, the target model and the source model remain complementary to each other for achieving an acceptable final outcome (see Ⓑ&Ⓒ of the US-SED column). It appears that the target model cannot maintain the transferred knowledge from the source model while learning the new sound event. Hence, in order to continuously learn new sound events, we always need to keep the whole TL structure presented in Figure 3.1. This is not feasible for continuous learning in applications that are operated on resource limited platforms, for example, wearable devices. In the case of using the **US-8K** dataset, it has similar aspect of **DCASE16**, but it is hard to see a difference as large as **DCASE16**.

3.4 Conclusion and Discussion

We present an incremental learning algorithm utilizing a TL paradigm for SED application. We use a neural adapter to effectively bridge the gap between the previously learned information in the source model and a target model for learning new sound events. Our extensive analysis shows that utilizing such a mechanism improves the performance of recognizing both known/unknown sound events without forgetting the previously learned knowledge. Thus, our proposed model suits well the scalable and incremental SED applications.

This approach can also be used as a low footprint framework for continuous learning in applications that involve less noisy and well annotated data. However, for the more realistic applications, such as acoustic scene classification systems that involve more noisy data, both the target model and the source model might need to remain connected to achieve the desirable performance. Addressing such a challenge remains the focus of our future work.

Chapter 3 is adapted from published material in ”Incremental Learning Algorithm for Sound Event Detection”. Koh, Eunjeong, Saki, Fatemeh, Guo, Yinyi, Hung, Cheng-Yu, and Visser, Erik. IEEE International Conference on Multimedia and Expo, 2020. The dissertation/thesis author was the primary researcher and author of the paper.

Part III.

Music Generative Model

Chapter 4

Rethinking Recurrent Latent Variable Model for Music Composition

We present a model for capturing musical features and creating novel sequences of music, called the Convolutional-Variational Recurrent Neural Network. To generate sequential data, the model uses an encoder-decoder architecture with latent probabilistic connections to capture the hidden structure of music. Using the sequence-to-sequence model, our generative model can exploit samples from a prior distribution and generate a longer sequence of music. We compare the performance of our proposed model with other types of Neural Networks using the criteria of Information Rate that is implemented by Variable Markov Oracle, a method that allows statistical characterization of musical information dynamics and detection of motifs in a song. Our results suggest that the proposed model has a better statistical resemblance to the musical structure of the training data, which improves the creation of new sequences of music in the style of the originals.

4.1 Introduction

Neural networks have enabled automatic music composition with little human interruption. Many approaches have been proposed to generate symbolic-domain music, such as Recurrent Neural Networks (RNNs) [CM01, WERA, YCY17] and RNN combined with Restricted Boltzmann Machine (RNN-RBM) [BLBV12]. However, previous studies on RNN-based music generation lack in: 1) understanding the higher level semantics of a musical structure, which is critical to music composition; 2) generating novel and creative patterns that avoid literal repetitions [BWH16]. Most of the previous studies for music generation use so called one-to-many RNNs, where a single musical unit (such as a single note or one bar of music) is used to predict the next unit in a recurrent manner.

In addition, recent studies exploiting Convolutional Neural Networks (CNNs) for the generation of symbolic-domain music use rich representations that are more adaptive to creating complex melodies, such as C-RNN-GAN [Mog16], MidiNet [YCY17], and MuseGAN [DHYY17]. In general, the frameworks' processes consist of: 1) representing multi-channel MIDI files using filters learned by CNN layers; 2) setting a discriminator to learn the distributions of melodies; and 3) processing longer sequences of data. CNNs have been well-established as choices for recognition and classification tasks in 2D data such as images, so they make better candidates for extracting melodies (horizontal) or chord (vertical) structure in musical time-pitch space.

The Variational Autoencoder (VAE) has been also explored as a generative model for creating multimedia structure. In [HUW17, REE], VAE has been trained for musical creation which can better capture musical structure and generate complex sequential results. VAE exploits samples from a prior distribution and generates a longer sequence. In addition to VAE, Variational Recurrent Neural Networks have been introduced in [FvA14, CKD$^+$15]. These studies show that Variational Recurrent Neural Networks can create sequential data by integrating latent random variables in recurrent ways. To do this, the model utilizes encoded data in latent space in each step. This suggests that these recurrent steps can make it possible to generate more diverse styles tasks while incorporating features from data in a recognizable way. However, these previous studies do not analyze the outputs in music generation, and how to maintain a designated theme across the entire song remains unchallenged.

In this paper, we propose a Convolutional-Variational Recurrent Neural Network which combines the strength of CNN and VAE together. We show that: 1) CNN feature learning can improve statistical resemblance to musical structure of the training data; 2) utilizing encoded data in latent space can extend the dynamic creation of new sequences of music. Our model consists of a CNN to learn a better representation of bar-level of music and a Variational Recurrent Neural Network for generating novel sequences of music. In this model, random sampling and data interpolation can generate sequential data more dynamically while including learned aspects of the original structure. We model the class of bar-level data points to enable the recurrent model to infer latent variables.

To validate our model, we adopt Information Rate (IR) as an independent measure of musical structure [WD14], in order to assess the effect of the repetition versus variation structure constraints and compare our approach with that of RNN models for music generation [WERA, YCY17]. We use IR implementation by Variable Markov Oracle (VMO) to discover optimal predictive structure in the audio output of the different models. The IR analysis using VMO provides an independent evaluation of the structure of the song as captured by the sequence of audio Chroma features. Furthermore, we present a detailed motif analysis of the data and provide a qualitative discussion of generated musical samples.

The rest of the paper is structured as follows: Section 2 gives an overview of related models and computational approaches to music generation. Section 3 describes the components involved in the Variational Recurrent Neural Network approach. Section 4 describes the IR experimental validation of the sequential modeling in the context of Nottingham dataset [not], a collection of 1200 British and American folk tunes. We discuss the empirical findings in Section 5 and give future perspectives.

4.2 Backgrounds

4.2.1 Music Generation with Recurrent Neural Networks

Automatic music generation with neural networks is a task to automatically generate music using parameters learned from a collection of music samples. Neural networks have enabled automatic music composition with little human interruption. In this chapter, we review previous studies of music generation by Recurrent Neural Network (RNN) and describe the pros and cons of this neural network for music generation purposes. We also review some variations of RNN such as RNN-RBM, or Variational RNN.

First, we look at ways to generate music sequences based on RNN. RNNs are the type of neural networks that have been explored for sequential information. RNNs perform the same function for every single element of a sequence with the result based on previous computation (see Figure 4.1).

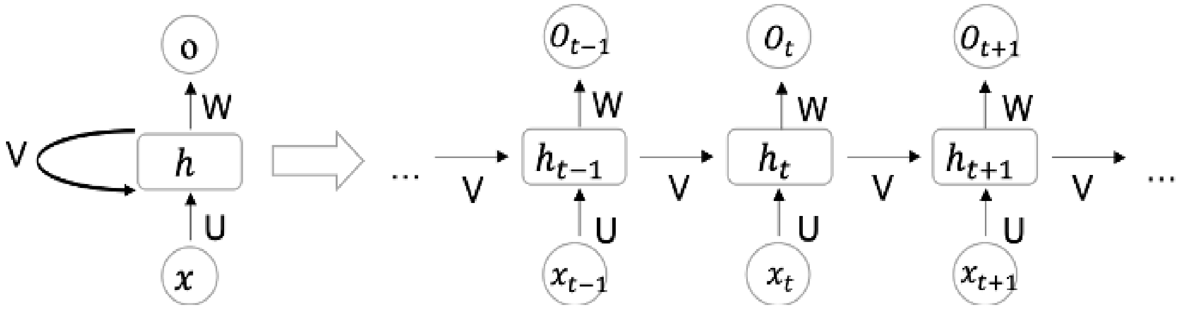


Figure 4.1: Block diagram of RNN structure (x_t: input vector, h_t: hidden layer vector, o_t: output vector, W,U,V: parameter matrices/vector)

The most common RNNs is the Long Short-Term Memory (LSTM) network which is used for modeling long-term dependencies. This is a type of RNNs that can learn via gradient descent and recognize long-term patterns. LSTM structure has been widely utilized for solving problems where the network has to memorize information for the long term.

We can use RNNs to design a generative model. With the increase in computational resources and recent advancements in RNN architecture, we can generate a sequence of music such as a song with repetition, or a random sequence of notes. During the generation process, we get input sequence data by normalizing the hidden layer to a probability distribution and calculating the product of the probabilities. Ideally, the generated song could have variations with different levels of similarity to the training data.

There have been various approaches to implementing multiscale RNNs. The most popular approach is to set the timescales as hyperparameters instead of treating them as dynamic variables that can be learned from the data [EHB96, KGGS14]. In the same vein, the Variational Recurrent Neural Network has been shown to perform well on generating sequential outputs by integrating latent random variables in RNNs [FvA14, CKD$^+$15]. For the latent variable structure, the model utilizes encoded data in latent space in each step. The previous studies showed that these recurrent steps can make it possible to be flexible on the generation of more diverse styles of music while incorporating features from data in a concrete way.

In general, music generation studies on RNNs face challenges related to the ways of capturing and learning from training data [CM01, WERA, BLBV12]. In the following sections, we explore a recent methodology that has been approached for improving music generation based on RNNs.

4.2.2 Music Generation with Variational Latent Model

Our architecture is inspired by the Variational Autoencoder (VAE) as a stochastic generative model [KW13, RMW14]. In general, the model consists of a decoding network with parameters θ that estimates the posterior distribution $p_\theta(\mathbf{x}|\mathbf{z})$, where $\mathbf{x}$ is the sample being estimated and $\mathbf{z}$ is an unobserved continuous random variable. The prior probability $p_\theta(\mathbf{z})$ in this case is assumed to be generated from a Gaussian random variable with zero mean and unit variance. In this form, the true posterior distribution $p_\theta(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})/p_\theta(\mathbf{x})$ is intractable, so an encoding network q with parameters ϕ is used to estimate the posterior as $q_\phi(\mathbf{z}|\mathbf{x})$. The encoding network is trained to estimate a multivariate Gaussian with a diagonal covariance.

$$\log q_\phi(\mathbf{z}|\mathbf{x}) = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathrm{I}) \tag{4.1}$$

Noise can then be sampled using Gaussian distribution with the mean and standard deviation learned by the encoding network.

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} = \mathcal{N}(0, \mathrm{I}) \tag{4.2}$$

Thus, the parameters of the encoding network ϕ can be estimated with gradient descent using the re-parameterization trick [KW13] and the total loss of the network is calculated as

$$\mathcal{L}(x; \theta, \phi) \simeq \frac{1}{2}\sum_j (1 + \log(\boldsymbol{\sigma}_j^2) - \boldsymbol{\mu}_j^2 - \boldsymbol{\sigma}_j^2) + \frac{1}{L}\sum_l \log p_\theta(\mathbf{x}|\mathbf{z}^{(l)}) \tag{4.3}$$

where the first term on the right-hand side is an approximation to the KL divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z})$.

Intuitively speaking, the variational approach adds a probabilistic element to latent model that allows not only generation of new variations through random sampling from a noise source, but it is also trying to distill more informative latent representation by making the **z**'s as independent as possible. In this view, the KL component in Equation 4.3 can be considered as a probabilistic regularization that seeks the simplest or least assuming latent representation. Taking this analogy one step further, one could say that a listener infers latent variables from the musical signal she/he hears, which in turn leads her/him to imagine the next musical event by predicting musical continuation in the latent space and then ”decoding” it into an actual musical sensation.

Technically speaking, during training, the model is presented with samples of the input which are encoded by q to produce the mean and standard deviation for the noise source. A noise sample is then generated and passed through the decoding network which calculates the posterior probability p to determine the sample generated by the network. The network is trained to reproduce the input sample from the noise source, so the second term on the right-hand side of Equation 4.3 can be either mean squared error in the case of a continuous random variable or cross entropy for discrete random variables. At test time, random samples are generated by the noise source, which is used by the decoder network to produce novel outputs.

4.2.3 Music Information Dynamics and Information Rate

We analyze our music generation output with IR value from VMO, in order to assess the predictability of a time series sequential data, and understand consistency in a song (e.g., motives, themes, etc) [WD15]. VMO allows to measure music information dynamics and higher IR value presents structural note transition in a generated music than the one with lower IR value. In Equation 5.1, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ denotes time series x with N observations, and $H(x)$ denotes the entropy of x. As a result, IR denotes corresponding information between the current and previous observations, which enables the understanding of variation and repetition in a song segment.

$$IR(\mathbf{x}_1^{n-1}, \mathbf{x}_n) = H(\mathbf{x}_n) - H(\mathbf{x}_n | \mathbf{x}_1^{n-1}) \tag{4.4}$$

For quantitative evaluation, VMO has also been explored by other deep learning research focusing on music generation [BHP17, LGW16].

4.2.4 Search for Optimal Threshold

Evaluation of IR requires knowledge of the marginal and conditional distributions of the samples $\mathbf{x}_n$ and $\mathbf{x}_1^{n-1}$. This function is not known and the whole purpose of modeling the data

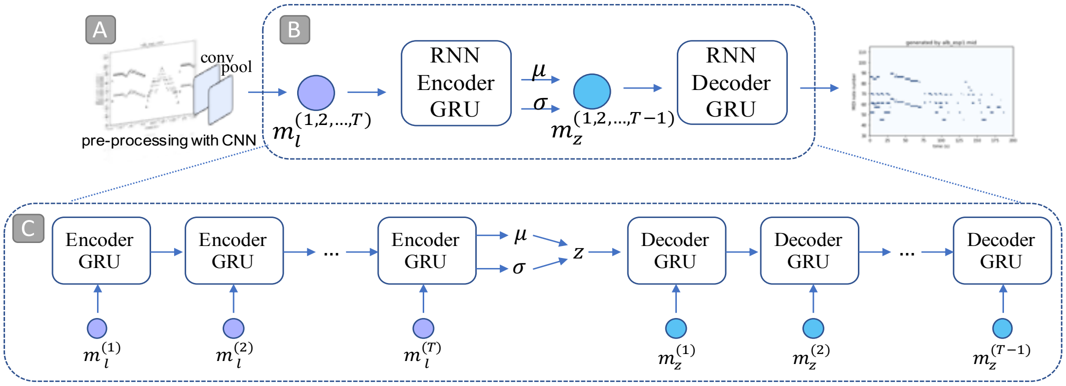


Figure 4.2: Convolutional-Variational Recurrent Neural Network architecture

with our variational latent model is to try to approximate such probabilities. So how can IR be used without an explicit knowledge of the distribution?

The idea behind Music Information Dynamics analysis is estimating mutual information between present and past in musical data in a non-parametric way. This is done by computing similarity between features extracted from an audio signal that was synthesized from MIDI, using human engineered features and distance measures known from musical audio processing. VMO uses a string matching algorithm, called Factor Oracle (FO), to search for repeated segments (suffixes) at every time instance in the signal.

A crucial step in VMO is finding a threshold θ to establish similarity between features. For each threshold value, a string compression algorithm is used to compute the mutual information between present and the past, measured in terms of the difference in the coding length of individual frames versus block encoding using repeated suffixes. So the optimal IR in VMO is found by searching over all possible threshold values and selecting a threshold that gives an overall best compression ratio.

4.2.5 Links between Variational Latent Model and IR

A motivation for using IR as a method to estimate the efficiency of dynamic latent models can be found through the relation between the variational inference loss function and IR using formulation of free energy. The loss function in Equation 4.3, known also as Evidence Lower Bound (ELBO), can be shown to represent so called free energy of the system.

$$\mathcal{L} \simeq E_q[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x})] = -\mathcal{F} \tag{4.5}$$

Let us further assume that the samples x depend only on the most recent z. In such case, the first term averaged by q over all possible z values approximately corresponds to negative of marginal entropy of the data x, $-H(x_n)$. The second term captures the entropy of z that contains the residual information in the measurements, similar to information that is captured by the entropy rate of x as $H(x_n|x_1^{n-1})$ for asymptotically large n. Under such assumptions $-\mathcal{F}$ is similar to the IR expression given in Equation 5.1. Accordingly, finding the minimum of $\mathcal{L}$ is equivalent to maximizing $\mathcal{F}$, which in case of our time signal assumptions[1] approximately equals to IR.

4.3 Methodology

4.3.1 Feature Learning with CNN

We adopt a CNN in order to learn a better representation of polyphonic music by treating the input as a 2D binary feature map. This is predicated on the notion that the arrangement of notes in a musical piece yields salient spatial relationships when visualized in a form such as a piano-roll and thus are conducive to being modeled by a CNN. In this, the input MIDI is first preprocessed into a piano-roll, with the beat resolution set to 8th notes. This gives us a feature map representation $\mathbf{x}^{(t)} \in \{0,1\}^{n \times r \times 1}$ at time step t, where n is a number of time steps in a frame and r is the note range. The piano-roll is then processed by a CNN with two convolutional layers separated by max-pooling layers and a final flattening layer. The output of this network is the latent feature $\mathbf{m}_l^{(t)} \in \mathbb{R}^k$ at time step t (See Panel A in Figure 4.2).

[1] It is worth noting that we are assuming here that the entropy of the latent states is equal to entropy rate of the data.

4.3.2 Encoder & Decoder Network

The models presented in Figure 4.2 can generate a track of music bar by bar, with a possibly polyphonic structure among several bars. We adopt a recurrent architecture for our VAE, which includes an RNN encoder and RNN decoder (See Panel B and C). The encoder RNN takes the latent feature $\mathbf{m}_l^{(t)}$ at each time step and produces a final hidden state $\mathbf{h}_q^{(T)} \in \mathbb{R}^e$ for a sequence of T MIDI frames.

$$\mathbf{h}_q^{(T)} = f_{\text{RNN}}(\mathbf{m}_l^{(1)}, ..., \mathbf{m}_l^{(T)}) \tag{4.6}$$

The hidden state is then subject to two linear transformations to determine the mean and standard deviation of the noise distribution given in Equation 4.1.

$$\begin{aligned} \boldsymbol{\mu} &= \mathbf{W}_\mu \mathbf{h}_q^{(T)} + \mathbf{b}_\mu \\ \boldsymbol{\sigma} &= \mathbf{W}_\sigma \mathbf{h}_q^{(T)} + \mathbf{b}_\sigma \end{aligned} \tag{4.7}$$

Where $\mathbf{W}_\mu, \mathbf{W}_\sigma \in \mathbb{R}^{z \times e}$ and $\mathbf{b}_\mu, \mathbf{b}_\sigma \in \mathbb{R}^z$. Noise is then generated as in Equation 4.2. Since we are modeling sequential data, the decoder network is trained to predict $p_\theta(\mathbf{x}^{(t)}|\mathbf{x}^{(1:t-1)}, \mathbf{z})$. In this, the RNN takes in the generated noise $\mathbf{z}$ at the first time step. At each subsequent time step, the latent feature $\mathbf{m}_l^{(t)}$ for an input sample $\mathbf{x}^{(t)}$ is linearly transformed into the same dimensionality as the noise source and then passed into the RNN.

$$\mathbf{m}_z^{(t)} = \mathbf{W}_z \mathbf{m}_l^{(t)} + \mathbf{b}_z \tag{4.8}$$

Where $\mathbf{W}_z \in \mathbb{R}^{z \times k}$ and $\mathbf{b}_z \in \mathbb{R}^z$. The RNN produces a hidden state $\mathbf{h}_p^{(t)}$ at each time step, which is passed through a logistic layer estimating $p_\theta(\mathbf{x}^{(t)}|\mathbf{x}^{(1:t-1)}, \mathbf{z})$.

$$\mathbf{h}_p^{(t)} = f_{\text{RNN}}(\mathbf{z}, \mathbf{m}_z^{(1)}, ..., \mathbf{m}_z^{(t)}) \tag{4.9}$$

$$\tilde{\mathbf{x}}^{(t)} = \sigma(\mathbf{W}_p \mathbf{h}_p^{(t)} + \mathbf{b}_p) \tag{4.10}$$

Where $\sigma(\cdot)$ is the logistic sigmoid function, $\mathbf{h}_p^{(t)} \in \mathbb{R}^d$, and $\mathbf{W}_p \in \mathbb{R}^{nr \times d}$. This effectively yields a binary feature map of the same dimensionality as the input which is used to predict a piano-roll based on the input at the previous time steps and the noise. Finally, we use the Gated Recurrent Unit (GRU) [CvM+14] for both the encoder and decoder RNN, which is defined by the following equations for $f_{\text{RNN}}(\mathbf{x}^{(1)}, ..., \mathbf{x}^{(t)})$ at time step t.

$$\mathbf{s}^{(t)} = \sigma_g(\mathbf{W}_s \mathbf{x}^{(t)} + \mathbf{U}_s \mathbf{h}^{(t-1)} + \mathbf{b}_s)$$

$$\mathbf{r}^{(t)} = \sigma_g(\mathbf{W}_r \mathbf{x}^{(t)} + \mathbf{U}_r \mathbf{h}^{(t-1)} + \mathbf{b}_r)$$

$$\mathbf{h}^{(t)} = \mathbf{s}^{(t)} \odot \mathbf{h}^{(t-1)} + (1 - \mathbf{s}^{(t)}) \odot \sigma_h(\mathbf{W}_h \mathbf{x}^{(t)} + \mathbf{U}_h(\mathbf{r}^{(t)} \odot \mathbf{h}^{(t-1)}) + \mathbf{b}_r)$$

Here, $\sigma_g(\cdot)$ denotes the logistic sigmoid function and $\sigma_h(\cdot)$ denotes hyperbolic tangent. In our implementation, we use 256 hidden units for the encoder and 512 hidden units for the decoder. With GRU, the model can create sequential output combined with decoded noise and previous output utilized for next input. As shown in Figure 4.2, the model sequentially generates bars one after another based on VAE structure, which takes inputs of mean and variance, then proceeding to next step which processes a random noise z and output received by the previous GRU.

4.3.3 Training Details

We train the network on training MIDI files, segmenting the MIDI input into batches of 8 bars, half a bar to a time step, and 8th note as the note resolution, resulting in 16 time steps. In each epoch, we train on the entire song with non-overlapping batches. We use dropout as a regularizer on the output of the CNN and the output of the decoder RNN. For optimization, we use the Adam optimizer with a learning rate of 0.001. In addition, we clip the gradients of the weight matrices so the L2 norms are less than 10. The loss of our network is that of Equation 4.3, with the log loss in the second term being cross entropy loss between the input samples and the output of the decoder. The model generally converges around 200 epochs. By enabling loss function calculation automatically, we observe and measure the model by the cost function. During training, our model is to focus the posterior of probability by training network to process the mean and variance of this posterior. In the aspects of variational inference, as the learning is repeated, the difference is minimized (Equation 4.3).

4.4 Experiments

To evaluate the structural quality of the musical result, we compare our model with the MelodyRNN model [WERA]. The MelodyRNN model is designed in several different ways (basic RNN, lookback RNN, attention RNN, and polyphony RNN), and we chose attention RNN and polyphony RNN model, which allow the model to capture longer dependencies, and result in melodies that involve arching themes [Mel]. Specifically, polyphony RNN aims at polyphonic music generation, so it is an appropriate baseline to compare with our model. For our experiments, our training data comes from the Nottingham Dataset, a collection of 1200 folk songs [not]. Each training song is segmented into frames (piano-roll), and for the preprocessing of our dataset, we implement our method based on the `music21,librosa`, and `pretty_midi` packages for feature extraction on MIDI file [CA10, MRL$^+$15, RE]. We use an input of 128 binary visible units and aligned on the 8th note beat level. With these data, we train each model of MelodyRNN and our proposed network to create MIDI sequences. Both our proposed model and MelodyRNN model converge around 200 epochs. Our implementation is now available on github[2].

[2] https://github.com/skokoh/c_vrnn_mmsp_2018

Table 4.1: Total IR Results (Averaged scores)

Melodies	Total IR (8 bars)	Total IR (16 bars)	Total IR (32 bars)
Nottingham Original [not]	4974.61	7412.91	18567.01
Proposed	3463.81	6047.28	16044.91
PolyphonyRNN [WERA]	3023.44	6027.04	15425.27
AttentionRNN [WERA]	3381.71	5712.87	14192.60
MidiNet [YCY17]	3117.68	-	-
Time (s)	15.3	29.2	67

4.4.1 Model comparison

After training, we compare generated samples from each of 3 models (proposed, polyphony RNN, attention RNN) for each of 3 settings in generated sample duration of 8 bars, 16 bars, and 32 bars. We use IR from VMO [WD15] as a basis of comparison and each generated MIDI sample was synthesized to audio signal. For comparison, we extracted 30 unique generated songs from each setting (See Table 4.1), thus 273 individual sample songs are tested for evaluation[3]. In the case of MidiNet, only 3 different testing samples are available within 8 bar length of audio sample. We want to see the variation in Total IR value which could be affected by the length of song in structural analysis. We report an averaged value of IR in Table 4.1.

We empirically analyze our model in several settings against MelodyRNN model. We share key observations:

• Table 4.1 shows average IRs for original Nottingham MIDI datasets and for generated samples from several models, where higher IRs report more distinct self-similarity structures. The IR of the original dataset is higher than that of the generated music. Self-similarity in audio refers to the multi-scalar feature in a set of relationships, and it commonly indicates musical coherence and consistency [Foo99].

• In Table 4.1, polyphony RNN and attention RNN models present lower IR than our proposed model does. Results in each setting show that the convolutional recurrent latent variable

[3] 3 audio samples (8 bars length only) are generated by MidiNet Model, which are uploaded on `https://github.com/RichardYang40148/MidiNet/tree/master/v1/`

sampling approach increases the IR of the produced musical material over other neural network approaches, indicating a higher degree of structure. Accordingly, the results manifest our proposed model can generate higher level of musically consistency structure.

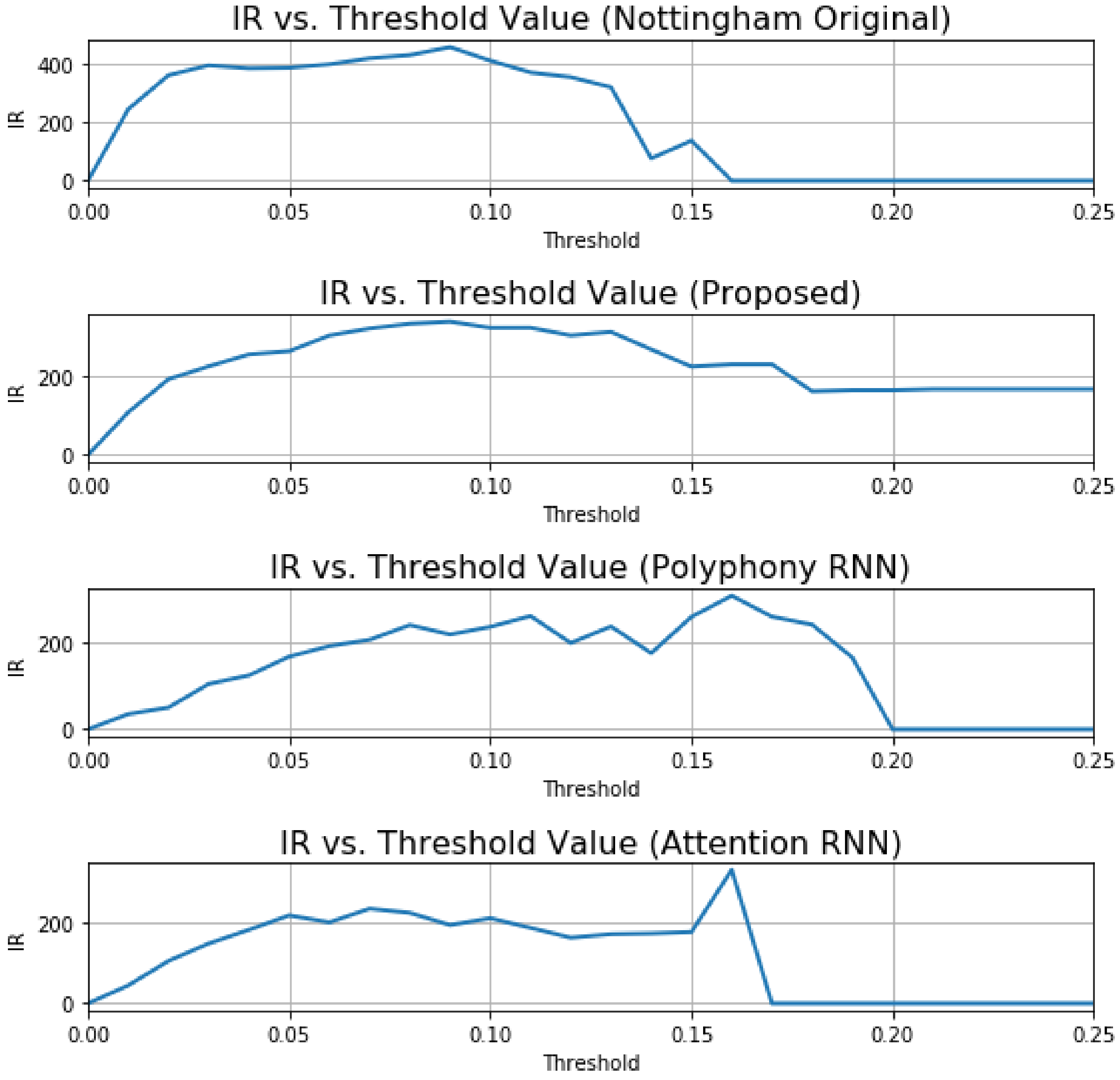


Figure 4.3: Total IR vs. Threshold θ value (VMO)

• In Figure 4.3, the visualizations of the IR values versus different θ on one song are represented. From top to bottom, we share the results of the sample songs from each setting, training dataset, our proposed model, polyphony RNN, and attention RNN model. The results show the relation between IR and threshold value and implies different musical structures are generated by different θ values. In terms of the results graphs, the attention RNN recopies longer segments, but they are interrupted, which is dropped down in figure, while our proposed model relies on shorter previous patterns, but the transitions are smoother thus the blocks are longer.

• In Figure 4.4, the results for finding repeated patterns in one of the audio samples generated from each setting are displayed from top to bottom. The y-axis indicates the pattern index of repeated motifs of a signal sampled at discrete times shown along the x-axis. The lines represent repeated motifs, which are longer and fewer in the RNN case. In the graph from Nottingham Original, we can recognize that the original has many more shorter musical pattern indexes appearing at multiple frame numbers. The overall distribution of repeated themes seems to be captured better in the outputs of our proposed approach, suggesting that it captures some structural aspects of patterns distribution of the data as well.

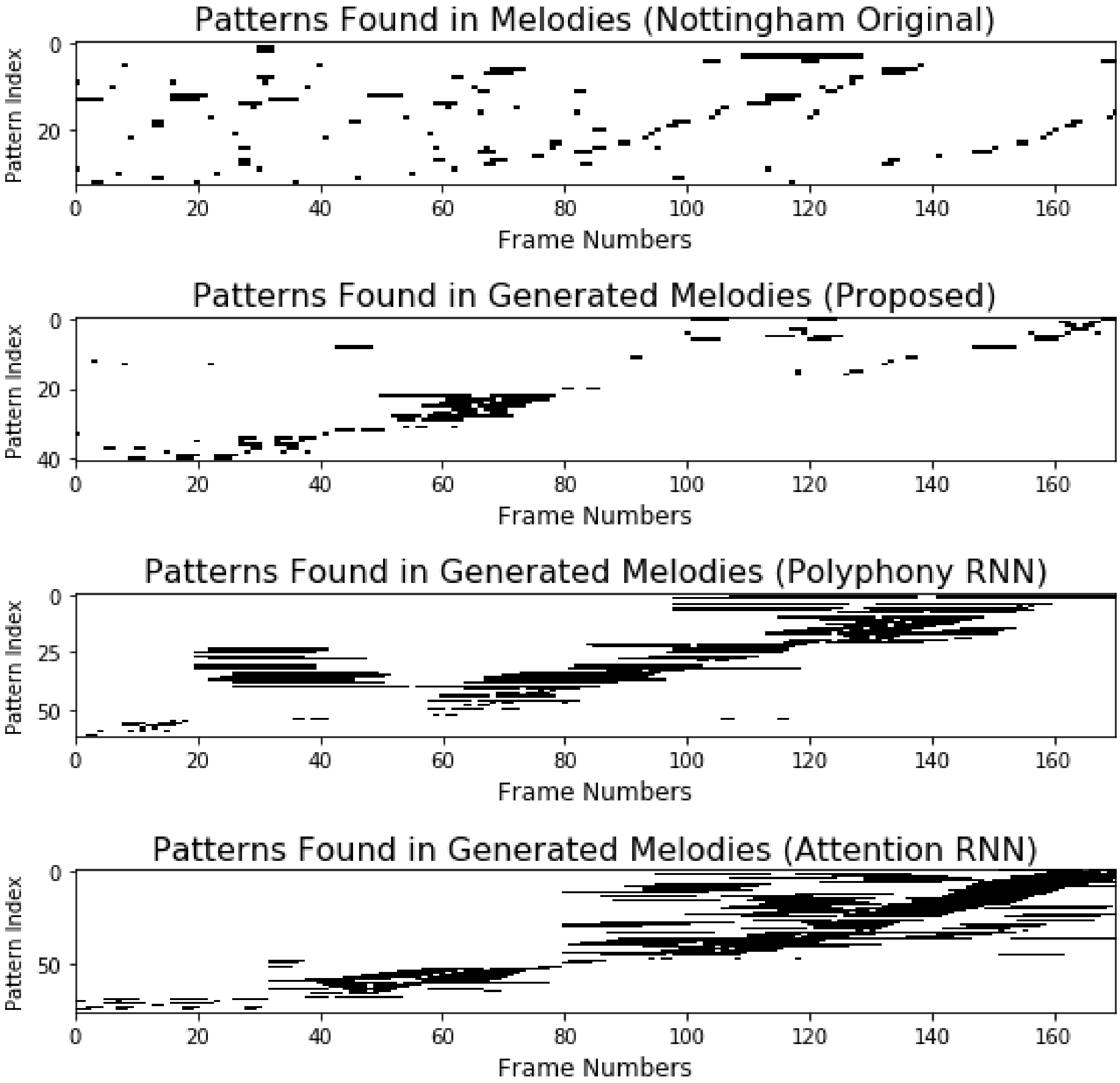
4.4.2 Application

In this section, we share our generated melodies in terms of following the research question: can we build a model capable of learning long-term structure and capable of including the method to generate polyphonic music pieces?

Considering an application level, we explore video game music generation and emulate a specific song from music samples for creating a new sequence of music (See Figure 4.5). By doing this, we use 10 different MIDI files derived from a corpus of Video Game music[4] and we generate 10 unique MIDI outputs based on each training sample. The MIDI files are mainly composed of 4-5 different instruments with multi-tracks. From this approach, our model copies

[4] https://www.vgmusic.com

the theme from previous music sample and mimics the style of music with a new sequence.



Patterns Found in Melodies (Nottingham Original)

Patterns Found in Generated Melodies (Proposed)

Patterns Found in Generated Melodies (Polyphony RNN)

Patterns Found in Generated Melodies (Attention RNN)

Figure 4.4: Pattern Findings with VMO

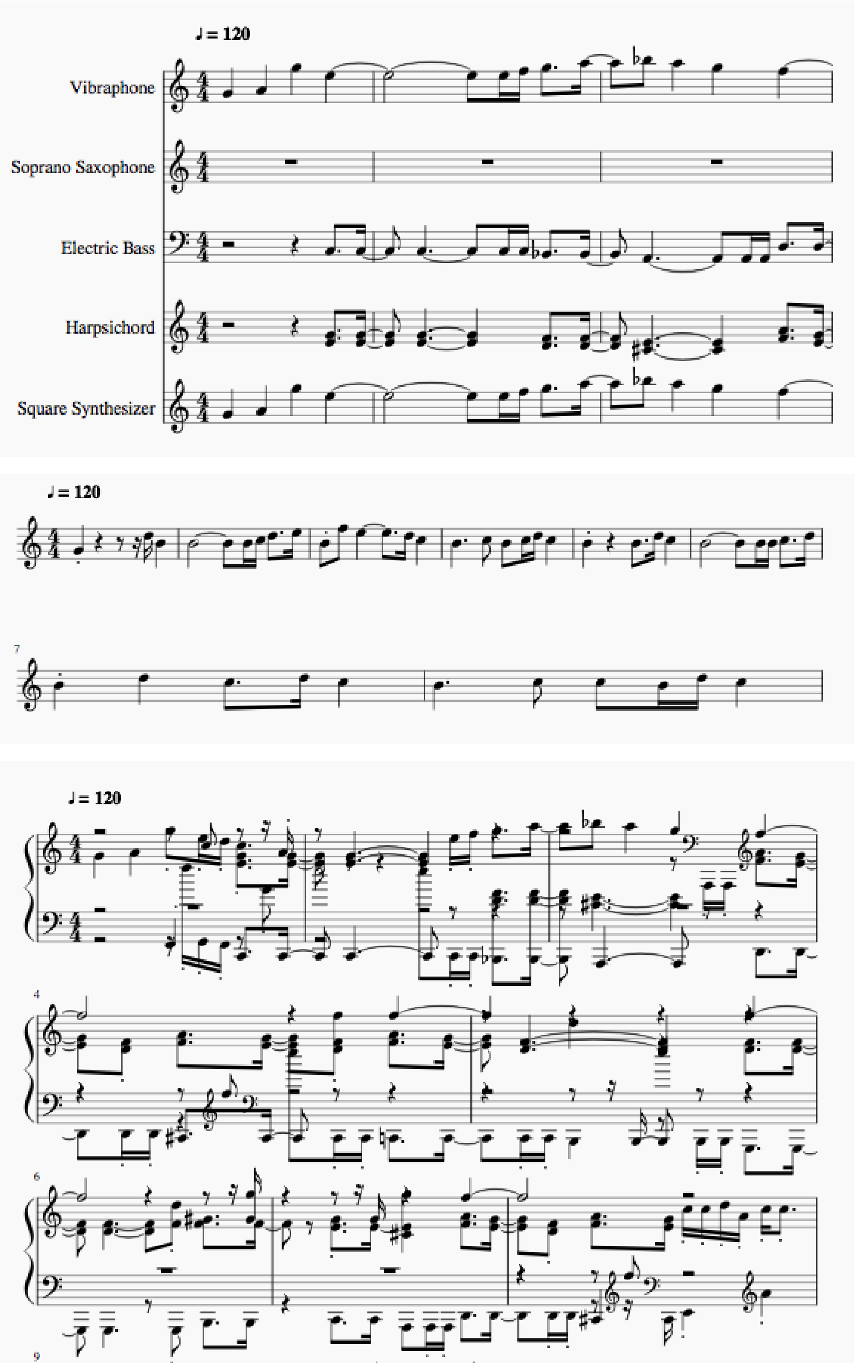


Figure 4.5: Examples from the Video Game music sample and generated results from attention RNN and proposed method. From top to bottom: original sonicstarlightzone.mid, attention RNN result, proposed method result.

In Figure 4.5, the results indicate that our proposed model can generate music beyond monophonic melodies for various types of music, depending on the input data. The result of the attention RNN differs in our model and in the complexity of the results, since attention RNN model covers simple melody generation/progression and repetitive patterns appearing in the generation results. Our generated melodies shows that we can create long-term structure of music and can compose complex sequence of music while including the original theme. Moreover, our proposed model can process training samples from a prior distribution and generate the sequence more dynamically. Our sample results for video game music are also posted on soundcloud[5].

4.5 Discussion

In this study, we show initial proof that our proposed model applied to MIDI sequence representations can capture the structure of the song and create polyphonic music. The motivations behind combining CNN, RNN and VAE were to explore significant problems in music generation which are related to representation issues that are handled via CNN, repetitive patterns in generated output that are known in RNN and ability to generate variations from progression of melody sequence. In our study, we used IR as a critera to evaluate the generated output and compare it to other models.

From the quantitative evaluation, the results show that the latent variable sampling approach substantially increases the IR of the generated musical material over other neural network approaches, implying a higher degree of semantic structure. At the application stage of our method, we introduce the model to emulate a specific song from a video game and generate background music similar in style to those examples. Some musical applications need to work with fewer samples in order to generate a specific musical result and our Convolutional-Variational Recurrent Neural Network would be flexible about the size of dataset.

[5] https://soundcloud.com/user-431911640/sets

In addition to VAE utilized in this paper, other generative models have been actively challenged in different ways for music generation purpose. Given the recent enthusiasm in deep learning with music, we also practice introducing combined neural network models and data representations that effectively process the melodic polyphonic harmonic structure in music.

Chapter 4 is adapted from published material in ”Rethinking recurrent latent variable model for music composition”. Koh, Eunjeong, Dubnov, Shlomo, and Wright, Dustin. IEEE International Workshop on Multimedia Signal Processing, 2018. The dissertation/thesis author was the primary researcher and author of the paper.

Chapter 5

Information Dynamics in Machine Generated Music

This paper reports on a study that analyzes the structure of music created by neural networks. We tackle an important problem of the evaluation and interpretation of machine learning models with a focus on music models. To quantitatively evaluate the quality of music, we use Information Rate which can compare how information flows in original music versus in music generated by an artificial neural network that learned that music. We review mostly known but often underappreciated properties relating to the evaluation and interpretation of machine learning models with a focus on music models. We aim to study the perspectives from statistics and information theory as to how creativity can be measured in both a computationally and musically meaningful way. Information Rate can measure mutual information between past and present in music and can be used to find repeated motifs and the main theme in music which are the basis of the composition. This approach represents the attempt at defining evaluation metrics for automatic music generation by neural networks which possess measure-level musical structure in terms of novelty.

5.1 Introduction

In terms of evaluation of machine-generated music, a lot of qualitative evaluation is conducted through user studies in general automatic music composition researches. In these experiments, they evaluated the quality of music by asking the participants about their opinions regarding generated music. On the other hand, statistical approaches are used for quantitative evaluation, such as log-likelihood or prediction accuracy. In this study, we analyze the musical results generated by machine learning and the quality of the corresponding music. To enable a quantitative analysis of musical structure, we apply the Information Rate (IR) from Variational Markov Oracle (VMO) study. Using IR, we are able to understand the structural properties of music, which is important for understanding the relation between musical organization and perception, as well as for automatic music composition. For example, using IR, we can assess the effect of repetition versus variation structure constraints and higher Information Rates indicate more distinct self-similarity structures. This also introduces the problems in modeling the content-invariant self-similarity property as well as some sampling noise which increases predictive entropy with lowering creativity.

5.2 Methodology

Rather than understanding the distance between existing music data and generated results, we consider the internal working of neural network models and generated musical sequences in terms of IR by VMO. VMO allows measuring music information dynamics where higher IR values capture longer structural note transitions in generated music compared to ones with lower IR values [WD15].

$$IR(\mathbf{x}_1^{n-1}, \mathbf{x}_n) = H(\mathbf{x}_n) - H(\mathbf{x}_n|\mathbf{x}_1^{n-1}) \tag{5.1}$$

Analyzing musical structure in terms of IR, we can compare how information flows in original music versus in music generated by an artificial neural network that learned that music. In order to do so, we generalize the notion of IR to include an additional variable that represents the state of the neural network. In Equation 5.1, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ denotes time series $\mathbf{x}$ with N observations, and $\mathbf{H}(x)$ denotes the entropy of $\mathbf{x}$. As a result, IR denotes corresponding information between the current and previous observations, which enables the understanding of variation and repetition in a song segment. The IR analysis using VMO provides an independent evaluation of the structure of song as captured by the sequence of audio chroma features. Furthermore, it presents a detailed motif analysis of the data and provides a qualitative discussion of generated musical samples.

5.3 Experiment

We compare the generated music pieces from some recent generative neural network models [1]. The experiment shows average IRs for the original MIDI training dataset and for generated samples from the models, Boulanger et al [BLBV12] and Koh et al [KDW18], where higher IRs report more distinct self-similarity structures. Self-similarity in audio refers to multi-scalar features in a set of relationships, and it commonly indicates musical coherence and consistency. Results in each setting show that the latent variable sampling approach increases the IR of the produced musical material over RNN-based approaches, indicating a higher degree of structure. In addition, the distribution in each group shows that the result of [KDW18] has more distribution than the training dataset or another model. This is because the research focuses on the creation of a diverse and complex new sequence of music.

For comparing music quality between the groups, we set some evaluation rubrics of understanding music quality based on IR: (1) visualized motifs based on IR by VMO, (2) number of motifs and their lengths, and (3) how IR changes over time in generative models. VMO analysis

[1] Sound example:
https://soundcloud.com/user-431911640/sets/model-comparisontraining-data-whole-nottingham-dataset

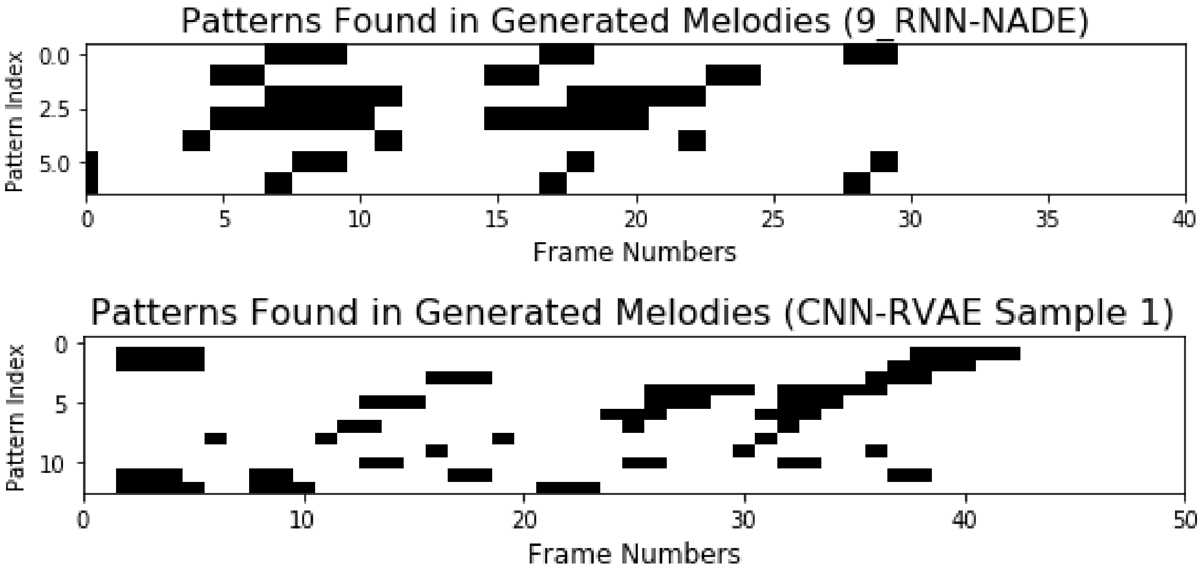


Figure 5.1: (Top) VMO Pattern Findings with RNN-RBM [BLBV12], (Bottom) VMO Pattern Findings with CNN-Recurrent VAE [KDW18]

is able to detect the motifs themselves as repeated patterns of notes since it finds repetitions using approximately matching suffix search (See Figure 5.1). The motifs over time (and sub-motifs when the lines overlap vertically) allows visual inspection of such structures. The higher level repetitions also exist in the arrangement of motifs themselves. Since each VMO analysis optimizes the threshold of similarity for approximate suffix matching, the motifs shown in the different figures appear slightly different. Also, the VMO takes into consideration also the later motif structure and adjusts its sensitivity so as to produce the most informative representation of the overall information in each piece. The results show the relation between IR and threshold value and imply different musical structures are generated by different θ values. In terms of the results graphs, the RNN-RBM recopies longer segments, but they are interrupted, while the CNN-Recurrent VAE model relies on shorter previous patterns. The CNN-Recurrent VAE uses a latent variable model to design the graph, so we can see that the chord progression of the music is more variously than iterative or repetitive structure.

Chapter 5 is adapted from published material in ”Information Dynamics in Machine

Generated Music". Koh, Eunjeong and Dubnov, Shlomo. SoCal Machine Learning Symposium, 2018. The dissertation/thesis author was the primary researcher and author of the paper.

Part IV.

Music and Emotion

Chapter 6

Mugeetion: Musical Interface Using Facial Gesture and Emotion

People feel emotions when listening to music. However, emotions are not tangible objects that can be exploited in the music composition process as they are difficult to capture and quantify in algorithms. We present a novel musical interface, Mugeetion, designed to capture occurring instances of emotional states from users' facial gestures and relay that data to associated musical features. Mugeetion can translate qualitative data of emotional states into quantitative data, which can be utilized in the sound generation process. We also presented and tested this work in the exhibition of sound installation, Hearing Seascape, using the audiences' facial expressions. Audiences heard changes in the background sound based on their emotional state. The process contributes multiple research areas, such as gesture tracking systems, emotion-sound modeling, and the connection between sound and facial gesture.

6.1 Introduction

Electronic music researchers use various components as inputs for their music generation process [PLF^{+}01, Lyo17, LT01, Çam12]. Music and emotion are strongly linked, and listeners can feel different emotions directly or indirectly through music. Engaging emotion as a component of a musical interface has great potential for composing creative music and expressing messages in an effective way [VOC09]. However, there are several difficulties in using emotion for sonification [Lem08, WW13]. First, emotion is qualitative and thus hard to utilize for sound generation applications, which rely on quantitative inputs. Second, emotion is represented on a continuous spectrum. Measurement of affect requires a complex and multi-faceted approach. In this paper, we use a facial gesture tracking system to define emotional states based on facial gesture information.

Facial gestures express various information related to emotion, cognition, and inspiration [DTM14, KCT00]. Further, facial gestures are more straightforward indicators of emotion than other bodily gestures. There are several studies related to the connection between facial gestures and sound itself [dAD13, McD16]. In this paper, we propose an interactive audio interface that sonifies emotion. The idea is to use facial gesture data to detect emotion and categorize these into several emotional states for sonification. We implement two approaches for this prototype: (1) music style transition based on user's emotion and (2) auditory interface based on the connection between facial components and musical metadata. We also installed our system in a digital exhibition for facial interaction with an audience at the *Hearing Seascape* installation. During the exhibition, Mugeetion detected audience's facial expression in real-time and audience were able to hear the sounds simultaneously which was mapped with their specific facial gestures.

6.2 Related Work

There has been a rich history of creating novel sound interfaces using gesture-based motion-tracking for live performance and improvisation [KTPLW14, Jen12, CPL11]. A motion tracking system can allow a musician to generate their own creative music in real-time [WB98, LXC$^+$15]. Previous studies demonstrated interesting new audio interfaces for sonification through body gesture. A number of systems have looked at capturing gestures and utilizing gesture data for the sonification process either stepwise or in real-time [CPL11, MSE14]. Regarding previous studies, there are two approaches, which have used sound as an input for tracking facial gestures or facial components as input for sound generation. Some studies utilized auditory input for focusing on the visualization of facial gesture [Kra00, SCT04, Haw02]. For example, Kapuscinski [Kap10] conducted listening tests of Chopin pieces and recorded facial expressions from the participants. Other experiments have focused on sound generation using facial parameters as an input [Art28]. These studies use FaceOSC software to apply facial gesture data to the sound generation process. McDonald [McD16] created FaceOSC software to track facial gestures directly to Max as input. There are several interesting experiments linking facial gestures and sound on Youtube [Art28]. However, these experiments are more targeted toward application, rather than music cognition research. Few computer music researchers delve into the relationship between emotion and sound itself.

Music psychologists have studied the relationship between emotion and sound, and tried to model its connection. However, music cognition research has not contributed to music sonification research. In this paper, we propose a musical interface with the facial gesture tracking system and Facial Action Coding System (FACS) [DTM14] in order to capture emotional states. FACS can allow a concrete data representation of the facial gesture and its corresponding emotional state. Thus, facial gestures can be reference points for observing emotion and translating emotion into sound. In this context, we use the tracked facial data to the sonification process. We will discuss its musical implementation in the following sections.

6.3 Methods

6.3.1 Understanding Facial Gesture

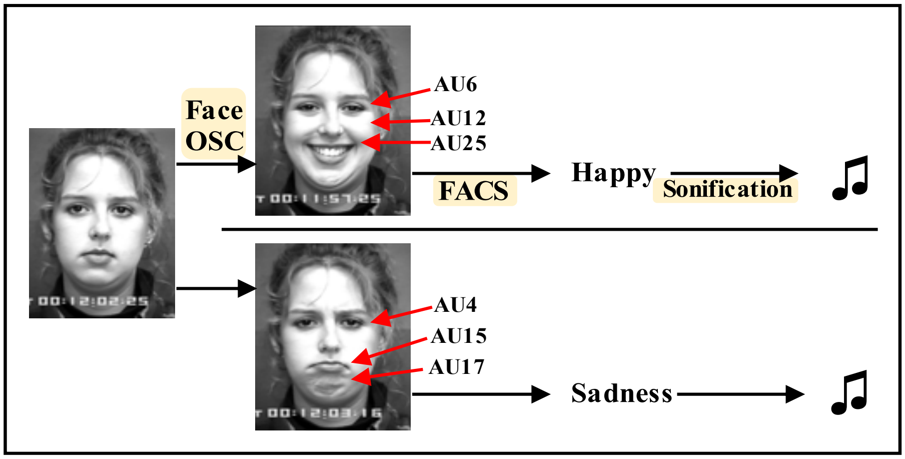


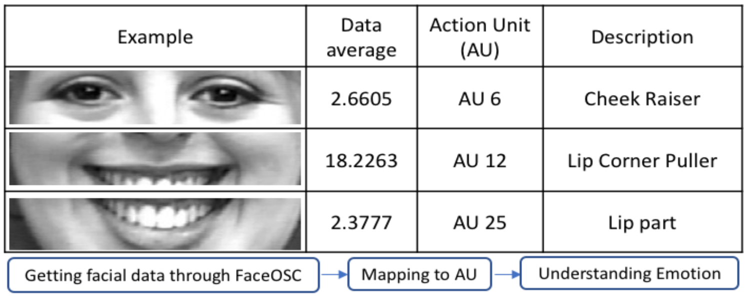
Figure 6.1: System structure: connection between facial gesture, compound facial expressions of emotion, and sound

Figure 6.1 gives an overview of integrating the proposed system to connect facial expression to sound generation[1]. We generated the musical style based on facial expressions. In Figure 6.1, our system includes three sequential steps: (1) capturing facial gesture using FaceOSC, (2) connecting to compound facial expressions of emotion, and (3) synthesizing musical features based on the emotional state. FaceOSC software is used to help the Mugeetion system understand the user's facial gestures and generate sound based on the user's emotional state. The emotion detection module uses the software for real-time facial gesture tracking and transmits raw-level facial data over the Open Sound Control (OSC) protocol. If the detector finds multiple potential faces within the frame, the closest face will get the priority of recognition, analyzing a single face

[1] In Figure 1 and 2, printed images are copyrighted by ©Jeffrey Cohn, which come from Cohn-Kanade (Ck & CK+) database. `http://www.consortium.ri.cmu.edu/ckagree/`

at a time. For analyzing facial expression, we use the FACS and Action Unit classification[2]. We chose the Action Unit (AU) combinations of three basic emotions: (A) happy, (B) neutral, and (C) sad. Each emotional state is combined with several individual AUs. For example, the facial expression of happy includes AU 6 (cheek raiser), AU 12 (lip corner puller), and AU 25 (lips part) (See Figure 6.1).

We practiced our sonification method with face images from The Cohn-Kanade AU-Coded Facial Expression Data-base [KCT00, LCK+10]. By training with multiple images, we made the system work well with different faces. We selected representative facial images for linking with our sound generation process. We used 20 images for each emotional state: happy, neutral, sad (60 images total). We measured these data to create a data range for each emotional state and defined the differences between each emotion. We manually annotated the range of facial gestures for mapping each muscle activation to AU components (See Table 6.1)[3]. Figure 6.2 shows the data range for AU components with our training images. For example, the average AU6 scale of the happy face is 2.6605, AU12 is 18.2263, and AU25 is 2.3777. After learning the range of AUs, the system can classify facial gestures to pre-defined states with each individual photo or real-time face input through the connected web-cam. Along with categorizing, the system attempts to translate the musical style based on the input of emotional states (See Figure 6.2).



Example	Data average	Action Unit (AU)	Description
	2.6605	AU 6	Cheek Raiser
	18.2263	AU 12	Lip Corner Puller
	2.3777	AU 25	Lip part

Getting facial data through FaceOSC
Mapping to AU
Understanding Emotion

Figure 6.2: Data transition process: from facial gesture to emotion

[2] Description of Facial Action Coding System and Action Units `https://www.cs.cmu.edu/~face/facs.htm`

[3] The unit in this table is followed by FaceOSC data measurement.

Table 6.1: Facial data configuration from FaceOSC

position	details	data range (min/max)
mouth	width	6.0244/19.2747
	height	0.8893/3.0010
eyebrow	left	6.7666/8.0714
	right	6.6787/7.9785
eye	left	2.4329/3.4357
	right	2.3950/3.3144
jaw	-	18.9888/22.9718
nostrils	-	5.6477/8.8061

6.3.2 Sonification with Action Units

In this section, we focus on sonification with AUs in detail. We generate musical output based on the connection between AUs and emotional state. We then apply the formula between emotion and sound features, such as how the energetic happy face is mapped to the pitch/loudness increasing, and the dynamics in the sad face are mapped to white noise/distortion parameters. We also connect specific AUs to MIDI notes for sonification. The MIDI packets are mapped to controls of different parameters, resulting in different musical sounds based on how the emotional state moves. For example, when a user moves their mouth, the mouth height data is inputted and we normalize the data between 0-127 scales for generating MIDI notes or dynamics. Then, these 0-127 scales correspond to MIDI note scales. There are a few studies have explored this method before [dAD13, MSE14, SCT04], and we explore the linkage between other sound features and the emotion conveyed in the AUs.

6.3.3 Connecting between Emotion and Sound

In this section, we explain how the system has been implemented for connecting emotion to sound. The system can interpolate the sound results from facial gesture inputs. In this approach, we generate the sound based on pre-recorded sound. We can play different sounds based on the user's happy, neutral, or sad emotional state. Our Mugeetion interface automatically plays the specific song related to the user's emotional state. We list our sound files below, which have been played to a number of subjects interacting with the system.

· Happy

Mozart - The Piano Sonata No 16 in C major

Mozart - Eine Kleine Nachtmusik K 525 Allegro

· Neutral

Mozart - Piano Sonata No 11 in A major K 331

· Sad

Mozart - Symphony No 25 in G Minor K 183

1st Movement

Mozart - Requiem in D minor

The selection of the list is based on the study of the Mozart Effect [PCBCC13]. For the sound files, we use Piano-midi.de dataset[4].

6.4 Prototypes

6.4.1 Demo

For the prototype of our system, we explored adding more musical variation, such as pitch height, loudness, distortion, or tempo change, as parameters to be controlled. We show a possibility of sound generation in real-time. Our preliminary demo video is uploaded on Youtube[5]. In order to allow users to easily interact with their sound generation process, we built a Max application, which utilizes facial data and FAC for sound creation.

[4] http://www.piano-midi.de/mozart.htm

[5] https://www.youtube.com/playlist?list=PLjaQX_vKy2Jcv0r9wrc_yU2gbRhZh39GV

6.4.2 Sound Installation Work with Mugeetion

Our sonification method, Mugeetion, has also been used in the sound installation exhibition, *Hearing Seascape* (See Figure 3) at the Qualcomm Institute at UC San Diego in February 2018[6]. This exhibition was a part of a collaborative effort with the Scripps Institution of Oceanography at UC San Diego to interpret their coral reef image data in a musical way. To convey the importance of engaging in the soundscapes of coral reefs, we suggested that our Mugeetion would be effective in fulfilling the goal of the project. Our prototype of the exhibition can be found in Youtube[7]. The main goals of this project were to display different aspects of sound and innovative graphic design to create an enjoyable environment for the audience, and to create an inviting soundscape with a synergy among voices, images, synthesized sounds, and human emotion.

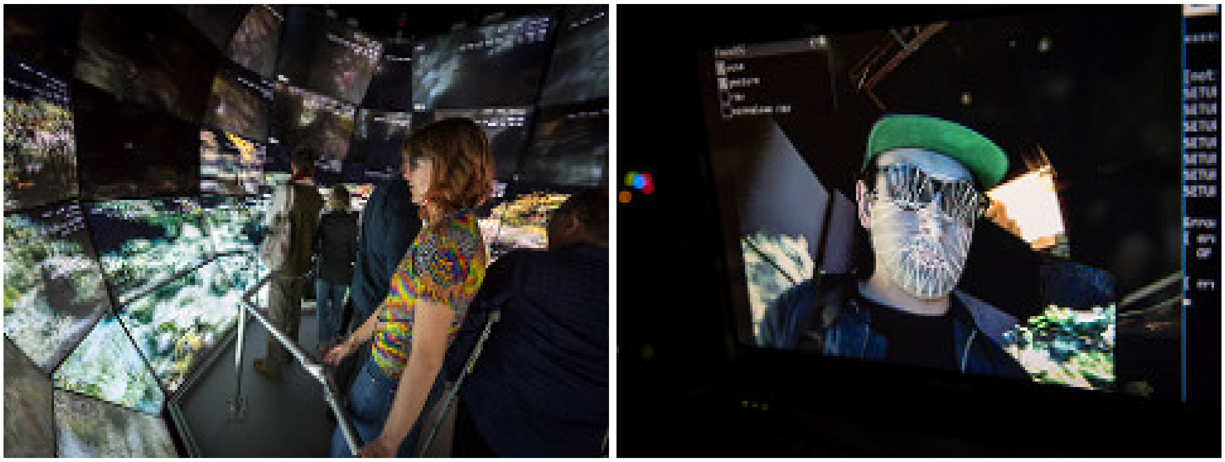


Figure 6.3: Left: *Hearing Seascape* exhibition, Right: Interaction with Mugeetion during the exhibition (Neutral state)

Characteristics of the sounds in the Hearing Seascape

There were two sound components in the sound installation. First, regarding sound input, we made recordings of singing and speaking in bowls of water. We recorded various sounds, such as giggling, clicking with tongue, singing, spoken dialogue, low/high pitches, both in the air and

[6] Photo by Alex Matthews ©2018 Regents of the University of California.

[7] `https://youtu.be/c-kHwnYuF44`

in the water. This specific sonification process was related to the goal of the project. The sound of voices underwater showed a variation of pitch and vagueness of speech. This is representative of the confusion and misunderstanding that surrounds coral reef research [SBC^{+}16]– there is so much yet to be discovered and understood about these creatures. Second, using Mugeetion, our method detected the audience's facial expression in real-time, and the detected emotional state was used to display of the coral reef images and synthesize the soundscape. The audience can hear the sound that is simultaneously mapped with their specific facial gestures. For instance, when audiences expressed strong emotions with their facial gestures, these dynamics connected to sound components to increase intensity, tempo, and pitch height. The interaction through Mugeetion invited the audience to participate in the exhibition.

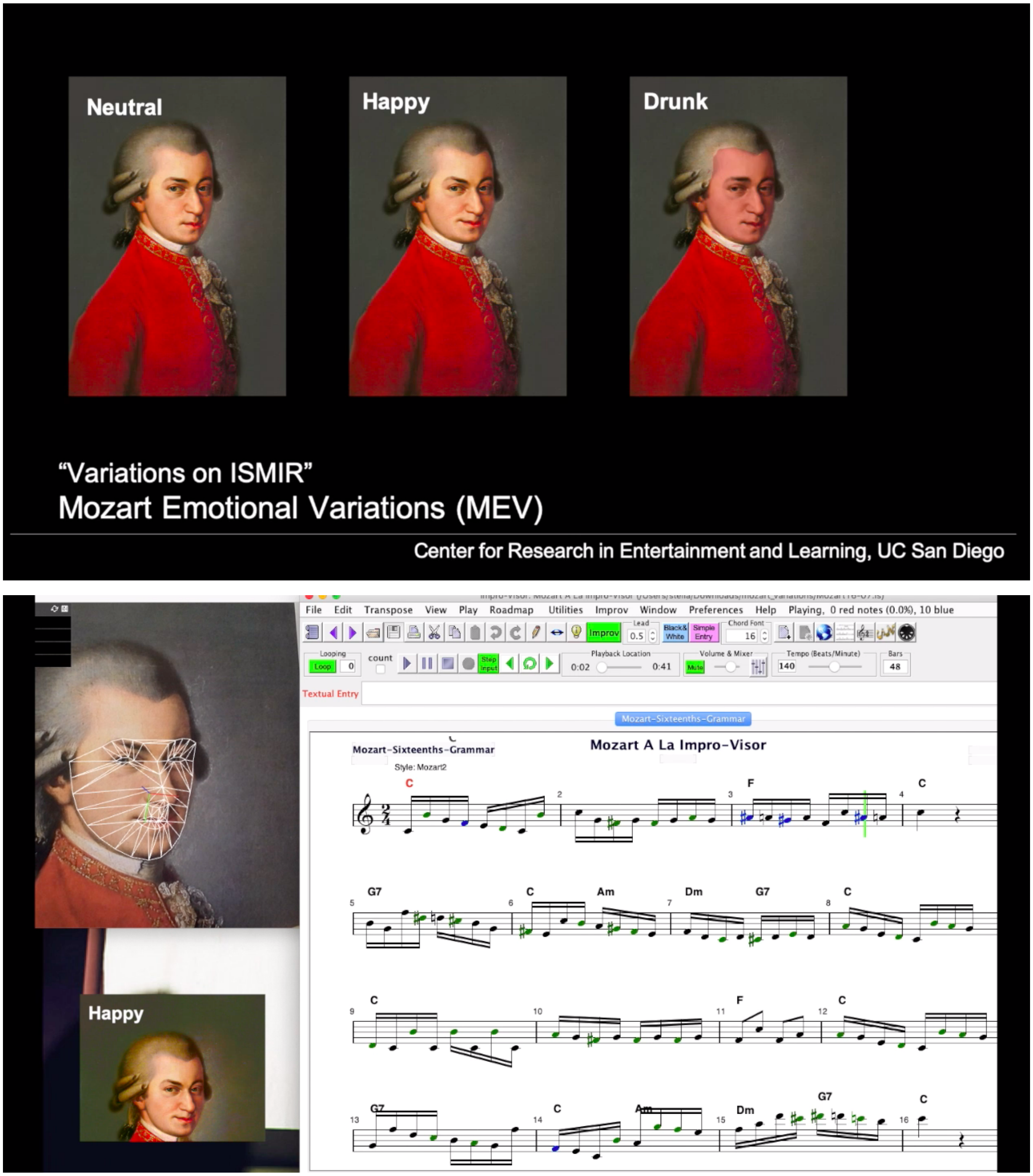
6.4.3 Application: Music Submission for ISMIR 2019

• **Title**: Mozart Emotional Variations (MEV)

• **Authors**: Eunjeong Koh, Robert M. Keller, and Shlomo Dubnov (Center for Research in Entertainment and Learning, UC San Diego)

Description of the content/concept/idea

Mozart felt different emotions when he performed different variations. In this video, we propose an interactive audio performance that sonifies emotion. The idea is to use facial gesture data to detect emotion and categorize these into several emotional states for improvisation. For doing this, we incorporated with two previous studies: Impro-Visor [Kel12] and Mugeetion [KY18]. Using Mugeetion, we captured occurring instances of emotional states from users' facial gestures and relayed that data to associated musical features. Next, we transcribed the Mozart Variations 1 and 3 to Impro-Visor's leadsheet notation, inferring chord symbols from the original by hand.



Neutral

Happy

Drunk

“Variations on ISMIR”

Mozart Emotional Variations (MEV)

Center for Research in Entertainment and Learning, UC San Diego

File Edit Transpose View Play Roadmap Utilities Improv Window Preferences Help Playing, 0 red notes (0.0%), 10 blue
Improv
Lead 0.5
Black& White
Simple Entry
Chord Font 16
Looping Loop 0
count
Step Input
Playback Location 0:02 0:41
Volume & Mixer Mute
Tempo (Beats/Minute) 140
Bars 48
Textual Entry
Mozart-Sixteenths-Grammar
Mozart-Sixteenths-Grammar
Mozart A La Impro-Visor
Style: Mozart2
C F C G7 C Am Dm G7 C C F C G7 C Am Dm G7 C
Happy

Figure 6.4: Snapshots of the Mozart Emotional Variations application where Impro-Visor and Mugeetion algorithms are embedded. (Top) Different Mozart’s faces showing emotional changes with different facial gestures and colors. (Bottom) Mozart’s face interacts with the Impro-Visor.

Then for each variation, Impro-Visor automatically learned grammars that imitate the corresponding melodic style. Each grammar was inspired by different emotional settings and we generated new melodies from these grammars. The backing harmony and bass lines were created from simple style files constructed by hand but exploited as the improvisations were generated in real-time. This process contributes to multiple research areas, such as gesture tracking systems, emotion-sound modeling, and new musical interface designs. Video material is available in here `https://youtu.be/LyEJ0IYEOHk`.

6.5 Conclusions

Mugeetion makes several contributions to previous work. Rather than simply detecting facial gesture data, it also automatically extracts emotional states and produces sound output transition. Mugeetion provides a sound generation model to users based on the components of emotion and musical metadata. We focus on how sound can be changed based on users' emotional movement. In the presented soundscape installation, the interaction between emotion and sound occurred based on user's emotional states. We explore how audience participation in artwork can be utilized in interactive systems and how it changes the sound generation output. In future work, we will collect continuous auditory feedback during the exhibition in order to evaluate the sound generation output. For example, audiences would be asked how satisfied they were with the reflection between sound output and their emotional states.

Furthermore, the system would be able to store a collection of data, which creators can use to improve their sonification process. Every AU per second and audio files would be automatically saved. The system would collect and store a repository of the memory units that users can look back on in order to re-utilize their composition process.

We will further develop the system based on the following issues:

· increasing training images for covering multiple faces and optimizing different emotional states

- implementing an AU indicator or other emotional measure on the FaceOSC display for better interaction with users
- exploring other similar emotion interactive system to compare the sonification result

6.6 Acknowledgements

This paper has been generously supported by Dr. Lei Liang, other members from Calit2 Center of Graphics, Visualization and Virtual Reality, and The Smith Lab in Scripps Institution of Oceanography at UC San Diego.

Chapter 6 is adapted from published material in ”Mugeetion: Musical interface using facial gesture and emotion”. Koh, Eunjeong and Yadegari, Shahrokh. International Computer Music Conference. 2018. The dissertation/thesis author was the primary researcher and author of the paper.

Chapter 7

Using Deep Audio Embeddings for Music Emotion Recognition

Emotion is a complicated notion present in music that is hard to capture even with fine-tuned feature engineering. In this chapter, we investigate the utility of state-of-the-art pre-trained deep audio embedding methods to be used in the Music Emotion Recognition (MER) task. Deep audio embedding methods allow us to efficiently capture the high dimensional features into a compact representation. We implement several multi-class classifiers with deep audio embeddings to predict emotion semantics in music. We investigate the effectiveness of L^3-Net and VGGish deep audio embedding methods for music emotion inference over four music datasets. The experiments with several classifiers on the task show that the deep audio embedding solutions can improve the performances of the previous baseline MER models. With this approach, we conclude that deep audio embeddings represent musical emotion semantics for the MER task without expert human engineering.

7.1 Music Emotion Recognition

It is an essential step for music indexing and recommendation tasks to understand emotional information in music. Previous MER studies explore sound components that can be used to analyze emotions such as duration, pitch, velocity, and melodic interval. Those representations are high-level acoustic features based on domain knowledge [WWL15, MGG18, CZX$^+$16, LCY13].

Relying on human expertise to design the acoustic features for pre-processing large amounts of new data is not always feasible. Furthermore, existing emotion-related features are often fine-tuned for the target dataset based on music domain expertise and are not generalizable across different datasets [PMP18b].

One of the goals of the MER task is to automatically recognize the emotional information conveyed in music [KSM$^+$10]. Although there are many studies in the MER field [SCS$^+$13, YC12, YDL18], it is a complex process to compare features and performances of the studies because of the technical differences in data representation, emotion labeling, and feature selection algorithm. In addition, different studies are difficult to reproduce as many of them use different public datasets or private datasets with small amounts of music clips and different levels of features.

7.2 Deep Audio Embeddings

Advancement in deep neural networks now allows us to learn useful domain-agnostic representations, known as deep audio embeddings, from raw audio input data with no human intervention. Furthermore, it has been reported that deep audio embeddings frequently outperform hand-crafted feature representations in other signal processing problems such as Sound Event Detection (SED) and video tagging task [Wil20, DCA].

The power of deep audio embeddings is to automatically identify predominant aspects in the data at scale. Specifically, the Mel-based Look, Listen, and Learn network (L^3-Net) embedding method recently matched state-of-the-art performance on the SED task [CWSB19]. Using a sufficient amount of training data (around 60M training samples) and carefully designed training choices, Cramer et al. were able to detect novel sound features in each audio clip using the L^3-Net audio embeddings [CWSB19]. Cramer et al. released their optimal pre-trained L^3-Net model which can now be extended to new tasks.

Previous studies have utilized neural networks to efficiently extract emotional information and analyze the salient semantics of the acoustic features. Recent works explore neural networks given the significant improvements over hand-crafted feature-based methods [Pic15, SB17, PS19, SZ14]. Specifically, using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) based models, several studies attempt to extract necessary parameters for emotion prediction and reduce the dimensionality of the corresponding emotional features [CLB$^+$20, THR19, DYZL19, LFH19]. After careful feature engineering, these methods are suitable for a target data set for emotion prediction, however, a considerable amount of training and optimization process is still required.

Deep audio embeddings are a type of audio features extracted by a neural network that take audio data as an input and compute features of the input audio. The advantages of deep audio embedding representations are that they summarize the high dimensional spectrograms into a compact representation. Using deep audio embedding representation, 1) information can be extracted without being limited to specific kinds of data, and 2) it can save time and resources.

Several studies have used deep audio embedding methods in music classification tasks. For example, Choi et al. implemented a convnet feature-based deep audio embedding and showed how it can be used in six different music tagging tasks such as dance genre classification, genre classification, speech/music classification, emotion prediction, vocal/non-vocal classification, and audio event classification [CFSC17]. Kim et al. proposed several statistical methods to understand deep audio embeddings for usage in learning tasks [KULH19]. However, there are currently no studies analyzing the use of deep audio embeddings in the MER task across multiple datasets.

Knowledge transfer is getting increased attention in the Music Information Retrieval (MIR) research as a method to enhance sound features. Recent MIR studies report considerable performance improvements in music analysis, indexing, and classification tasks by using cross-domain knowledge transfer [HE10, VdODS13]. For automatic emotion recognition in speech data, Feng and Chaspari used a Siamese neural network for optimizing pairwise differences between source and target data [FC20]. In the context of SED, where the goal is to detect different sound events in audio streams, Cramer et al. [CWSB19] propose a new audio analysis method, using deep audio embeddings, based on computer vision techniques. It remains to be seen if knowledge transfer can be successfully applied on deep audio embeddings from the SED domain to the MIR domain for the task of MER.

In this study, we use deep audio embedding methods designed for the SED task and apply it over four music emotion datasets for learning emotion features in music.

7.3 Methodology

7.3.1 Downstream Task: Music Emotion Recognition

We employ a two-step experimental approach (see Figure 7.1).

Step 1. Given a song as an input, a deep audio embedding model extracts the deep audio embeddings that indicate the acoustic features of the song.

Step 2. After extracting deep audio embeddings, the selected classification model predicts the corresponding emotion category that indicates the emotion label of the song.

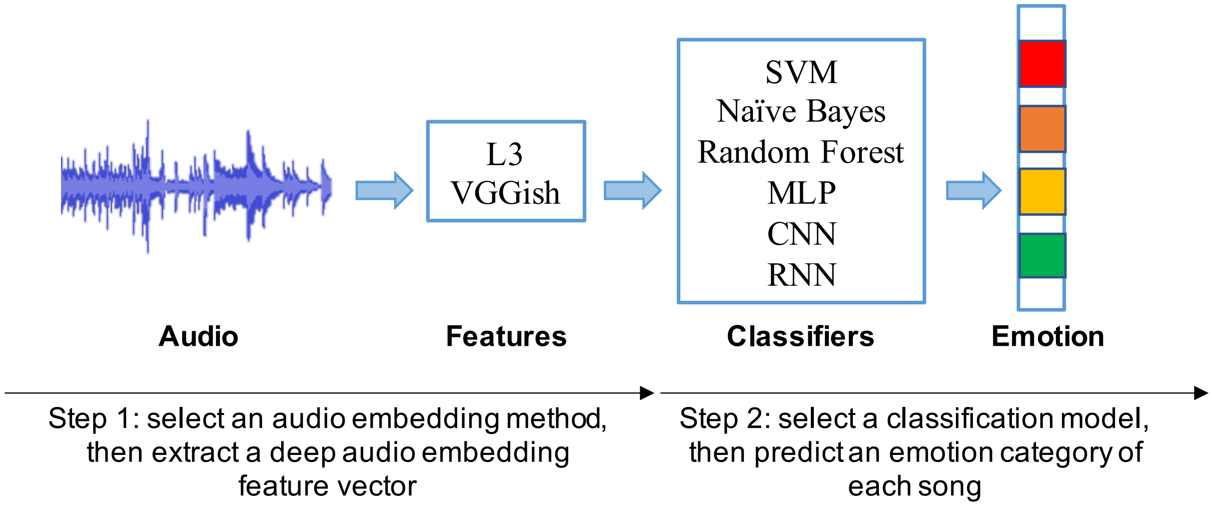

L3
VGGish
SVM
Naïve Bayes
Random Forest
MLP
CNN
RNN
Audio
Features
Classifiers
Emotion
Step 1: select an audio embedding method, then extract a deep audio embedding feature vector
Step 2: select a classification model, then predict an emotion category of each song

Figure 7.1: **The Proposed Workflow.** The figure shows the proposed approach using deep audio embeddings for the MER task.

7.3.2 Deep Audio Embeddings

We choose two deep audio embedding methods, L^3-Net and VGGish, which are state-of-the-art audio representations pre-trained on 60M AudioSet [GEF$^+$17] and Youtube-8M data [AEHKL$^+$16]. AudioSet and Youtube-8M are large labeled training datasets that are widely used in audio and video learning with deep neural networks.

Look, Listen, and Learn network (L^3**-Net)**

L^3-Net is an audio embedding method [CWSB19] motivated by the original work of Look, Listen, and Learn (L^3) [AZ17] that processes Audio-Visual Correspondence learning task in computer vision research. The key differences between the original L^3 (by Arandjelović and Zisserman) and L^3-Net (by Cramer et al.) are (1) input data format (video vs. audio), (2) final embedding dimensionality, and (3) training sample size.

The L^3-Net audio embedding method consists of 2D convolutional layers and 2D max-pooling layers, and each convolution layer is followed by batch normalization and a ReLU nonlinearity (see Figure 7.2). For the last layer, a max-pooling layer is performed to produce a

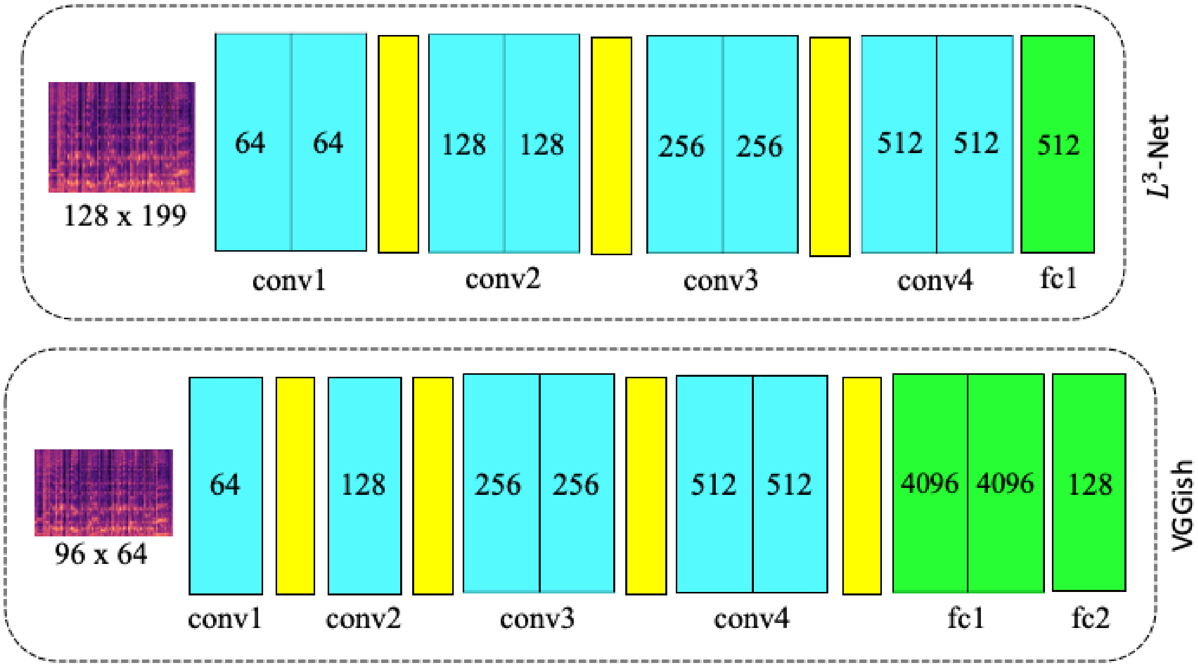


Figure 7.2: **Network Architecture of L^3-Net and VGGish.** The input spectrogram representations are 128x199 for L^3-Net and 96x64 for VGGish. Blue boxes, yellow boxes, and green boxes denote the 2D convolutional layers, max-pooling layers, and fully-connected layers, respectively. The number inside of the blue box is the size of filters and the number inside of the green box is the number of neurons.

single 512 dimension feature vector (L^3-Net serves as an option for output embedding size such as 6144 or 512, and we choose 512 as our embedding size). The L^3-Net method is pre-trained on Google AudioSet 60M training samples containing mostly musical performances [GEF$^+$17].

We follow the design choices of the L^3-Net study which result in the best performance in their SED task. We use Mel spectrograms with 256 Mel bins spanning the entire audible frequency range, resulting in a 512 dimension feature vector. We revise OpenL3 open-source implementation[1]for our experiments.

VGGish

We also verify another deep audio embedding method, VGGish [SZ14], VGGNet based deep audio embedding model. VGGish is a 128-dimensional audio embedding method, motivated by VGGNet [SZ14], and pre-trained on a large YouTube-8M dataset [AEHKL$^+$16]. Original VGGNet is targeting large scale image classification tasks, and VGGish is targeting extracting acoustic features from audio waveforms. The VGGish audio embedding method consists of 2D convolutional layers and 2D max-pooling layers to produce a single 128 dimension feature vector (see Figure 7.2). We modify a VGGish open-source implementation[2] for our experiments.

7.3.3 Music Emotion Classifiers

From the computed deep audio embeddings, we predict an emotion category corresponding to each audio vector as a multi-class classification problem. We employ six different classification models, Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), Multilayer Perceptron (MLP), Convolution Neural Network (CNN), and Recurrent Neural Network (RNN).

For each classification task, we use 80% of the data for training, 10% for testing, and 10% for validation. All six classification models are implemented in Scikit-learn [PVG$^+$11],

[1] OpenL3 open-source library:https://openl3.readthedocs.io/en/latest/index.html

[2] VGGish:https://github.com/tensorflow/models/tree/master/research/audioset/vggish

Keras [Cho15], and Tensorflow [ABC$^+$16]. In the case of MLP, CNN, and RNN classification models, we share some implementation details below.

• MLP: We implement the MLP model with two of a single hidden layer with 512 nodes, a ReLU activation function, an output layer with a number of emotion categories, and a softmax activation function. The model is processed using the categorical cross-entropy loss function and we use Adam stochastic gradient descent [KB14]. We fit the model for 1000 training epochs with the default batch size of 32 samples and evaluate the performance at the end of each training epoch on the test dataset.

• CNN: For CNN classification model, we revise the convolutional filter design proposed by Abdoli et al. [ACK19], which includes four 1D convolution layers and a 1D max-pooling operation layer. Each layer processes 64 convolutional filters. The input to the network is a Mel spectrogram, size of 512 feature vector extracted from a deep audio embedding method. This input size is varied depending on the type of embedding methods. For example, in the case of L^3-Net, the embedding size is 512, VGGish embedding size is 128. ReLU activation functions are applied to the convolutional layers to reduce the backpropagation errors and accelerate the learning process [GBC16]. The softmax function is used as the output activation function with a number of emotion categories. Adam optimizer, categorical cross-entropy loss function, and the batch size of 32 samples are used. The stopping criterion is set as 1000 epochs with an early-stopping rule if there is no improvement to the score during the last 100 learning epochs.

• RNN: Weninger et al. [WES14] propose LSTM-RNN design as an automaton-like structure mapping from an observation sequence to an output feature sequence. We use LSTM networks with a pointwise softmax function based on a number of emotion categories. Adam optimizer, the categorical cross-entropy loss function, and the batch size of 32 samples are used. The same stopping criterion is set as CNNs.

Table 7.1: **Dataset Details.** The number of emotion categories in each dataset and the number of clips in each emotion category are described. Q1, Q2, Q3, Q4 means the emotion categories of the four Arousal-Valence (A-V) quadrants based on Russell's model [Rus03]: Q1 (A+V+), Q2 (A+V-), Q3 (A-V-), Q4 (A-V+). For RAVDESS singing data, it has been classified into six emotion categories, N:Neutral, C:Calm, H:Happy, S:Sad, A:Angry, F:Fearful

	Emotion Category				
Dataset	Q1	Q2	Q3	Q4	Total
4Q Audio	225	225	225	225	900
Bi-modal	52	45	31	34	162
Emomusic	305	87	241	111	744

	Emotion Category						
Dataset	N	C	H	S	A	F	Total
RAVDESS	92	184	184	184	184	20	848

7.4 Evaluation

7.4.1 Dataset

Four different datasets are selected for computing the emotional features in music data. In Table 7.1, we show the number of music files of each dataset by emotion category.

• 4Q Audio Emotion Dataset: This dataset is introduced by Panda et al. [PMP18a], annotated each music clip into four Arousal-Valence (A-V) quadrants based on Rusell's model [Rus03]: Q1 (A+V+), Q2 (A+V-), Q3 (A-V-), Q4 (A-V+). Each emotion category has 225 music clips, and each music clip is 30 seconds long. The total music clips for the dataset are 900 files.

• Bi-modal Emotion Dataset: This dataset is introduced by Malheiro et al. [MPGP16] in a context of bi-modal analysis in the emotion recognition with audio and lyric information. The emotion category is also annotated into four A-V quadrants by Russell's model. In this dataset, each emotion category has a different number of music clips, Q1: 52 clips; Q2: 45 clips; Q3: 31 clips, and Q4: 34 clips, and each music clip is 30 seconds long. The total music clips for the dataset are 162 files. The size of this dataset is the smallest for our experiments.

• Emotion in Music: Using a crowdsourcing platform, Soleymani et al. [SCS$^+$13] release a music emotion dataset with 20,000 arousal and valence annotations on 1,000 music clips. For our experiments, we map the arousal and valence annotation into four A-V quadrants followed by previous Russell's model settings. Each emotion category has a different number of music clips, Q1: 305 clips; Q2: 87 clips; Q3: 241 clips, and Q4: 111 clips, and each music clip is 45 seconds long. We use 744 music clips of the dataset in our experiments. This dataset is one of the most frequently used datasets for the MER task.

• Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): This dataset is introduced by Livingstone et al. [LR18] for understanding the emotional context in speech and singing data. In singing data, it includes the recording clips of human singing with different emotional contexts. 24 different actors were asked to sing in six different emotional states: neutral, calm, happy, sad, angry, fearful. We choose singing data only for our experiments. Each emotion category has a different number of music clips, neutral: 92 clips; calm: 184 clips; happy: 184 clips, sad: 184 clips, angry: 184 clips, and fearful: 20 clips, and each music clip is 5 seconds long. The total music clips for the dataset are 848 files.

7.4.2 Baseline Audio Features

As a baseline feature, we use Mel-Frequency Cepstral Coefficients (MFCCs), which are known to be efficient low-level descriptors for timbre analysis, used as features of music tagging tasks [CFSC17, KLN18]. MFCCs describe the overall shape of a spectral envelope. We first calculate the time derivatives of the given MFCCs and then take the mean and standard deviation over the time axis. Finally, we concatenate all statistics into one vector. We generate the MFCC features of each music clip into a matrix of 20 x 1500. Librosa is used for MFCCs extraction and audio processing [MRL$^+$15].

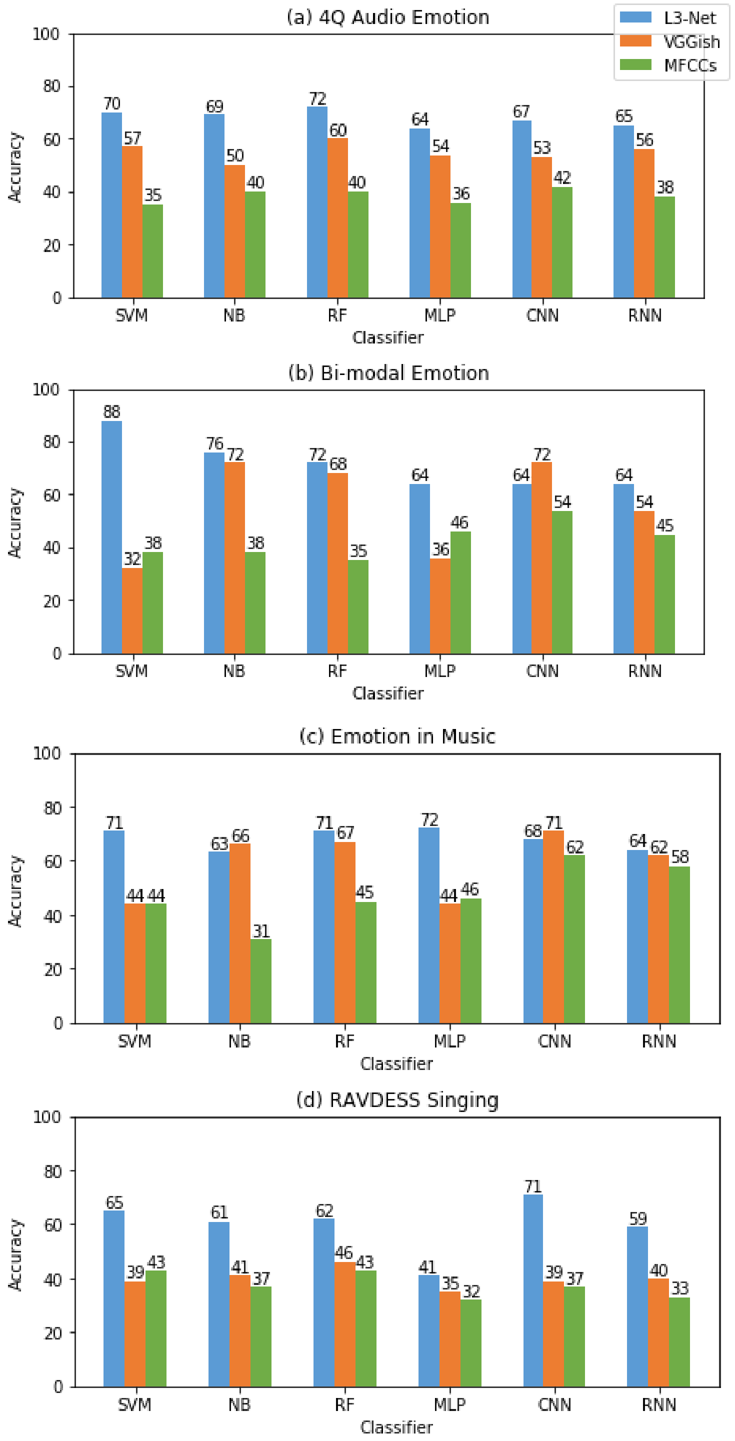


Figure 7.3: **Performance of Emotion Recognition on the Music Emotion Datasets.** Blue bar means the performance of L^3-Net, orange bar for VGGish, and green bar for MFCCs. X-axis indicates the type of classifiers we used, and Y-axis indicates the classification accuracies of the emotion category recognition.

7.4.3 Performance Measures

For classification problems, classifier performance is typically defined according to the confusion matrix associated with the classifier. We use accuracy measure as a primary evaluation criterion. We also calculate F1-score and r^2 score for comparison with other baseline models.

7.4.4 Evaluation of Music Emotion Recognition

In Figure 7.3, we show the performance of deep audio embeddings over four music emotion datasets. We empirically analyze deep audio embeddings in several settings against baseline MFCC features. The experiments are validated with 20 repetitions of cross-validation where we report the average results. We share key observations in the next sections.

7.4.5 Performance Analyzed by Features

The L^3-Net embedding has the best performance in all considered cases except for two, CNN classifier accuracy both in Bi-modal Emotion and Emotion in Music dataset (see Figure 7.3). Even though the L^3-Net embedding is generally not a descriptor for any music-related tasks before, the performance convinces us to use a pre-trained L^3-Net audio embedding model for the MER task.

Since the direct use of the L^3-Net embedding shows the better performance, we also investigate more about the different embedding dimension of the L^3-Net and compare the performance between 512 and 6144. Interestingly, we observe decreasing results with the dimension of 6144 L^3 embeddings. This indicates that those extra features might not be relevant but introducing noise. While the 512 L^3 embeddings show consistent higher performance in many cases, based on our observations, even we increase the depth and number of parameters, 6144 L^3-Net embeddings perform slightly lower on this MER task. Thus, we have not included the performance in the figure. Note that reported results in Figure 7.3 are only considered the performance of 512.

Comparing between L^3-Net and VGGish, L^3-Net outperforms VGGish across the dataset. This could be because L^3-Net was pre-trained on both visual and audio onto the same embedded space which can include more features. The performance of VGGish is better than MFCC baseline features with the rest of the classification models, even though it has fewer parameters, 128. This justifies our use of L^3-Net as a main deep audio embedding choice for MER task and VGGish is for some cases.

It is generally known that decision trees and in our case, RF is better than other neural network based classifiers where the data comprises a large set of categories [PS19]. It also deals better with dependence between variables, which might increase the error and cause some significant features to become insignificant during training. SVM uses kernel trick to solve non-linear problems whereas decision trees derive hyper-rectangles in input space to solve the problem. This is why decision trees are better for categorical data and it deals with co-linearity better than SVM. We still find that SVM outperforms RF in some cases. The reason can be that SVM deals better with margins and thus better handles outliers. Although these are tangential considerations, it seems to support the overall notion that MER is a higher level recognition problem that first needs to address the division of the data into multiple acoustic categories, also requiring the learning of a rather non-trivial partition structure within these sub-categories.

7.4.6 Performance Analyzed by Datasets

For comparison with prior works studying emotions in audio signals, we analyze the performance of previous studies on each dataset we used. We choose four baseline MER models for our experiments: 1) Panda et al. [PMP18a] release the 4Q Music Emotion dataset and present the study of musical texture and expressivity features, 2) Malheiro et al. [MPGP16] present novel lyrical features for MER task and release the Bi-modal Emotion dataset, 3) Choi et al. [CFSC17] present a pre-trained convnet feature for music classification and regression tasks and evaluate the model using Emotion in Music dataset, 4) Arora and Chaspari [AC18] present the method of

Table 7.2: **Performance Comparison with Baseline MER models.** This table shows the data and feature information used in previous baseline models. Data column indicates each dataset for the experiment. Feature column indicates the set of feature vectors extracted by the baseline model. Metric column indicates the metric used for the performance analysis. Baseline column includes the performance of the baseline models. Proposed L^3-Net column includes the best performance of L^3-Net embeddings on each music dataset.

Data	Feature	Metric	Baseline	Proposed L^3-Net
4Q Audio	Domain Knowledge	F	73.5%	72.0%
Bi-modal	Domain Knowledge	F	72.6%	88.0%
Emomusic	Convnet	r^2	A: 0.656 V: 0.462	A: 0.671 V: 0.556
RAVDESS	Siamese	Acc	63.8%	71.0%

a siamese network for speech emotion classification and evaluate the method using RAVDESS dataset. We compare those baseline models to the performance of our proposed method (see Table 7.2).

In the case of the 4Q Audio Emotion dataset, the previous study by Panda et al. obtained its best result of 73.5% F1-score with a high number of 800 features. In Table 7.3, Domain Knowledge means a feature set defined by domain knowledge in the study. For achieving the performance of the previous study, the following steps are needed. First, we need to pre-process standard or baseline audio features of each audio clip. The study used Marsyas, MIR Toolbox, and PsySound3 audio frameworks to extract a total of 1702 features. Second, we need to calculate the correlation between the pair of features for normalization. After the pre-processing, the number of features can be decreased to 898 features. Third, after computing these baseline audio features, we also need to compute novel features of each audio clip proposed by the study. Those features were carefully designed based on domain expertise, such as glissando features, vibrato, and tremolo features. Finally, baseline features and extracted novel features are combined for the MER task. For the evaluation, the study conducted post-processing of the features with the ReliefF feature selection algorithm [RŠK03], ranked the features and evaluated its best-suited features. Since the performance has been evaluated by hyperparameter tuning and feature selection algorithms, these factors may influence the performance of the MER task significantly. Note that in our proposed approach, we show the performance without any post-processing.

In the case of the Bi-modal Emotion dataset, the previous study by Malheiro et al. [MPGP16] presented its best classification result of 72.6% F1-score on the dataset which is lower than the performance we have, 88% F1-score from the result of L^3-Net embedding with SVM classifier.

In the case of the Emotion in Music dataset, previous studies predicted the time-varying arousal and valence annotation and calculated r^2 score as a performance measure [WES14, LLPL19, KLN18, CFSC17]. We previously map these time-varying annotations into four A-V quadrants based on Rusell's model and show our prediction performance with four emotion categories (see Figure 7.3-(c)). For a fair comparison, we also verify the original time-varying dynamic annotations from the dataset [SCS$^+$13] and compare the result with the baseline model. Using the Emotion in Music dataset, Choi et al. reported its r^2 scores of arousal annotation, 0.656 and valence annotation, 0.462 [CFSC17]. The best performance of L^3-Net embeddings achieves 0.671 r^2 score on arousal and 0.556 r^2 score on valence annotation. The result shows that we have a considerable and higher performance on arousal and valence annotation. The result confirms that L^3-Net embedding method shows favorable performance than the previous embedding features over Emotion in Music data.

In the case of RAVDESS data, the study by Arora and Chaspari [AC18] reported its best classification accuracy of 63.8% over the dataset which is lower than our accuracy, 71.0%, from the result of L^3-Net embedding with CNN classifier (see Figure 7.3-(d)).

7.4.7 Performance Analyzed by A-V Quadrants

In Table 7.3, we show the results analyzed by each quadrant. This classification report gives us a further understanding of the characteristic of each emotion category in music. The meaning of each quadrant (Q1, Q2, Q3, Q4) information is described in Table 7.1.

Table 7.3: **Classification Results of Each Quadrant.** The top table indicates the classification report of L^3-Net embedding with Random Forest classifier on 4Q Audio Emotion Dataset. The bottom table indicates the classification report of L^3-Net embedding with SVM classifier on Bi-modal Emotion Dataset.

4Q Audio Emotion	Precision	Recall	F1-score
Q1	0.64	0.85	0.73
Q2	0.85	0.80	0.83
Q3	0.73	0.60	0.66
Q4	0.64	0.61	0.62
Accuracy	-	-	0.72
Weighted Average	0.73	0.72	0.72

Bi-modal Emotion	Precision	Recall	F1-score
Q1	0.80	1.00	0.89
Q2	1.00	0.89	0.94
Q3	1.00	0.67	0.80
Q4	0.80	0.80	0.80
Accuracy	-	-	0.88
Weighted Average	0.90	0.88	0.88

In the case of the 4Q Audio Emotion dataset, Q2 and Q3 categories obtain a higher score compared to the Q1 and Q4. This indicates that emotional features in music clips with lower valence components are easier to recognize. Specifically, the Q2 category shows higher performance which is distinctive than others. Based on the dataset [PMP18b], the study describes music clips of the Q2 category belong to specific genres, such as heavy metal, which have recognizable acoustic features than others.

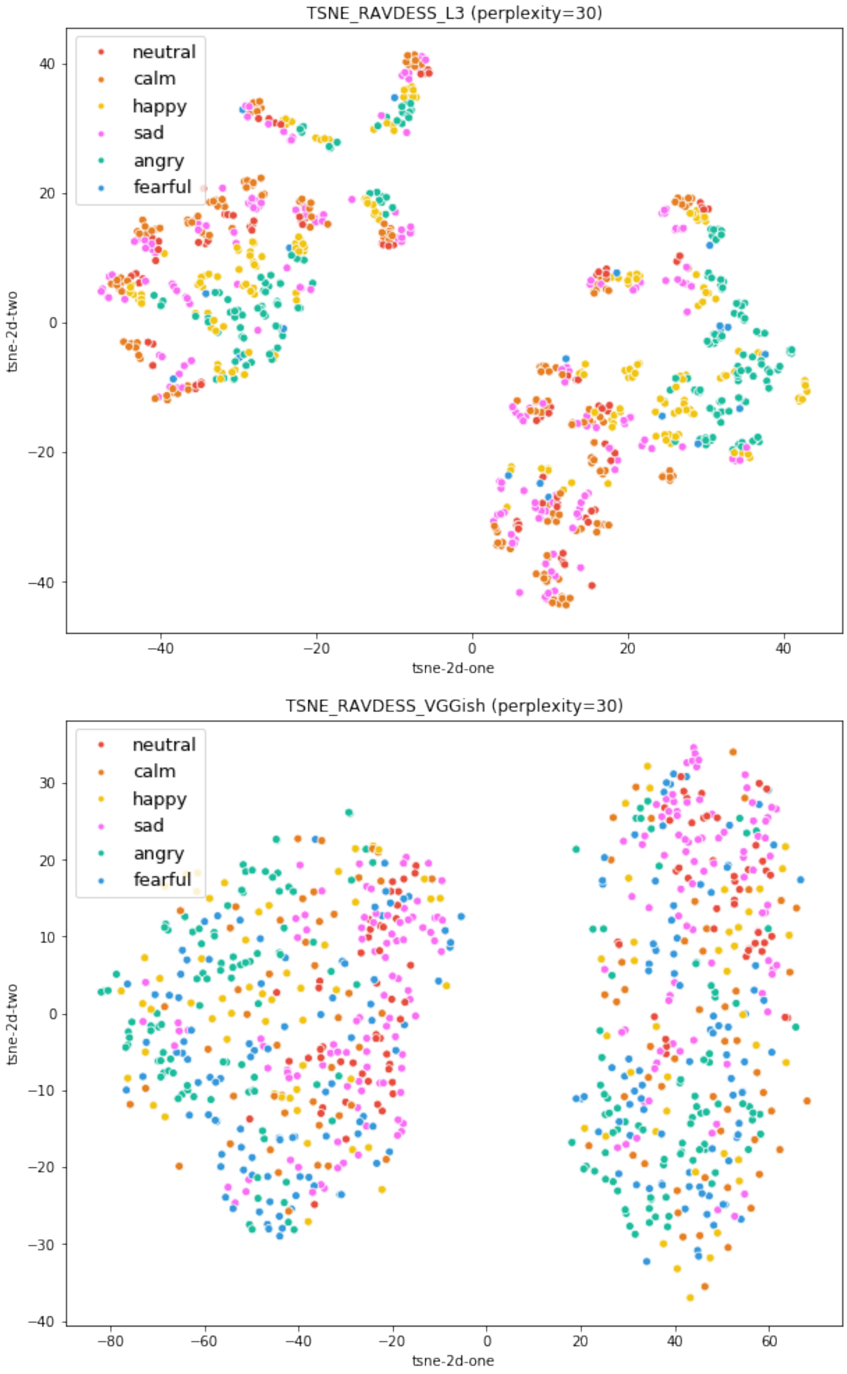


Figure 7.4: **T-SNE Visualization on RAVDESS dataset.** Different colors of dots indicate the type of emotion in the dataset. Both visualizations use a perplexity value of 30. Top: T-SNE Visualization of L^3-Net embeddings, bottom: T-SNE Visualization of VGGish embeddings.

Lower results in Q1 and Q4 categories may also reflect the characteristics of music clips.

For instance, the Q1 category indicates happy emotions, which are typically energetic based on positive arousal and positive valence components. Since Q1 and Q4 categories share the same valence axis based on Rusell's model, if the intensity of the song is not intense, the difference between the two quadrants (Q1&Q4 or Q2&Q3) may not be apparent. This aspect results in similar behaviors on the Q2 and Q3 categories' performances as well.

7.5 Conclusion

In this chapter, we evaluate L^3-Net and VGGish pre-trained deep audio embedding methods for MER task over 4Q Audio Emotion, Bi-modal Emotion, Emotion in Music, and RAVDESS datasets. Even though L^3-Net has not been intended for emotion recognition, we find that L^3-Net is the best representation for the MER task. Note that we achieve this performance without any additional domain knowledge feature selection method, feature training process, and fine-tuning process. Comparing to MFCC baseline features, the empirical analysis shows that L^3-Net is robust across multiple datasets with favorable performance. Overall, the result using L^3-Net shows improvement compared to baseline models for Bi-modal Emotion, Emotion in Music, and RAVDESS dataset. In the case of the 4Q Audio Emotion dataset, complex hand-crafted features (over 100 features) still seem to perform better. Specifically, our work does not consider rhythm or specific musical parameters over the time axis that 4Q Audio Emotion had, looking into time-based aspects could be the next step for future research.

In order to gain deeper insight into the meaning of acoustic features for emotional recognition, we use T-SNE visualization (see Figure 7.4). In both cases of L^3-Net and VGGish, two main clusters on the left and right side of the figure mean male/female singer groups. We can also see a relatively smooth grouping of samples by emotions with different colors. In the case of L^3-Net embeddings (top figure of Figure 7.4), multiple small groups in each cluster indicate individual singer which has audio recordings in different emotions. L^3-Net data seems to cluster into multiple smaller groups according to gender and individual categories, and this shows L^3-Net outperforms for detecting different timbre information than VGGish. This pattern seems to be consistent in the wild range of T-SNE perplexity parameters. This also shows that our study provides an empirical justification that L^3-Net outperforms VGGish, with the intuition discussed in the paper based on the clustering shown in Figure 7.4.

Accordingly, for the next step, a possible direction to validate different classifiers is to explore a combination of discrete neural learning methods, such as VQ-VAE, to first solve the categorical problem, and only later learn a more smooth decision surface. VQ-VAE has been recently explored for spectrogram-based music inpainting [BHEM20]. It would be interesting to explore similar high-level parameterization using L^3-Net embeddings.

7.6 Discussions: Music Analysis with VMO-RQA

In this chapter, we study affective aspects of music from the perspective of audio signal, demonstrating the importance of understanding the structural aspects of audio for the perception of emotion in music. In this discussion, we further investigate other relevant audio features for music emotion analysis with the method of nonlinear dynamics analysis in terms of symbolized recurrence properties.

7.6.1 Recurrence Quantification Analysis (RQA)

In the previous study of understanding affect in audio signals, the method of Recurrence Quantification Analysis (RQA) was proposed [MD17]. RQA is a methodology that computes nonlinear dynamics with a technique of adaptive time series understanding in terms of symbolized recurrence properties. For getting RQA features from the audio signals, first, we compute symbolic recurrence quantification measures from symbolic recurrence plots in the process of Variable Markov Oracle (VMO) computation. We use VMO to find the repetition structure in the audio sequence in order to analyze temporal aspects of sound and calculate several symbolic recurrence quantification properties from symbolic recurrence plots in VMO. Using VMO, we are able to analyze the graph structure to find summary statistics that capture the dynamics of the feature in terms of several measures, such as the density of the recurrence points, ratio of repeated motifs (or recurrence diagonals), relative to all recurrence points, the average length of motifs and so on.

7.6.2 Understanding Temporal Aspects in VMO-RQA

For classifying data according to emotion, here we try to understand the structural aspects of each data. The process of analysis can be summarized as follows. We model a sequence of the deep audio embeddings using VMO. In this step, a threshold search is performed to find the optimal recurrence graph in terms of its predictive properties which is Information Rate. Then, the optimal recurrence graph is converted to a matrix form, recurrence plot, which is further analyzed in terms of recurrence quantification features. After finding the structures, we then summarize using several statistics that are specifically designed to reveal recurrence structure.

We select a few temporal properties in RQA components, through structural understanding using VMO. During the process, we are able to recognize information dynamics in music, examine how emotional changes over time fluctuate according to music structure, and how the structure of music affects music emotion generation. This statistical analysis offers us a new perspective on the connection between emotion and recurrence quantification features.

This study demonstrates the possibility of an easily implementable computational method for music emotion recognition studies. We also visualize the musical structure using VMO matrix representation, which is a visualization tool of repetition statistics. In this work, we propose a systematic way to test the expressivity of deep representations, considering musical semantics. In our future research, we plan to further explore how semantic information in music may connect to humans' perceptions and intentions for a better understanding of information dynamics in music.

Chapter 7 is adapted from published material in ”Comparison and Analysis of Deep Audio Embeddings for Music Emotion Recognition”. Koh, Eunjeong and Dubnov, Shlomo. Association for the Advancement of Artificial Intelligence (AAAI) Workshop on Affective Content Analysis, 2021, and ”Understanding Affective Aspects of Music Using Deep Audio Embeddings”. Koh, Eunjeong and Dubnov, Shlomo. International Conference on Music Perception and Cognition, 2021. The dissertation/thesis author was the primary researcher and author of the paper.

Chapter 8

Other Music and Emotion Research

8.1 The Role of Musical Structure in Shaping Listener's Preference

What makes people like a song? Earworms are the repeats of a song that remain in a listener's head, which generate one's preference for the song. In this chapter, we explore the relationship between musical preference and the elements of musical structure, possibly independent from the semantic content of lyrics. We investigate positive correlations for emotional response between musical dynamics and the level of interest in replaying the song through listening experiments. Native and non-native speakers of English listened to excerpts of musical pieces, either with lyrics in their own language or another. We asked the participants to evaluate the level of their emotional response to multiple songs and which songs they were interested in replaying. Based on the unsupervised machine learning method, we extracted features from each excerpt that were correlated to participants' subjective responses. The participants' ratings of each song (e.g. appraisal of musical elements) were collected in several study sessions, where they eventually selected their most favorite song to be replayed from the playlist. We have analyzed the participants' data with statistical models, which would categorize the relationship between the data collected and the elements of musical structure. Our preliminary results have been consistent with the findings that emotional judgments could be related not only to the meaning expressed by the lyrics but also to the elements of musical structure, especially musical repeats. Our empirical data suggest that musical preference would have the following properties: 1) repetition-based structure in music should create strong impressions on the listener's long-term memory, and 2) certain compositional designs of musical structure could be an indicator of the recurring interest deriving from earworms. While preliminary, these results suggest the possibility of discovering a shared cognitive mechanism for meeting the musical expectations of listeners and enabling broader musical communication among different cultures.

8.1.1 Research Objectives

We design song-listening experiments to investigate the relationship between musical dynamics and the listener's interest in replaying the song. We also analyze compositional designs of musical structure as an indicator of the recurring interest.

8.1.2 Pilot study: Preference of the Seven Songs

Seven different samples of Korean pop songs were chosen as our data classes (e.g., dance, ballad, hip-hop, trot, folk-blues). Native speakers of English and Korean listened to excerpts of musical pieces either with lyrics in their own language or another. Then, they answered this question for understanding their preference for the seven songs.

”How much would you like to listen to the songs again?”

When the U.S. participants, non-native speakers of the Korean lyrics, were asked to choose the song that they liked the most (among the seven songs), 43.8% chose the song which is highly repetitive. Also, the Korean participants, native speakers of the Korean lyrics, chose the same song as U.S. participants. These are some examples of listener's comments: ”The steady and uplifting beat is comfortable to listen”, ”I also enjoy the fast beat”, ”Catchy, energetic”, ”Fast, cheerful, and upbeat!”

8.1.3 Conclusion

Highly repetitive songs were clustered together in unsupervised learning techniques and were preferred by U.S. participants over other songs. In terms of song preferences, musical repetition would play a key role than the lyrics, which may suggest the possibility of a shared cognitive mechanism for musical expectations among different cultures.

Chapter 8 is adapted from published material in ”The Role of Musical Structure in Shaping Listener’s Preference”. Koh, Eunjeong, and Kim, Min-ju. Society for Music Perception and Cognition, 2017. The dissertation/thesis author was the primary researcher and author of the paper.

Appendix A

Summary

In this part, we share the goals, achievements, and some details of each chapter.

A.1 Chapter 3

A.1.1 The Goal of the Experiments

We use a neural adapter to effectively bridge the gap between the previously learned information in the source model and a target model for learning new sound events.

A.1.2 Dataset

We evaluate our algorithm over three datasets; the DCASE 2016 challenge Task 2 (DCASE16) [MHV16b], the UrbanSound-SED (US-SED), and UrbanSound-8K (US-8K) [SJB14] dataset.

A.1.3 Implementation

· Existing tools

We revise this open-source implementation for creating soundscapes sound files `https://github.com/justinsalamon/scaper_waspaa2017`

· New implementation

We implement the Neural Adapter structure from scratch. The neural adapter structure ideas have been adapted from here.

A.1.4 The Scope of the Experiments

In this experiment, the source model and the target model perform limited sound events detection. The source model has been designed for detecting three different sound events and the target model has been designed for detecting four different sound events.

A.1.5 Accomplishment

We present an incremental learning algorithm utilizing a TL paradigm for SED application. Our extensive analysis shows that utilizing such a mechanism improves the performance of recognizing both known/unknown sound events without forgetting the previously learned knowledge. Thus, our proposed model suits well the scalable and incremental SED applications.

A.1.6 Future Works

This approach can also be used as a low footprint framework for continuous learning in applications that involve less noisy and well-annotated data. However, for the more realistic applications, such as acoustic scene classification systems that involve more noisy data, both the target model and the source model might need to remain connected to achieve the desired performance. Addressing such a challenge remains the focus of our future work.

A.2 Chapter 4

A.2.1 The Goal of the Experiments

We design CNN-VRNN machine learning model for automatic music composition using MIDI data. We compare the performance of our proposed model with other types of Neural Networks using the criteria of Information Rate that is implemented by Variable Markov Oracle, a method that allows statistical characterization of musical information dynamics and detection of motifs in a song. Our results suggest that the proposed model has a better statistical resemblance to the musical structure of the training data, which improves the creation of new sequences of music in the style of the originals.

A.2.2 Dataset

For our experiments, our training data comes from the Nottingham Dataset, a collection of 1200 folk songs [not]. Each training song is segmented into frames (piano-roll), and for the preprocessing of our dataset, we implement our method based on the music21, librosa, and pretty midi packages for feature extraction on MIDI file [CA10, MRL+15, RE]. We use an input of 128 binary visible units and are aligned on the 8th note beat level.

A.2.3 Implementation

· Baselines

MelodyRNN, PolyphonyRNN, AttentionRNN, and Midinet

`https://github.com/magenta/magenta/blob/main/magenta/models/melody_rnn/README.md`

· New implementation

We implement CNN-VRNN from scratch.

Github link: `https://github.com/skokoh/c_vrnn_mmsp_2018`

Soundcloud: `https://soundcloud.com/user-431911640/sets`

A.2.4 The Scope of the Experiments

We present a model for capturing musical features and creating novel sequences of music, called the Convolutional-Variational Recurrent Neural Network. To generate sequential data, the model uses an encoder-decoder architecture with latent probabilistic connections to capture the hidden structure of music. Using the sequence-to-sequence model, our generative model can exploit samples from a prior distribution and generate a long sequence of music. Qualitative experiments can improve this experiment's results understanding.

A.2.5 Musical Point of View

We show average IRs for original Nottingham MIDI datasets and for generated samples from several models, where higher IRs report more distinct self-similarity structures. The IR of the original dataset is higher than that of the generated music. Self-similarity in audio refers to the multi-scalar feature in a set of relationships, and it commonly indicates musical coherence and consistency [Foo99]. Self-similarity in audio refers to multi-scalar features in a set of relationships, and it commonly indicates musical coherence and consistency. Results in each setting show that the latent variable sampling approach increases the IR of the produced musical material over RNN-based approaches, indicating a higher degree of structure. In addition, the distribution in each group shows that the result of [KDW18] has more distribution than the training dataset or another model. This is because the research focuses on the creation of a diverse and complex new sequence of music.

A.2.6 Accomplishment

In this study, we show initial proof that our proposed model applied to MIDI sequence representations can capture the structure of the song and create polyphonic music. The motivations behind combining CNN, RNN, and VAE were to explore significant problems in music generation which are related to representation issues that are handled via CNN, repetitive patterns in generated output that are known in RNN and the ability to generate variations from the progression of melody sequence. In our study, we used IR as a criteria to evaluate the generated output and compare it to other models.

A.2.7 Future Works

Qualitative experiments can be helpful for a better understanding of our generated music structure.

A.3 Chapter 5

A.3.1 The Goal of the Experiments

We tackle an important problem of the evaluation and interpretation of machine learning models with a focus on music models. To quantitatively evaluate the quality of music, we use Information Rate which can compare how information flows in original music versus in music generated by an artificial neural network that learned that music. We review mostly known but often underappreciated properties relating to the evaluation and interpretation of machine learning models with a focus on music models. We aim to study the perspectives from statistics and information theory as to how creativity can be measured in both a computationally and musically meaningful way.

A.3.2 Dataset

For our experiments, our training data comes from the Nottingham Dataset, a collection of 1200 folk songs [not].

A.3.3 Implementation

- Baselines

  Boulanger et al [BLBV12]

- New implementation

  Soundcloud:

  `https://soundcloud.com/user-431911640/sets`

A.3.4 The Scope of the Experiments

Results in each setting show that the latent variable sampling approach increases the IR of the produced musical material over RNN-based approaches, indicating a higher degree of structure. In addition, the distribution in each group shows that the result of [KDW18] has more distribution than the training dataset or another model. This is because the research focuses on the creation of a diverse and complex new sequence of music.

A.3.5 Musical Point of View

The results show the relation between IR and threshold value and imply different musical structures are generated by different values. In terms of the results graphs, the RNN-RBM recopies longer segments, but they are interrupted, while the CNN-Recurrent VAE model relies on shorter previous patterns. The CNN-Recurrent VAE uses a latent variable model to design the graph, so we can see that the chord progression of the music is more variously than iterative or repetitive structure.

A.3.6 Accomplishment

The motifs over time (and submotifs when the lines overlap vertically) allow visual inspection of such structures. The higher level repetitions also exist in the arrangement of motifs themselves. Since each VMO analysis optimizes the threshold of similarity for approximate suffix matching, the motifs shown in the different figures appear slightly different. Also, the VMO takes into consideration also the later motif structure and adjusts its sensitivity so as to produce the most informative representation of the overall information in each piece.

A.4 Chapter 6

A.4.1 The Goal of the Experiments

We present a novel musical interface, Mugeetion, designed to capture occurring instances of emotional states from users' facial gestures and relay that data to associated musical features. Mugeetion can translate qualitative data of emotional states into quantitative data, which can be utilized in the sound generation process.

A.4.2 Dataset

We list our sound files below, which have been played to a number of subjects interacting with the system. The selection of the list is based on the study of the Mozart Effect [PCBCC13]. For the sound files, we use Piano-midi.de dataset.

· Happy

Mozart - The Piano Sonata No 16 in C major Mozart - Eine Kleine Nachtmusik K 525 Allegro

· Neutral

Mozart - Piano Sonata No 11 in A major K 331

· Sad

Mozart - Symphony No 25 in G Minor K 183 1st Movement Mozart - Requiem in D minor

A.4.3 Implementation

· Existing tools

FAC, FaceOSC

· New implementation

We built a Max application, which utilizes facial data and FAC for sound creation. Our preliminary demo video is uploaded on Youtube. `https://www.youtube.com/playlist?list=PLjaQX_vKy2Jcv0r9wrc_yU2gbRhZh39GV`

A.4.4 The Scope of the Experiments

We also presented and tested this work in the exhibition of sound installation, Hearing Seascape, using the audiences' facial expressions. Audiences heard changes in the background sound based on their emotional state. The process contributes to multiple research areas, such as gesture tracking systems, emotion-sound modeling, and the connection between sound and facial gesture.

A.4.5 Accomplishment

Mugeetion makes several contributions to previous work. Rather than simply detecting facial gesture data, it also automatically extracts emotional states and produces sound output transition. Mugeetion provides a sound generation model to users based on the components of emotion and musical metadata. We focus on how sound can be changed based on users' emotional movement.

A.4.6 Future Works

We will further develop the system based on the following issues: increasing training images for covering multiple faces and optimizing different emotional states, implementing an AU indicator or other emotional measure on the FaceOSC display for better interaction with users exploring other similar emotion interactive systems to compare the sonification result.

A.5 Chapter 7

A.5.1 The Goal of the Experiments

We evaluate L3-Net and VGGish pre-trained deep audio embedding methods for MER task over 4Q Audio Emotion, Bi-modal Emotion, Emotion in Music, and RAVDESS datasets. Even though L3-Net has not been intended for emotion recognition, we find that L3-Net is the best representation for the MER task. Note that we achieve this performance without any additional domain knowledge feature selection method, feature training process, and fine-tuning process.

A.5.2 Dataset

Four different datasets are selected for computing the emotional features in music data, 4Q Audio Emotion Dataset [PMP18a], Bi-modal Emotion Dataset [MPGP16], Emotion in Music [SCS+13], Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [LR18].

A.5.3 Implementation

· Existing tools

We choose two deep audio embedding methods, L3-Net and VGGish, which are state-of-the-art audio representations pre-trained on 60M AudioSet [GEF+17] and Youtube-8M data [AEHKL+16]. AudioSet and Youtube-8M are large labeled training datasets that are widely used in audio and video learning with deep neural networks.

· New implementation

From the computed deep audio embeddings, we predict an emotion category corresponding to each audio vector as a multi-class classification problem. We employ six different classification models, Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), Multilayer Perceptron (MLP), Convolution Neural Network (CNN), and Recurrent Neural Network (RNN).

A.5.4 The Scope of the Experiments

We didn't design our own method of deep audio embedding for this experiment. We used pre-trained L3-Net and VGGish and adapted it to different tasks from SED to MER.

A.5.5 Accomplishment

Comparing to MFCC baseline features, the empirical analysis shows that L3-Net is robust across multiple datasets with favorable performance. Overall, the result using L3-Net shows improvement compared to baseline models for Bi-modal Emotion, Emotion in Music, and RAVDESS dataset.

A.5.6 Future Works

A possible direction to validate different classifiers is to explore a combination of discrete neural learning methods, such as VQ-VAE, to first solve the categorical problem, and only later learn a more smooth decision surface. VQ-VAE has been recently explored for spectrogram-based music inpainting [BHEM20]. It would be interesting to explore similar high-level parameterization using L3-Net embeddings.

A.6 Chapter 8

A.6.1 The Goal of the Experiments

We explore the relationship between musical preference and the elements of musical structure, possibly independent from the semantic content of lyrics. We investigate positive correlations for emotional response between musical dynamics and the level of interest in replaying the song through listening experiments.

A.6.2 The Scope of the Experiments

Native and non-native speakers of English listened to excerpts of musical pieces, either with lyrics in their own language or another. We asked the participants to evaluate the level of their emotional response to multiple songs and which songs they were interested in replaying.

A.6.3 Accomplishment

Our preliminary results have been consistent with the findings that emotional judgments could be related not only to the meaning expressed by the lyrics but also to the elements of musical structure, especially musical repeats. Our empirical data suggest that musical preference would have the following properties: 1) repetition-based structure in music should create strong impressions on the listener's long-term memory, and 2) certain compositional designs of musical structure could be an indicator of the recurring interest deriving from earworms.

Bibliography

[ABC$^+$16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Kudlur Manjunath Isard, Michael, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[AC18] Priya Arora and Theodora Chaspari. Exploring siamese neural network architectures for preserving speaker identity in speech emotion classification. In *Proceedings of the 4th International Workshop on Multimodal Analyses Enabling Artificial Agents in Human-Machine Interaction*, pages 15–18, 2018.

[ACK19] Sajjad Abdoli, Patrick Cardinal, and Alessandro Lameiras Koerich. End-to-end environmental sound classification using a 1d convolutional neural network. *Expert Systems with Applications*, 136:252–263, 2019.

[AEHKL$^+$16] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[Art28] Various Artists. Search ”faceosc” at youtube, Last checked: 2018/02/28. Available at `https://www.youtube.com/results?search_query=faceosc`.

[AZ17] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017.

[BHEM20] Théis Bazin, Gaëtan Hadjeres, Philippe Esling, and Mikhail Malt. Spectrogram inpainting for interactive generation of instrument sounds, 2020.

[BHP17] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep learning techniques for music generation-a survey. *arXiv preprint arXiv:1709.01620*, 2017.

[BLBV12] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1881–1888. Omnipress, 2012.

[BWH16] Mason Bretan, Gil Weinberg, and Larry Heck. A unit selection methodology for music generation using deep neural networks. *arXiv preprint arXiv:1612.03789*, 2016.

[CA10] Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. 2010.

[Çam12] Anıl Çamcı. A cognitive approach to electronic music: theoretical and experiment-based perspectives. In *Proceedings of the International Computer Music Conference*, pages 1–4, 2012.

[CFSC17] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. *arXiv preprint arXiv:1703.09179*, 2017.

[Cho15] François Chollet. keras, 2015.

[CKD$^+$15] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015.

[CLB$^+$20] Kin Wai Cheuk, Yin-Jyun Luo, BT Balamurali, Gemma Roig, and Dorien Herremans. Regression-based music emotion prediction using triplet neural networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[CM01] C-CJ Chen and Risto Miikkulainen. Creating melodies with evolving recurrent neural networks. In *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, volume 3, pages 2241–2246. IEEE, 2001.

[CM19] Lingzhen Chen and Alessandro Moschitti. Transfer learning for sequence labeling using source model and target data. *arXiv preprint arXiv:1902.05309*, 2019.

[CPL11] Anthony Churnside, Chris Pike, and Max Leonard. Musical movements—gesture based audio interfaces. In *Audio Engineering Society Convention 131*. Audio Engineering Society, 2011.

[CvM$^+$14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[CWSB19] Jason Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello. Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3852–3856. IEEE, 2019.

[CZX$^+$16] Peilin Chen, Lei Zhao, Zongyu Xin, Yumeng Qiang, Ming Zhang, and Tiemeng Li. A scheme of midi music emotion classification based on fuzzy theme extraction and neural network. In *2016 12th International Conference on Computational Intelligence and Security (CIS)*, pages 323–326. IEEE, 2016.

[dAD13] Nicolas d'Alessandro, Maria Astrinaki, and Thierry Dutoit. Mageface: Performative conversion of facial characteristics into speech synthesis parameters. In *International Conference on Intelligent Technologies for Interactive Entertainment*, pages 179–182. Springer, 2013.

[DCA] Detection and Classification of Acoustic Scenes and Events 2019 (DCASE2019). Task 4: sound event detection in domestic environments. `http://dcase.community/challenge2019/\task-sound-event-detection-in-\domestic-environments`. Accessed: 2019-12-06.

[DHYY17] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Symbolic-domain music generation and accompaniment with multi-track sequential generative adversarial networks. *arXiv preprint arXiv:1709.06298*, 2017.

[DTM14] Shichuan Du, Yong Tao, and Aleix M Martinez. Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462, 2014.

[DYZL19] Yizhuo Dong, Xinyu Yang, Xi Zhao, and Juan Li. Bidirectional convolutional recurrent sparse network (bcrsn): An efficient model for music emotion recognition. *IEEE Transactions on Multimedia*, 21(12):3150–3163, 2019.

[EHB96] Salah El Hihi and Yoshua Bengio. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in neural information processing systems*, pages 493–499, 1996.

[EML$^+$18] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

[FC20] Kexin Feng and Theodora Chaspari. A siamese neural network with modified distance loss for transfer learning in speech emotion recognition. *arXiv preprint arXiv:2006.03001*, 2020.

[Foo99] Jonathan Foote. Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 77–80. ACM, 1999.

[FvA14] Otto Fabius and Joost R van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[GEF$^+$17] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin 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. IEEE, 2017.

[GFG18] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–53, 2018.

[Haw02] Tim Hawkins. Emoter, 2002. Available at `https://www.artsy.net/artwork/tim-hawkinson-emoter`.

[HE10] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, volume 10, pages 339–344. Utrecht, The Netherlands, 2010.

[Hur08] David Huron. *Sweet anticipation: Music and the psychology of expectation*. MIT press, 2008.

[HUW17] Jay A Hennig, Akash Umakantha, and Ryan C Williamson. A classifying variational autoencoder with application to polyphonic music generation. *arXiv preprint arXiv:1711.07050*, 2017.

[Jen12] Alexander Refsum Jensenius. Motion-sound interaction using sonification based on motiongrams. 2012.

[JPL19] Seokwon Jung, Jungbae Park, and Sangwan Lee. Polyphonic sound event detection using convolutional bidirectional lstm and synthetic data-based transfer learning. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 885–889. IEEE, 2019.

[Kap10] Jaroslaw Kapuscinski. Where is chopin?, 2010. Available at `http://www.jaroslawkapuscinski.com/Where_Is_Chopin/index.php`.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KCT00] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 46–53. IEEE, 2000.

[KDW18] Eunjeong Stella Koh, Shlomo Dubnov, and Dustin Wright. Rethinking recurrent latent variable model for music composition. In *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2018.

[Kel12] Robert Keller. Impro-visor. *Harvey Mudd Computer Science Department,[online] Available from: http://www. cs. hmc. edu/˜ keller/jazz/improvisor/(Accessed 27 March 2013)*, 2012.

[KGGS14] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. In *International Conference on Machine Learning*, pages 1863–1871. PMLR, 2014.

[KKF18] Anurag Kumar, Maksim Khadkevich, and Christian Fügen. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 326–330. IEEE, 2018.

[KLN18] Taejun Kim, Jongpil Lee, and Juhan Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 366–370. IEEE, 2018.

[Kra00] Gregory Kramer. *Auditory display: sonification, audification and auditory interfaces*. Addison-Wesley Longman Publishing Co., Inc., 2000.

[KSM$^+$10] Youngmoo E Kim, Erik M Schmidt, Raymond Migneco, Brandon G Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *Proc. ISMIR*, volume 86, pages 937–952, 2010.

[KTPLW14] Mats B Küssner, Dan Tidhar, Helen M Prior, and Daniel Leech-Wilkinson. Musicians are more consistent: Gestural cross-modal mappings of pitch, loudness and tempo in real-time. *Frontiers in psychology*, 5, 2014.

[KULH19] Jaehun Kim, Julián Urbano, Cynthia Liem, and Alan Hanjalic. Are nearby neighbors relatives?: Are nearby neighbors relatives?: Testing deep music embeddings. *Frontiers in Applied Mathematics and Statistics*, 5:53, 2019.

[KW13] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[KY18] Eunjeong Stella Koh and Shahrokh Yadegari. Mugeetion: Musical interface using facial gesture and emotion. *arXiv preprint arXiv:1809.05502*, 2018.

[LCK+10] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010.

[LCY13] Yi Lin, Xiaoou Chen, and Deshun Yang. Exploration of music emotion recognition based on midi. In *ISMIR*, pages 221–226, 2013.

[Lem08] Marc Leman. Embodied music cognition and music mediation technology, 2008.

[LFH19] Huaping Liu, Yong Fang, and Qinghua Huang. Music emotion recognition using a variant of recurrent neural network. In *2018 International Conference on Mathematics, Modeling, Simulation and Statistics Application (MMSSA 2018)*. Atlantis Press, 2019.

[LGW16] Stefan Lattner, Maarten Grachten, and Gerhard Widmer. Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints. *arXiv preprint arXiv:1612.04742*, 2016.

[LLPL19] Donmoon Lee, Jaejun Lee, Jeongsoo Park, and Kyogu Lee. Enhancing music features by knowledge transfer from user-item log data. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 386–390. IEEE, 2019.

[LR18] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[LT01] Michael J Lyons and Nobuji Tetsutani. Facing the music: a facial action controlled musical interface. In *CHI'01 extended abstracts on Human factors in computing systems*, pages 309–310. ACM, 2001.

[LXC+15] Yilong Liu, Feng Xu, Jinxiang Chai, Xin Tong, Lijuan Wang, and Qiang Huo. Video-audio driven real-time facial animation. *ACM Transactions on Graphics (TOG)*, 34(6):182, 2015.

[Lyo17] Michael J Lyons. Machine intelligence, new interfaces, and the art of the soluble. *arXiv preprint arXiv:1707.08011*, 2017.

[MC89] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[McD16] Kyle McDonald. Faceosc, Latest Release: 2016. Available at `https://github.com/kylemcdonald/ofxFaceTracker/releases`.

[MD17] Pauline Mouawad and Shlomo Dubnov. On modeling affect in audio with non-linear symbolic dynamics. *Advances in Science, Technology and Engineering Systems Journal*, 2(3):1727–1740, 2017.

[Mel] Melodyrnn. Available at `https://github.com/tensorflow/magenta/tree/master/magenta/models/`.

[MGG18] Rishi Madhok, Shivali Goel, and Shweta Garg. Sentimozart: Music generation based on emotions. In *ICAART (2)*, pages 501–506, 2018.

[MHV16a] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162, 2016.

[MHV16b] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Tut database for acoustic scene classification and sound event detection. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1128–1132. IEEE, 2016.

[MHV18] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. A multi-device dataset for urban acoustic scene classification. *arXiv preprint arXiv:1807.09840*, 2018.

[Mog16] Olof Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

[MPGP16] Ricardo Malheiro, Renato Panda, Paulo Gomes, and Rui Paiva. Bi-modal music emotion recognition: Novel lyrical features and dataset. 9th International Workshop on Music and Machine Learning–MML'2016–in . . . , 2016.

[MRL$^+$15] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[MSE14] Alex Migicovsky, Jonah Scheinerman, and Georg Essl. Moveosc—smart watches in mobile music performance. In *ICMC*, 2014.

[not] Nottingham database. Available at `http://ifdo.ca/~seymour/nottingham/nottingham.html`.

[OE18] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

[OIM$^+$16] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016.

[PCBCC13] Leonid Perlovsky, Arnaud Cabanac, Marie-Claude Bonniot-Cabanac, and Michel Cabanac. Mozart effect, cognitive dissonance, and the pleasure of music. *Behavioural Brain Research*, 244:9–14, 2013.

[Pic15] Karol 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. IEEE, 2015.

[PLF$^+$01] Ivan Poupyrev, Michael J Lyons, Sidney Fels, et al. New interfaces for musical expression. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*, pages 491–492. ACM, 2001.

[PMP18a] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. Musical texture and expressivity features for music emotion recognition. In *ISMIR*, pages 383–391, 2018.

[PMP18b] Renato Panda, Ricardo Manuel Malheiro, and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 2018.

[PS19] Jordi Pons and Xavier Serra. Randomly weighted cnns for (music) audio classification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 336–340. IEEE, 2019.

[PVG$^+$11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[RE] Colin Raffel and Daniel PW Ellis. Data with pretty_midi.

[REE] Adam Roberts, Jesse Engel, and Douglas Eck. Musicvae. Available at `https://github.com/tensorflow/magenta/tree/master/magenta/models/music_vae`.

[RMW14] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[RŠK03] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2):23–69, 2003.

[Rus03] James A Russell. Core affect and the psychological construction of emotion. *Psychological review*, 110(1):145, 2003.

[SB17] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

[SBC$^+$16] Jennifer E Smith, Rusty Brainard, Amanda Carter, Saray Grillo, Clinton Edwards, Jill Harris, Levi Lewis, David Obura, Forest Rohwer, Enric Sala, Peter S Vroom, and Stuart Sandin. Re-evaluating the health of coral reef communities: baselines and evidence for human impacts across the central pacific. In *Proc. R. Soc. B*, volume 283, page 20151985. The Royal Society, 2016.

[SCS$^+$13] Mohammad Soleymani, Micheal N Caro, Erik M Schmidt, Cheng-Ya Sha, and Yi-Hsuan Yang. 1000 songs for emotional analysis of music. In *Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia*, pages 1–6. ACM, 2013.

[SCT04] Anne Sedes, Benoit Courribet, and Jean-Baptiste Thiebaut. From the visualization of sound to real-time sonification: different prototypes in the max/msp/jitter environment. In *ICMC*, 2004.

[SGH$^+$19] Fatemeh Saki, Yinyi Guo, Cheng-Yu Hung, Lae-Hoon Kim, Manyu Deshpande, Sunkuk Moon, Eunjeong Koh, and Erik Visser. Open-set evolving acoustic scene classification system. 2019.

[SJB14] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM, 2014.

[SMC$^+$17] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello. Scaper: A library for soundscape synthesis and augmentation. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 344–348. IEEE, 2017.

[SSKS17] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[SZL$^+$18] Yang Song, Jingwen Zhu, Dawei Li, Xiaolong Wang, and Hairong Qi. Talking face generation by conditional recurrent adversarial network. *arXiv preprint arXiv:1804.04786*, 2018.

[THR19] Ha Thi Phuong Thao, Dorien Herremans, and Gemma Roig. Multimodal deep models for predicting affective responses evoked by movies. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1618–1627. IEEE, 2019.

[TM95] Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46, 1995.

[VdODS13] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.

[VOC09] Francisco Ventura, A Oliveira, and Amílcar Cardoso. An emotion-driven interactive system. In *Portuguese Conference on Artificial Intelligence*, 2009.

[WB98] Ce Wang and Michael S Brandstein. A hybrid real-time face tracking system. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3737–3740. IEEE, 1998.

[WD14] Cheng-i Wang and Shlomo Dubnov. Guided music synthesis with variable markov oracle. In *The 3rd international workshop on musical metacreation, 10th Artificial intelligence and interactive digital entertainment conference*, 2014.

[WD15] Cheng-i Wang and Shlomo Dubnov. Pattern discovery from audio recordings by variable markov oracle: A music information dynamics approach. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 683–687. IEEE, 2015.

[WERA] Elliot Waite, Douglas Eck, Adam Roberts, and Dan Abolafia. Project magenta. Available at `https://github.com/tensorflow/magenta/tree/master/magenta/models/melody_rnn`.

[WES14] Felix Weninger, Florian Eyben, and Björn Schuller. On-line continuous-time music mood regression with deep recurrent neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5412–5416. IEEE, 2014.

[Wil20] Kevin Wilkinghoff. On open-set classification with l3-net embeddings for machine listening applications. In *28th European Signal Processing Conference (EUSIPCO)*, 2020.

[WW13] R Michael Winters and Marcelo M Wanderley. Sonification of emotion: Strategies for continuous display of arousal and valence. In *The 3rd International Conference on Music & Emotion, Jyväskylä, Finland, June 11-15, 2013*. University of Jyväskylä, Department of Music, 2013.

[WWL15] Ju-Chiang Wang, Hsin-Min Wang, and Gert Lanckriet. A histogram density modeling approach to music emotion recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 698–702. IEEE, 2015.

[YC12] Yi-Hsuan Yang and Homer H Chen. Machine recognition of music emotion: A review. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–30, 2012.

[YCY17] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR'2017), Suzhou, China*, 2017.

[YDL18] Xinyu Yang, Yizhuo Dong, and Juan Li. Review of data features-based music emotion recognition methods. *Multimedia Systems*, 24(4):365–389, 2018.

[ZGR$^+$18] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 570–586, 2018.

[ZWF$^+$18] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. Visual to sound: Generating natural sound for videos in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3550–3558, 2018.