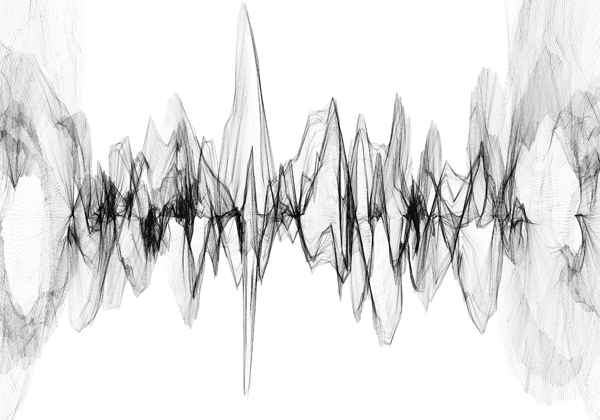


Proposal of Gender-based Vocal Recognition

Final Project of CSYE7374



#### Members:

Shiqi Dai

Wenqi Cui

#### Advisor:

Sri Krishnamurthy

# Topic Description

In Gender-based Vocal Recognition, we will design a classifier which is used to recognize the gender of a speaker from a given speech audio file(wav format), in other words, to classify audio files to genders(male and female).

The data source is AudioSet provided by Google Research. It is a large scale audio dataset collected from youtube videos. We will build machine learning models with TensorFlow, trying some wildly used neural networks such as CNN, RNN. Different models for the model evaluation will be taken to obtain the best model. Afterward, our product will be able to process the newly coming wav files, automatically applying feature engineering and generating training examples in memory-friendly data format. Finally, use our pre-trained neural network model to label it for future research usage.

Achievement of our project could be very valuable on video/audio labeling, and it will help to generate more data for further study. For example, building a separator to separate a conversation audio file into two part according to different speakers. Combined with IoT technology which becomes more and more popular these days, our project is helpful in empowering “smart home”. For example, virtual assistants, such as Siri or Alexa, can use our product to validate if the voice of a certain user has access.

# 

# Data Sources

## [AudioSet](https://research.google.com/audioset/index.html)

AudioSet is provided by Google Research. It consists of an expanding ontology of 632 audio event classes and a collection of 2,084,320 human-labeled 10-second sound clips drawn from YouTube videos.

We will use four parts of data from AudioSet. They are:

1. **Male speech, man speaking:** <https://research.google.com/audioset/ontology/male_speech_man_speaking_1.html>
2. **Female speech, woman speaking:** <https://research.google.com/audioset/ontology/female_speech_woman_speaking_1.html>
3. **Child speech, kid speaking:** <https://research.google.com/audioset/ontology/child_speech_kid_speaking_1.html>
4. **Conversation to be separated:** <https://research.google.com/audioset/ontology/conversation_1.html>

In which, 1, 2, and 3 will be used to train classifier(s) that recognize different types of human voice. The last dataset(No.4) will be used as input to the separator.

# Project Pipeline

### Data Collection

Google Research offers the AudioSet dataset for download in two formats:

1. Text (CSV) files describing, for each segment, the YouTube video ID, start time, end time, and one or more labels.
2. 128-dimensional audio features extracted at 1Hz. The audio features were extracted using a VGG-inspired acoustic model described in [Hershey et. al.](https://research.google.com/pubs/pub45611.html), trained on a preliminary version of [YouTube-8M](https://research.google.com/youtube8m/index.html). The features were PCA-ed and quantized to be compatible with the audio features provided with YouTube-8M. They are stored as TensorFlow Record files. The model used to generate the features is available in the [TensorFlow models GitHub repository](https://github.com/tensorflow/models/tree/master/research/audioset) (see below for more details).

We will use the extracted features dataset. The total size of the features is 2.4 gigabytes. They are stored in 12,228 TensorFlow record files, sharded by the first two characters of the YouTube video ID, and packaged as a tar.gz file. The labels are stored as integer indices. They are mapped to sound classes via class\_labels\_indices.csv.

The first line defines the column names:

|  |  |  |
| --- | --- | --- |
| *index* | *mid* | *display\_name* |

Subsequent lines describe the mapping for each class. For example:

|  |  |  |
| --- | --- | --- |
| *0* | */m/09x0r* | *"Speech"* |

This means the index 0 indicates the label “Speech”.

There are two ways provided by Google to download the features dataset:

1. Manually download the tar.gz file
2. Use gsutil rsync, with commands

### 

### Data Preprocessing

Data preprocessing will be a vital part of our project. There are two reasons: First, machine learning is always affected deeply by the feature representation. The preprocessing plays an important role in representing the most remarkable feature of our data helping the classifier work efficiently and successfully. More importantly, audio data contains a large amount of information. Without preprocessing, there are not enough computational resources for us to train classifiers later.

**Preprocessing on the downloaded dataset**

For downloaded dataset, audio data(frame-level features) is stored in *tensorflow.SequenceExample* protocol buffers in TFRecord files.

*A SequenceExample is an Example representing one or more sequences and some context. The context contains features which apply to the entire example. The feature\_lists contain a key, value map where each key is associated with a repeated set of Features (a FeatureList). A FeatureList thus represents the values of a feature identified by its key overtime / frames.*

All the feature information is extracted and represented already as embeddings by a pre-trained neural network model as feature extractor.

**Preprocessing on the future data**

For future data which needs to be predicted, we assume the audio file is in the type of WAV. For WAV files, we would use the same strategy to do the feature engineering, extracting features and saving them as *tensorflow.SequenceExample* to TFRecord files. This process will follow the steps below, a WAV file is:

1. read in with the API provided by “soundfile” package.
2. converted into spectrogram examples
3. fed into the pre-trained VGGish model
4. post-processed to generate whitened and quantized embeddings(represented features)
5. Written in a *SequenceExample* to a TFRecord file (using the same format as the embedding features released in AudioSet)

The benefits of choosing TFRecord to store features is explained by Thomas Gamauf at [here](https://medium.com/mostly-ai/tensorflow-records-what-they-are-and-how-to-use-them-c46bc4bbb564).

### Classifiers Training

**Model Selection**

The main approach for the classifiers training will be neural-network-based. Typical neural network types will include CNN(Convolutional Neural Network), RNN(Recurrent Neural Network). Certain customization will be taken if necessary.

***CNN(Convolutional Neural Network)***

*Typically, CNN is used on image/audio data. It plays a vital role in cognitive computing like image/voice recognition.*

***LSTM(Long Short-Term Memory Neural Network)***

*Long short-term memory (LSTM) units are units of a recurrent neural network (RNN). An RNN composed of LSTM units is often called an LSTM network.*

**Optimization**

We would compare different optimizers including SGD, SGD with momentum, Adam, RmsProp, etc.

### Evaluation

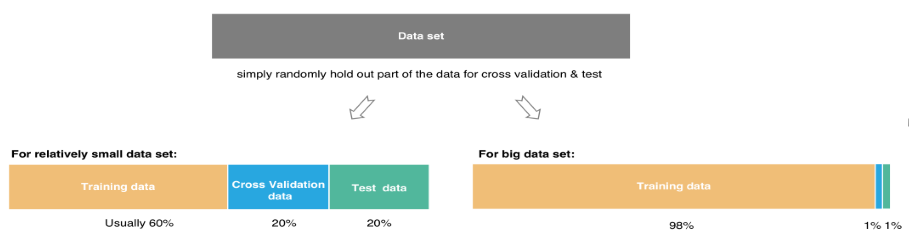
**Loss Function and Metrics**

Our task is a three-class classification(man, woman, and child). Because of the usage of artificial neural networks, we would choose categorical cross-entropy as our loss function.

For metrics, confusion-matrix-related metrics will be taken. This will include accuracy, recall, precision, and categorical accuracy.

**Testing and Validation**

Based on the scale of the provided data(17,716 man speaking, 8,513 woman speaking, 11,816 child speaking), we would choose hold-out validation.



# References

[1] Gemmeke, Jort F., et al. “Audio Set: An Ontology and Human-Labeled Dataset for Audio Events.” *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, doi:10.1109/icassp.2017.7952261.

[2] Hershey, Shawn, et al. “CNN Architectures for Large-Scale Audio Classification.” *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, doi:10.1109/icassp.2017.7952132.

[3] Ephrat, Ariel, et al. “Looking to Listen at the Cocktail Party.” *ACM Transactions on Graphics*, vol. 37, no. 4, 2018, pp. 1–11., doi:10.1145/3197517.3201357.

[4] Google. “Google/Youtube-8m.” *GitHub*, 5 Nov. 2018, github.com/google/youtube-8m.

DTaoo. “DTaoo/VGGish.” *GitHub*, 30 Nov. 2017, github.com/DTaoo/VGGish.

[5] “Looking to Listen: Audio-Visual Speech Separation.” *Google AI Blog*, 11 Apr. 2018, ai.googleblog.com/2018/04/looking-to-listen-audio-visual-speech.html.

[6] *Acapella Extraction with ConvNets*, madebyoll.in/posts/cnn\_acapella\_extraction/.

[7] *AIFF / AIFC Sound File Specifications*, www-mmsp.ece.mcgill.ca/Documents/AudioFormats/.

[8] Rogerdettloff. “Rogerdettloff/speech\_seg\_sep.” *GitHub*, 28 Sept. 2017, github.com/rogerdettloff/speech\_seg\_sep.

[9] Gamauf, Thomas. “Tensorflow Records? What They Are and How to Use Them.” *Medium.com*, Medium, 20 Mar. 2018, medium.com/mostly-ai/tensorflow-records-what-they-are-and-how-to-use-them-c46bc4bbb564.