Voiceprint Identification

Alastair Noble, Harley Latsky, Eli James, Alex Beamish

QMIND – Queen's AI Hub
Queen's University, Kingston, Ontario K7L 3N6, Canada. Queen's

1-email alastair.noble@queensu.ca
2- email harley.latsky@queensu.ca
3-email 19ewj@queensu.ca
4-email 19akb3@queensu.ca

Abstract: *Voiceprint recognition is the process by which a trained model identifies the specific individual who is speaking arbitrary phrases. Speaker recognition is distinct from speech recognition, which involves converting spoken words to text, but these two types of systems are often combined in practice, as is the case with virtual assistants such as Siri, Alexa and Google Assistant. Currently, largely due to the training approaches used, many speaker recognition models can only identify speakers when a specific phrase is uttered by the speaker. In an attempt to address this issue, and to create a complete voiceprint recognition system, we have developed a multi-faceted software application in python. The application includes a neural network, a set of features that allow for flexible training on multiple words and phrases, and a user-friendly GUI. Our application can identify speakers with roughly 80% accuracy, although we are currently working to improve that. We plan on continuing to experiment with different approaches for training, with the goal of maximizing identification accuracy even when the speaker uses novel words and phrases. Future uses of our software could include audio captioning/transcription, and authentication for security or customer service purposes.*

1. INTRODUCTION

1.1 Motivation

Accurate voiceprint recognition systems are becoming increasingly important to develop more accessible and secure technology [1]. Voiceprint recognition models are used in areas such as voice assistants, two-step authentication security systems and customer service [2]. Two areas in which existing voiceprint recognition systems could improve are their ability to accurately differentiate between two speakers and their ability to accurately identify whether it is the same person speaking but under different conditions. These systems identify more than just speech itself; they are also taking background noise, echoes, and other sound features into account, all while constrained by the hardware upon which the sound source is recorded.

We sought to build a voiceprint recognition model which could accurately differentiate between speakers in real time.

1.2 Related Works

In the past, the most common approaches to voiceprint recognition and verification models were Gaussian Mixture Models (GMMs) and hidden Markov Models (HMMs). The more common approach currently, and the one we developed, was using a convolutional neural network (CNN). The advantages and disadvantages of these approaches can be summarized by Mingyu Ma's paper [3], or by this Microsoft Research Paper [4] from 2014. One advantage of using a CNN is that it promotes more flexibility in handling the natural variability in speech. For a few years, the CNN has been known to be an effective approach for identifying and verifying voice using machine learning.

1.3 Problem Definition

Using a CNN, we sought to build a model which could accurately identify speakers in real time using multi-phrase voiceprint recognition with the aim of discovering improved ways of collecting and cleaning

training data to maximize the accuracy of identification.

2. METHODOLOGY

2.1 Prototype

In building a multi-phrase voiceprint recognition system, our team started by implementing a single-phrase system. This initial system was modelled after Jurgen Arias's work on voice classification, and used Librosa, Keras (sequential neural network), and MFCCs (Arias, 2019). To fully test our initial single voice classification system, we used data from an open-source Alexa dataset containing 86 users saying the word Alexa 4 times. Our group added data of ourselves saying Alexa 4 times to add to this data and ensure that the model is working through testing.

2.2 Data Processing

To convert audio to data that can be understood by the model, our group split the original audio file into 40 different audio chunks. For each chunk, a coefficient, namely a Mel-Frequency Cepstrum Coefficient (MFCC) was created. This MFCC was generated based on the frequency of the audio chunk put through a Fourier transform to bias the coefficient towards small changes in frequency. This created a 40 long decimal array that was fed into the model.

Before the training of the model, the data needed to be processed in order to avoid overfitting and ensure that the model focused on the correct indicators. An example of a situation that we wanted the model to avoid paying attention to is the time in-between the start of the recording and the first sound. To prevent the model from focusing on this, we utilized Librosa to add a hamming window to our data, which softened out large changes in audio. We also used a noise removal formula to help limit the bias of background noise from different microphones. Using the single phrase system, our team tried to increase the efficiency of the model and create code that can be expanded upon with future iterations of the model.

2.3 Final Design

After obtaining satisfactory results with the single phrase system, our team split up to build out as many features as possible for the final design. A data collection system that works with the previous model requires sentences to be split up into words. Our team used PyDub (the python library) to analyze full sentences and return separate audio files of all the individual words used. This relied on the silences in-between words and greatly sped up the data collection process. Next, we increased the accuracy of the model by iterating on the model. For example, the audio chunk number of 24 was determined a better indicator than the original 40. The last task was to bring all these pieces together into an application with a simple UI. The most significant feature of the UI is to test the model live using live input from a microphone while continually updating the UI.

3. RESULTS AND DISCUSSION

We trained the final model on 4 subjects using 35 seconds of speech data per subject. We broke the data up into between 60 and 70 audio chunks based on how fast the subject spoke. We evaluated the model in 2 ways: using prerecorded test data of each subject, and with live input during a meeting between the 4 subjects. The model had 88% accuracy on the test data. It averaged 80% accuracy during the live input, however it varied with different conversations. During presentations, when subjects spoke one at a time without interruption, the model reached 88% accuracy, however during regular meetings where there was a lot of back and forth, the accuracy averaged 72%. The lower accuracy during the live input was expected for a few reasons. First, conversations between subjects often have audio input from more than one party, in the form of background noise or talking over each other. And second, we had to collect the audio over set intervals to update the GUI, so we had to balance the speed of the prediction in the GUI with the quality of the audio chunks we collected.

We chose to only use 35 seconds of audio data for training because we prioritized usability during

meetings and presentations, and we wanted to be able to easily add subjects to the training set during the demonstration at CUCAI 2021. For different applications with more subjects or where higher accuracy is required, more training data should be used so that the model has more data to learn the unique identifiers of each subject.

A limitation we had was that we were displaying the predictions live. This meant we had to cut audio into intervals before breaking it up naturally, so we often had words cut in half at the beginning and end of intervals. For different applications, for example transcription, the audio would be broken up more naturally after it was all recorded, and would have better accuracy.

4. CONCLUSIONS AND FUTURE WORK

The progression from a single phrase model to a multi-phrase model went smoothly, and the final model performed well considering the limited amount of speech data that was used to train it. In the final stages of development, various software components were successfully merged into a usable application. As outlined in the previous section, there are certain limitations on what the final model can do, and more robust testing under various conditions is still required. Nonetheless, results so far have been very promising, and there do not appear to be any insurmountable barriers to further improvement of the model. Future work could include continuing to search for better phrases to train the model on, and continuing to optimize the identification accuracy through tuning of the model parameters.

Many of the major applications of voice identification technology are in authentication for security and customer service purposes, and in audio captioning. In the security and customer service space, our application could be used in conjunction with other methods to allow for real-time, continuous verification of identity while a speaker is speaking. Our application could also be used in digital audio captioning to identify a speaker across multiple audio files. Ultimately, speaker identification is a very general problem, so our application could be useful in many different contexts.

REFERENCES

[1] S.M. Schwartz (2017). Multi-Agent Path Planning for Lo-cating a Radiating Source in an Unknown Environment.Master's Thesis, Department of Mechanical Engineer-ing, Embry-Riddle Aeronautical University.

[2] M. Vrigkas, C, Nikou, and I. Kakadiaris, "A Review of HumanActivity Recognition Methods", Front. Robot. AI, 16 November 2015.

[3] M. Alzahrani, S. Kammoun, "Human Activity Recognition: Chal-lenges and Process Stages",International Journal of Innovative Re-search in Computer and Communication Engineering, Vol. 4, Issue 5,May 2016.

[4] C. McDaniel, S. Quinn, "Developing a Start-to-Finish Pipeline forAccelerometer-Based Activity Recognition Using Long Short-TermMemory Recurrent Neural Networks",PROC. OF THE 17th PYTHONIN SCIENCE CONF, 2018

[5] J. Arias, "Voice Classification," *GitHub*, 07-Dec-2019. [Online]. Available: https://github.com/jurgenarias/Portfolio/tree/master/Voice%20Classification. [Accessed: 17-Mar-2021].