Learning Disentangled Representations for Audio Classification

Nehemiah Skandera
UC San Diego
nskandera@ucsd.edu

Abstract

Much of this CSE 192 course has been centered around how we can manipulate and visualize the unique features of audio to develop our understanding of how audio signals behave. Thus, we see that digital audio manipulation can be a challenging task - especially as it relates to Machine Learning. Furthermore, we can generalize that the task of classifying audio is especially challenging. According to the information I have found in my limited research, typical audio classification methods struggle with details such as feature extraction, and the disentangling of more pronounced audio frequencies. This project has been a journey of taking steps to explore the combination of Convolutional Neural Networks (CNN) with Support Vector Machines (SVM), and taking ideas from other papers to improve the model - such as generating disentangled representations of data. The objective of this project is to develop a model that learns disentangled versions of content, rhythm, and pitch from audio signals in order to integrate the corresponding representation into a hybrid CNN-SVM framework that attempts to enhance/discover the performance of classification mechanisms.

1 Introduction

The classification of audio signals is fundamental to many real-world applications. For instance, we use audio classification in speech recognition, oceanic research, and music recognition. However, we often see the limits of these systems of classification; often, complexity is overlooked. What makes this project (which was mostly a process of discovery) unique is that it introduces a novel approach that combines CNNs and SVMs and leverages disentangled signal representations in attempting to improve the performance of audio classification.

2 Related Works

2.1 Learning Disentangled Representations for Timber and Pitch in Music Audio

Hung et al. [2018] Disentangling timber and pitch in audio signals forms the basis for understanding the importance of disentangled features in audio classification.

2.2 Unsupervised Speech Decomposition via Triple Information Bottleneck

Qian et al. [2021] This paper investigates the use of information bottlenecks (hence the name) for unsupervised speech decomposition. Thus, providing information on how we can disentangle audio representations and enhance the extraction of audio features.

3 Data Description

The dataset used in this project is gathered from room impulse responses (RIRs) in addition to noise sources from various databases. All of the data was gathered with a 16k sampling rate and 16-bit precision. Here is a description of the folders and data sources based on the data's README:

3.1 Real RIRs and Isotropic Noises

This set of real RIRs includes three databases:

- **RWCP Sound Scene Database**
- **2014 REVERB Challenge Database**
- **Aachen Impulse Response Database (AIR)**

Overall, there are 325 real RIRs. The isotropic noises available in these databases are used in conjunction with their associated RIRs. The data can be downloaded from the following links:

- AIR: `http://www.openslr.org/resources/20/air_database_release_1_4.zip`
- RWCP: `http://www.openslr.org/resources/13/RWCP.tar.gz`
- 2014 REVERB Challenge: `http://reverb2014.dereverberation.com/tools/reverb_tools_for_Generate_mcTrainData.tgz` `http://reverb2014.dereverberation.com/tools/reverb_tools_for_Generate_SimData.tgz`

3.2 Simulated RIRs

This folder contains simulated RIRs. Details regarding this dataset can be found in the following '$simulated_rirs/README$' file. The simulated RIR dataset can be downloaded here:

- `http://www.openslr.org/resources/26/sim_rir.zip`

4 Architecture

4.1 System Overview

The proposed system for my project integrates feature extraction using Fourier Transforms and creates a disentangled representation for learning through CNNs. Classification is then accomplished using a hybrid CNN-SVM model.

4.2 Components

- **Feature Extraction**: Fourier Transforms are used to extract frequency-domain features from audio signals.
- **Disentangled Representations**: The CNN-based encoder functions by extracting the content, rhythm, and pitch from audio signals.
- **Hybrid Model**: The CNN decoder functions by integrating the disentangled features. Meanwhile, the SVM classifier is used for further classification of the Fourier-transformed features.

4.3 Implementation Details

The system uses Python libraries such as 'joblib' for faster computing times, TensorFlow, scikit-learn, and scipy for audio processing. Key components of the system include data preprocessing, feature extraction, model training, and model analysis.

5 Experiments

5.1 Experimental Setup

The dataset used for the experiments is described in Section 3. The dataset uses both real and simulated RIRs, in addition to isotropic noises. The dataset was divided into training and testing subsets, and the system's performance was evaluated based on classification accuracy.

5.2 Methodology

The methodology can be broken into steps:

- **Step 1**: Download and extract the dataset
- **Step 2**: Load dataset RIR data
- **Step 3**: Extract Fourier transforms as features
- **Step 4**: Apply PCA for dimensionality reduction
- **Step 5**: Data Augmentation
- **Step 6**: Create Encoders
- **Step 7**: Create CNN Decoder
- **Step 8**: SVM Kernel System
- **Step 9**: Bringing everything together
- **Step 10**: System Evaluation

5.3 Evaluation Metrics

Accuracy is used as a metric to evaluate the performance of the CNN, SVM, and combined models.

6 Results

6.1 Results Presentation

- **SVM Accuracy**: The SVM classifier achieved an accuracy of 92% on the test set.
- **CNN Accuracy**: The CNN model achieved an accuracy of 61% on the test set.
- **Combined Model Accuracy**: The combined model achieved an accuracy of 85%.

6.2 Confusion Matrix

True \Predicted	RVB2014	RWCP	air
RVB2014	22	5	0
RWCP	0	40	0
air	0	8	9

Table 1: Confusion matrix showing the number of true vs. predicted classifications for each class.

6.3 Classification Report

Class	Precision	Recall	F1-score	Support
RVB2014	1.00	0.81	0.90	27
RWCP	0.75	1.00	0.86	40
air	1.00	0.53	0.69	17
Accuracy			0.85	
Macro avg	0.92	0.78	0.82	84
Weighted avg	0.88	0.85	0.84	84

Table 2: Classification report showing precision, recall, F1-score, and support for each class.

6.4 Plots

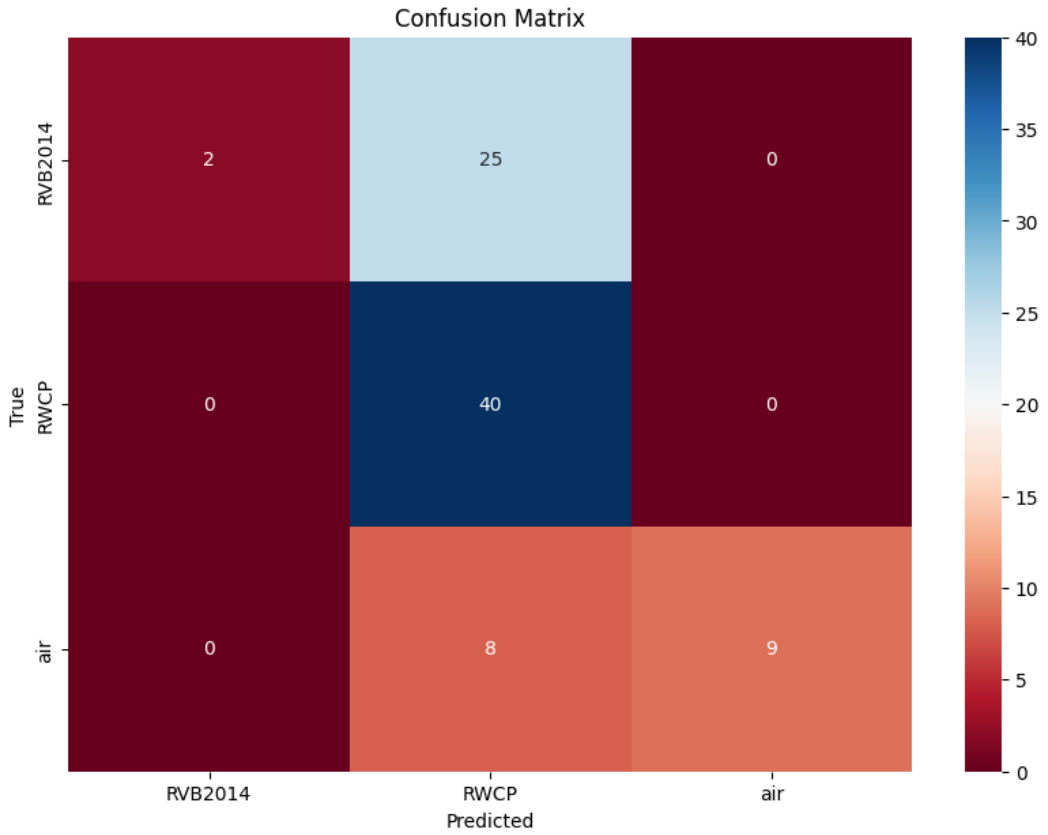

Figure 1: Confusion matrix heatmap showing true vs. predicted classifications.

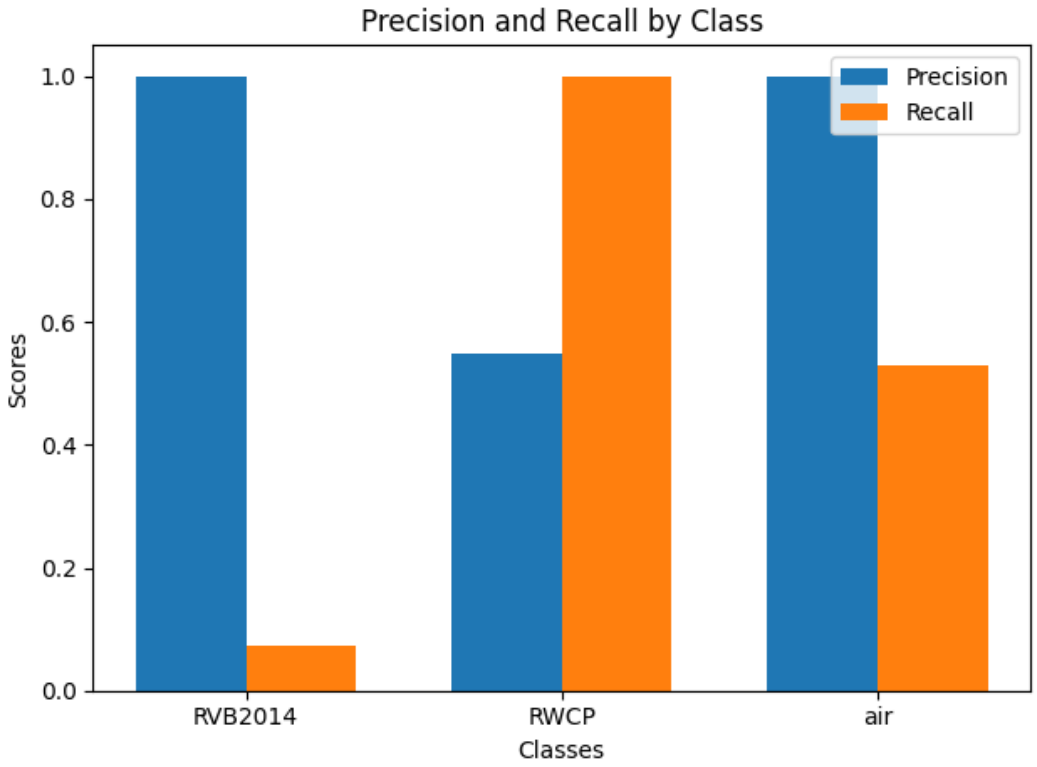

Figure 2: Precision and recall bar chart by class.

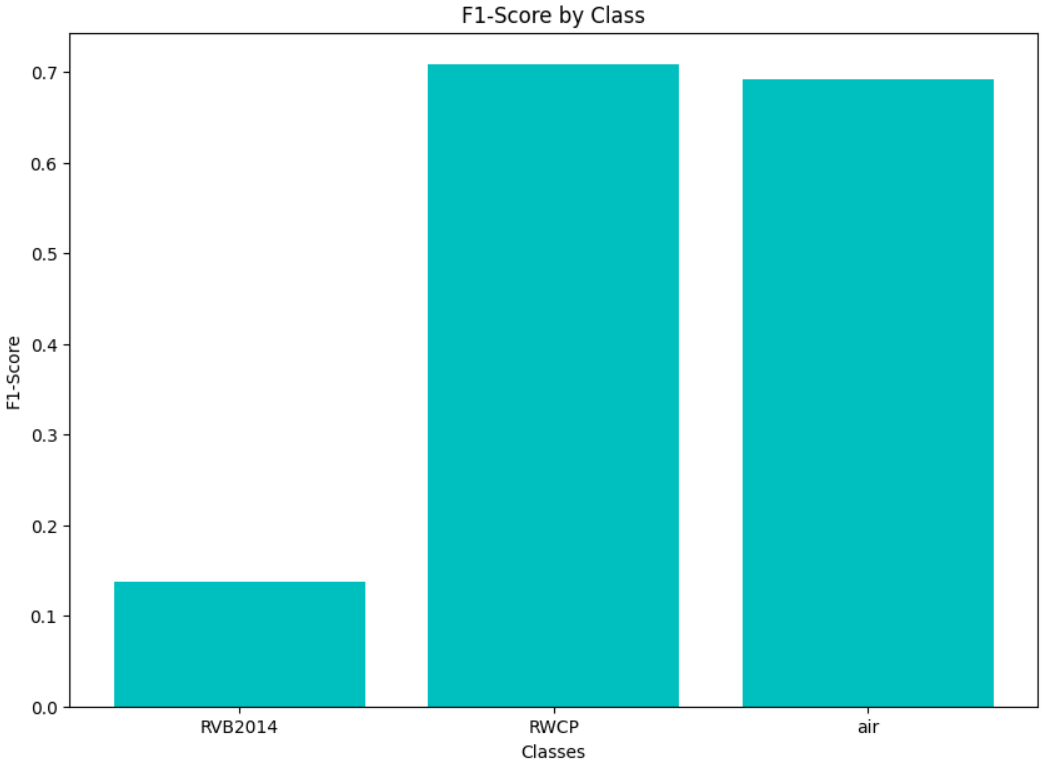

Figure 3: F1-score bar chart by class.

7 Conclusion

For the most part, the project was a process. I started with the idea of a simple CNN-SVM hybrid audio classification system. Over time, the project evolved. First, I was able to learn some new techniques for audio processing from a couple of papers, then I had to go through the process of making my models work cohesively together. This was the longest process. Finally, I had to find a way to train the model - given the computing resources I had. Initially, I was hoping to gain access to the San Diego Supercomputer (SDSC). However, this proved to not be an option. Thus, I learned several ways to optimize the resources that I had to perform computations. Thus, we arrive at the end product. I believe my project demonstrates the potential effectiveness of combining CNNs with SVMs for audio classification of disentangled signals. Overall, I was pleasantly surprised by the overall accuracy result of 85%. Future work could be done in exploring additional feature extraction methods and more complex hybrid models. Maybe this could eventually lead to instrument recognition?

8 Supplementary Materials

8.1 Code

The complete code for this project is available at `https://github.com/NehemiahSkan/CSE_192_Project`.

References

Yun-Ning Hung, Yi-An Chen, and Yi-Hsuan Yang. Learning disentangled representations for timber and pitch in music audio. *Research Center for IT Innovation, Academia Sinica, Taiwan; KKBOX Inc., Taiwan*, 2018.

Kaizhi Qian, Yang Zhang, Shiyu Chang, David Cox, and Mark Hasegawa-Johnson. Unsupervised speech decomposition via triple information bottleneck. https://arxiv.org/abs/2004.11284, 2021.