Vocal Replacement of Audio Recordings

Dao Vang

github.com/dao-v/Vocal_Replacement_of_Audio_Recordings

5/12/2020

Table of Contents

Project Goal

- To extract the audio properties of the user's recordings
- Using Deep Learning, predict a new audio file with the user's voice replaced with a speaker's voice who was used in the training process

Sound & Audio Files

- When perceiving sound, we are interpreting the vibrations in the air molecules
  - When computationally interpreting sound, small clips of sound are saved as Samples, which is like the term Frames when recording video
- When converted from machine-readable files to human-readable NumPy arrays, the audio file contains information on:
  - Number of audio channels (stereo or mono)
  - Sample rate (number of samples per second)
  - Length of the audio recording
  - Total amplitude for a sample

Audio Channels

- When converted to NumPy arrays, the dimension of the arrays informs the programmer how many audio channels the audio file contains:
  - Stereo track audio files have a NumPy array dimension of (X, 2)
    - X = Total number of samples in the audio file
  - Mono track audio files have a NumPy array dimension of (X, 1)
    - In this project, only audio files that are mono were used because vocal recordings in industry are typically recorded in mono

Sample Rate & Duration of Audio Files

- Sample Rates are defined as the number of samples in each second of an audio file
  - Videos are recorded in Frames Per Second (FPS), which means that if a video was recorded with 60 FPS, there are 60 sequential images per second recorded
    - Even though the human eye can theoretically perceive an infinite number of frames, many people do not notice changes in the video quality when increasing above 60 FPS
  - Using the same concept, audio files are recorded in the same method
    - Sample rates can be represented as 44100 samples per second or as 44100 Hz
      - In this project, all audio files used had a sample rate of 16000 Hz
- The Length of an audio file can be determined by simple mathematics:
  - Length (in seconds) = Total Number of Samples / Sample Rate

Amplitude

- Audio files are typically loaded in as NumPy arrays in Python, where each array element represents one sample with the value representing the amplitude
- A sample amplitude is the height of the net soundwave at a specific time point
  - The net soundwave means the sum of all soundwaves present
    - If two people were talking, then the net soundwave is the summation of the soundwaves being created by the two people

Data Gathering

- All data used in training the model was obtained from: http://www.openslr.org/12/
  - Only the datasets labelled with “clean” was used
- This dataset contained audio recordings with speakers reading various texts from books and other sources with a good ratio of female and male speakers

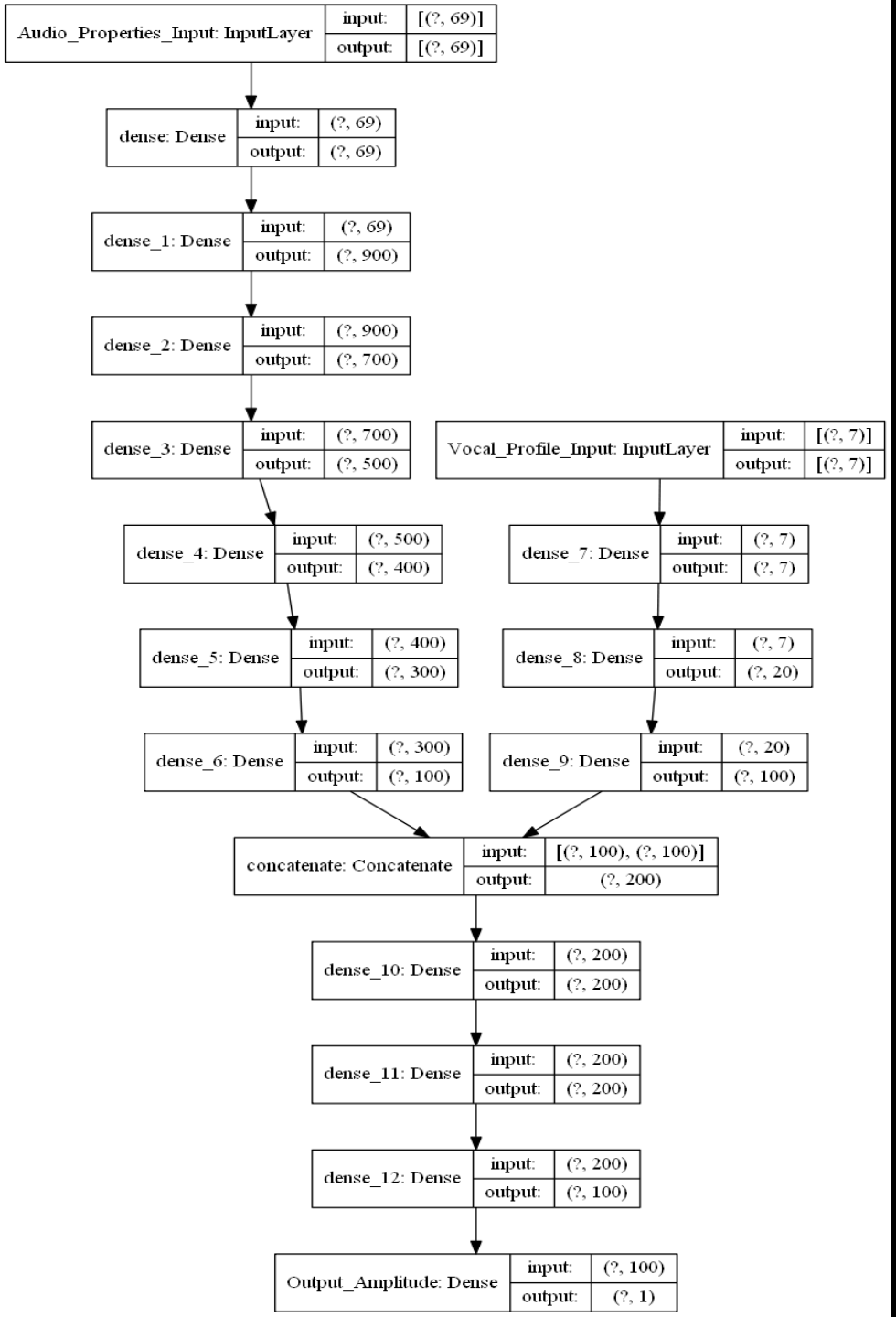
Data Preprocessing – Speaker Recordings

- To extract the audio properties of the dataset, pyAudioAnalysis was used to streamline the process for every sample
- The features extracted are depicted in the table, which was copied from the pyAudioAnalysis wiki page: github.com/tyiannak/pyAudioAnalysis/wiki/3.-Feature-Extraction

Feature ID	Feature Name	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The center of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.
22-33	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

Data Preprocessing – Vocal Profiles

- A vocal profile was created for every speaker, which contained the information based on the amplitudes in all the recordings associated to the speaker:
  - Minimum amplitude value
  - Maximum amplitude value
  - Mean amplitude value
  - Median amplitude value
  - Standard deviation



Deep Learning Model

- TensorFlow/Keras API was used to implement deep learning
- The model takes two inputs:
  - Speaker Vocal Profile
  - Audio Properties of the Recordings
- Then the model merges the two pathways to produce one output, a predicted amplitude value
  - This was expected to mimic reverse engineering from audio properties to amplitude
- The number of hidden layers and nodes were ideally set to provide flexibility in the learning process as amplitude values are a large range

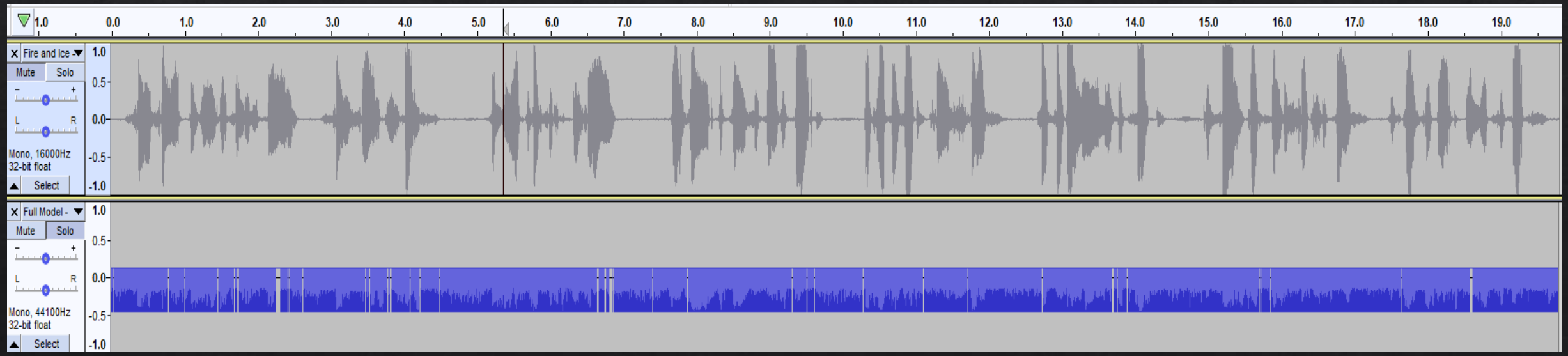
Results

- Predicting the amplitude values using a vocal recording reading “Fire and Ice” by Robert Frost as the audio properties input and the vocal profile of speaker 61 from the test-clean.tar.gz dataset resulted in a distorted audio file with faint hints a voice:
- Predicted Audio File: 
- Original Audio File: 
- Speaker’s Voice: 

Note: To hear the audio files, view the PowerPoint version of this presentation or listen to the audio files uploaded in the GitHub Repository.

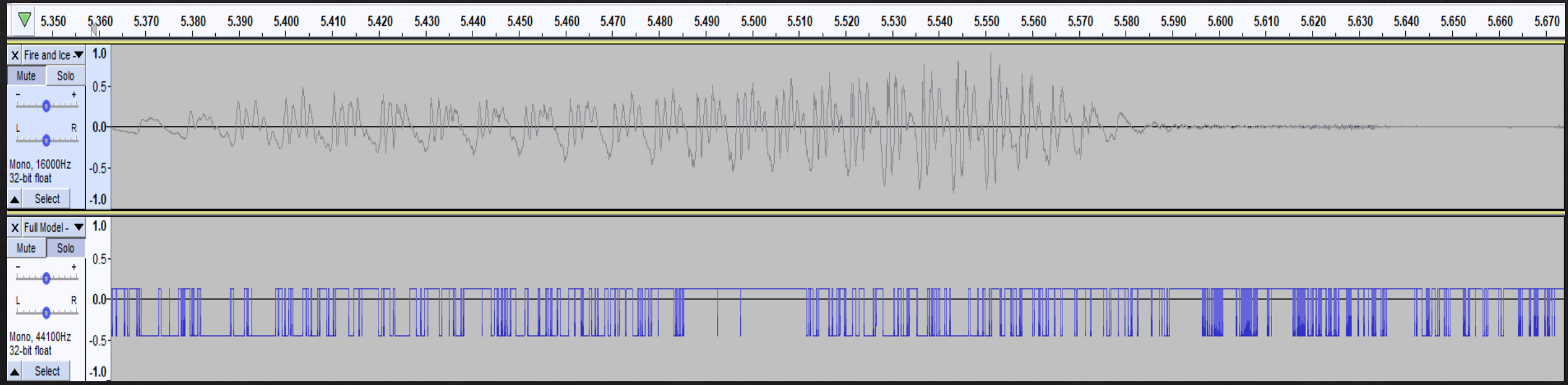
Results

- Examining the waveforms in Audacity:
  - The top track is the original recording and the bottom track is the predicted recording after normalizing
- The model was not able to predict silence (zero amplitude)
- Predicted silence in samples where there were high amplitude values in the original



Results

- Zooming in, the model seemed to mostly predict two main values at the minimum and maximum
- The shape of the predicted waveform did not represent the shape of typical audio waveforms



Conclusion

- Although the model failed to produce the expected outcome, there was a faint voice-like sound that could be heard sporadically indicating that the approach could work after adjustments to the data and model
- If re-approaching the goal of this project, this project can serve as starting point on how to design a new approach