Kusman, Matthew Richard

matthew.kusman@wsu.edu

Sound Emotion Recognition

Through Ml and ravdess data

Contents

[Introduction 2](#_Toc90130164)

[Real World Application 2](#_Toc90130165)

[Questions Sought to Answer 2](#_Toc90130166)

[Personal Motivation 2](#_Toc90130167)

[Challenges Faced 3](#_Toc90130168)

[Results 3](#_Toc90130169)

[The Task 4](#_Toc90130170)

# Introduction

The topic of this project is Sound Emotion Recognition (SER), the practice of extracting features from audio to determine emotion carried behind the delivery of the audio. SER is part of a broader Emotion Recognition objective, which seeks to accomplish the same goal through several additional features such as facial expression and body language.

## Real World Application

SER was one of the first machine learning feats and is present in various industries. Emotion is still very unknown and applying emotion to a computational approach is even more complex. However, with the use of machine learning, SER can be achieved through supervised learning.

SER can be seen around the world in various applications such as:

* Human-like AI systems
* Chat bots
* Security systems
* Psychology applications
* Driver fatigue monitoring
* Remote elderly health monitoring
* Voice to text
* Virtual assistants

SER exists in several other applications around the world. While its usage is common, ML SER is still not as strong and capable as most humans. The process for accurately employing Sound Emotion Recognition needs continual improvement to be on the same level as the human brain in determining others’ emotion.

## Questions Sought to Answer

In creating this project, the main questions I sought to answer were the features of speech and how emotion could be derived from them, and the accuracy of a simple algorithm and machine learning process. In my time carrying out the project, I certainly gained insight into answers for these questions, though there is certainly more for me to learn. With more exploration, I would like to find ways to make the prediction of emotion slightly more accurate and employ it in a real-world system such as AI.

## Personal Motivation

When seeking various data mining projects, I knew instantly I wanted to explore machine learning more. Until recently, I had no experience with it and was blown away by the applications of it. Upon learning the simplicity of some ML algorithms, my interest grew: “How could answers so complex come from such a simple process”. While there are certainly more complex implementations, my focus has been on more straightforward approaches that can, albeit tediously, be done by hand.

I have also been fascinated by AI and application of ML in AI. While our lives grow more and more dependent on AI, we grow closer to developing realistic human like AI. Because humans and their though process are so complex, emotion is very difficult to understand computationally. When I began reading about SER, I wanted to tackle such a project that could be applied to human-like AI on an elementary level.

## Challenges Faced

When I first decided to take on the project, I was very confused and had no idea where to start. My first challenge was deciding what dataset to use, and how to process it. I very quickly found the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset. Then, I had to decide my algorithmic approach to extract features from the data. After achieving that, I encountered issues with the sheer size of data and non-standard formats. All of these issues were resolved through algorithmic approaches and low complexity implementations.

## Results

After finishing the project and analyzing the results, I learned just how difficult it can be to determine emotion through speech. I discovered the less emotions available to choose from, the higher the accuracy. In general, my implementation achieved a ~70% accuracy rate for determining emotion given four options.

I saw varying results with different parameters to my classifier and different sizes of data.

# The Task

## An ML Approach to Sound Emotion Recognition

Using machine learning to tackle SER involved 2 main steps, as most supervised learning approaches do: Train a model, and use that model to predict some data.

### Training the Model

This step is the majority of the process. It involves reading the data, formatting the data and extracting features that will guide the algorithm, and using some model to fit the data.

The input data I used was the RAVDESS dataset, open to public use and containing 24.8 gigabytes of sound/video of 24 different actors speaking with various emotions. The videos were then rated by over 200 people to determine the emotion behind each. The dataset includes both speech and song, of which I focused on speech. I also used the .wav format of the audio to avoid unnecessary file size of the videos. Smartlabratory.org’s description of the data is:

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains 7356 files (total size: 24.8 GB). The database contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

To format the data, I used the SoundFile Python library, and read each .wav file as a soundfile object. This had its own challenges. Because the speech’s emotion is labeled in its filename, I had to use the glob library to determine the correct emotion from the filename. Then, I ran into an issue of file format: some files were stereo audio, but most were mono. I decided to disregard stereo-channel files and not include them in my model. Because there were only 6 in the entire dataset, I determined the change was negligible and therefore acceptable. Once properly formatted, I extracted the key features and stored them for use.

The process of extracting the features was my biggest challenge. After some research, I learned that three main features can be used to describe audio: the Mel-frequency cepstral coefficients, power spectrogram chromogram, and mel-scaled spectrogram (mfcc, chroma, and mel respectively). The Python library librosa has tools capable of extracting each of these, which I then stored in a numpy array.

Once all the data if read, formatted, and extracted, I used a Multi-layer Perceptron classifier (MLPC) from sklearn.neural\_network to fit the data. I found that the MLPC to be the most appropriate as it is an easy-to-use neural network classifier that accurately classifies data. Once fit, the model is able to predict the labels of data given the features.

### Predicting

Because of the MLPC’s ease of use, predicting was extremely simple. The model allows you to input the features through the predict function, which returns the predicted emotion labels. I scored the predicted labels using sklearn.metrics’ accuracy\_score function.

### Output

The program outputs two main data points: the overall accuracy of the prediction, that is what percentage of emotions it successfully predicted, and the accuracy for each emotion included. I chose to output this data because I was particularly interested in which emotions were the hardest to predict.