Text Independent Speaker Authentication in Noisy Environments

MLND Capstone Project

Abdullah AlOthman

March 2018

1 Definition

1.1 Project Overview

A Deep Neural Network That, given two audio inputs (A , B) verifies whether the speaker in A and the speaker in B are the same person.

1.2 Problem Statement

Most implementations of speaker verification systems are dependant on specific phrases (e.g. " Hey, Google..", "Hey, Siri..", "Alexa..") and consistant audio-environment (same microphone, audio quality, etc). What I want to tackle in this project is text-independent speaker verification, in noisy, uncontrolled environments.

Applications of speaker verification include:

- authentication in high-security systems
- authentication in forensic tests
- searching for persons in large corpora of speech data

All of these applications require reliable performance in 'wild' conditions (i.e. real-world scenarios). The challenges expected in this task come from two main fronts:

1. High variance in the environment (background music, crowds, recording quality, etc.)
2. High variance in the speaker age, accent, emotion, intonation, etc.

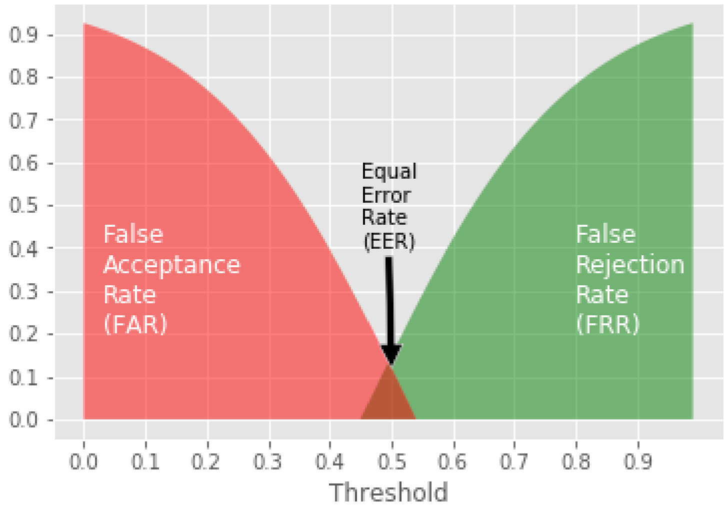


Figure 1: EER Visualized in an Example

1.3 Metrics

To evaluate the model, I'm choosing Equal Error Rate (EER) as the performance metric. For each authentication pair the model outputs a continuous variable X, which is the estimated probability of a match.

Given a threshold parameter T, the instance is accepted as a match if $X > T$, and rejected otherwise. X follows a probability density $f_1(x)$ if the instance is actually a match, and $f_0(x)$ otherwise [4].

The false rejection rate is given by

$$FRR(T) = 1 - \int_T^\infty f_1(x)\,dx \tag{1}$$

and the false acceptance rate is given by

$$\mathrm{FAR}(T) = \int_T^\infty f_0(x)\,dx \tag{2}$$

.

EER is defined as the rate at the threshold T where $FRR(T) = FAR(T)$ (see Figure 1) it is commonly used by verification systems, including our benchmark model.

number of speakers		1251	
number of male speakers		690	
	max	average	min
videos per speaker	36	18	8
utterances per speaker	250	116	45
length of utterances (seconds)	145.0	8.2	4.0

Table 1: dataset statistics

2 Analysis

2.1 Data Exploration and Visualization

for this problem, I'm using the VoxCeleb dataset[3]. VoxCeleb dataset was created by extracting audio from YouTube videos. it contains 100000 utterances for 1251 celebrities. The speakers are gender balanced and span a wide range of different ethnicities, accents, and ages.

These dataset includes varying audio qualities and challenging acoustic environments, such as:

- red carpet interviews
- outdoor stadium
- studio interviews
- speeches with large audience
- high production multimedia
- amateur multimedia (e.g. Vlogs)

The dataset includes uncontrolled, real-world noise such as background-chatter, audience reactions, room acoustics, etc. Table 1 and Figure 2 give more statistics on the dataset.

2.2 Algorithms and Techniques

Since speech is inherently temporal, I decided to take that into account and chose a Recurrent Neural Network as the basis of my model. The big drawback that one might face when using neural networks in general is not having enough data, which fortunately is not an issue in this case.

To be able to verify, you must first be able to identify. The first step I will do is train a model that identifies the speaker in a given audio file, once the identification model is trained, I'll use it as a base for the siamese network that will handle the classification task.

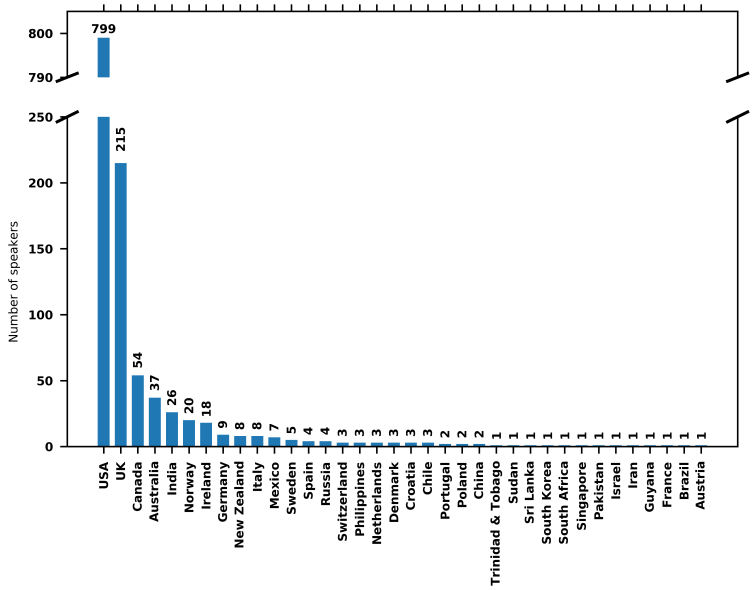


Figure 2: Nationality distribution

2.3 Benchmark

I'll be comparing my model to multiple benchmarks.

1. Random Classifier: the most basic benchmark, but important to know we're preforming better than random chance.
   - EER = 0.50
2. The GMM-UBM model proposed in the VoxCeleb paper.
   - EER = 0.15
3. The Embedding model proposed in the VoxCeleb paper.
   - EER = 0.08

3 Methodology

3.1 Data Preprocessing

since the audio length is not consistent, I only considered the first 3 seconds of each utterance as inputs to the model.

for feature extraction I used MFCC. Mel Frequency Cepstral Coefficents (MFCCs) are the state-of-the-art method for extracting audio features for speech and speaker recognition tasks. it is applied by first framing the signal into overlapping frames (standard parameters are 25ms length with 10ms ovelap). Then applying log filterbank on periodogram estimate of the power spectrum. Then finally taking the Discrete Cosine Transform (DCT) coefficients of the log-filterbanks.[2]

Given the noisy state of the dataset, I then normalized with respect to the mean and the variance to get the CMVN features, Figure 3 shows how normalization boosts the signal to noise ratio.

3.2 Implementation

I first attempted to tackle the verification task directly using a convolutional neural network. That effort proved fruitless as I got results that were worse than random. I then started working on an identification model, using the same train/val/test split used in assessing the base model. I attempted to use different features such as the signal's spectogram and the logfilterbank but they were computationally expense to extract and store, which is how I finally settled on using MFCCs.

I've been having no luck with CNNs even in the identification task, so I used the same architecture that I've been using in Kaggle's Toxic Comment Classification Challenge even though they operate on different data types (text / audio) and to my delight, the model actually preformed well (see table 2).

The identification model's architecture is as follows:

- input layer
- Spatial Dropout layer: prevents the neural network from over-fitting by deactivating feature maps with a given probability
- bidirectional RNN: takes into account the temporal relation between the input features (the value of X_t affects the value of X_{t+1})
- concatenated global max pooling and global average pooling
- Dropout: prevents the neural network from over-fitting by deactivating neurons with a given probability
- Batch Normalization: forces the activations of the previous layer to have a gaussian distribution [1]
- output layer with softmax activation

see figure 3.2 for the architecture and table 2 for the results of the identification model. For verification, I first created training pairs by creating balanced positive samples and negative samples for each utterance, where utterances must come from different video sources to be in a pair (for added difficulty). I then reserved a set of speakers for testing, these speakers are not included in the training set for either the verification or the identification models. After that I froze the weights of the identification model and used them for my verification

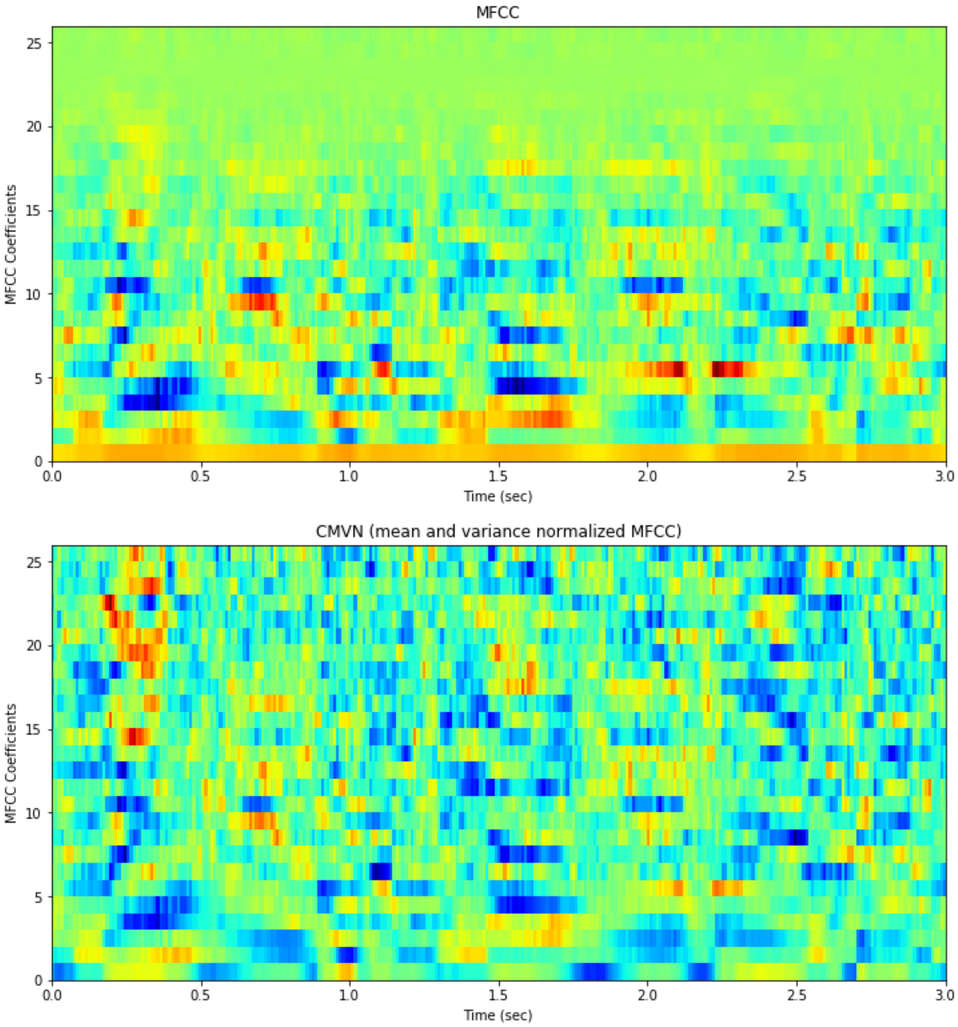
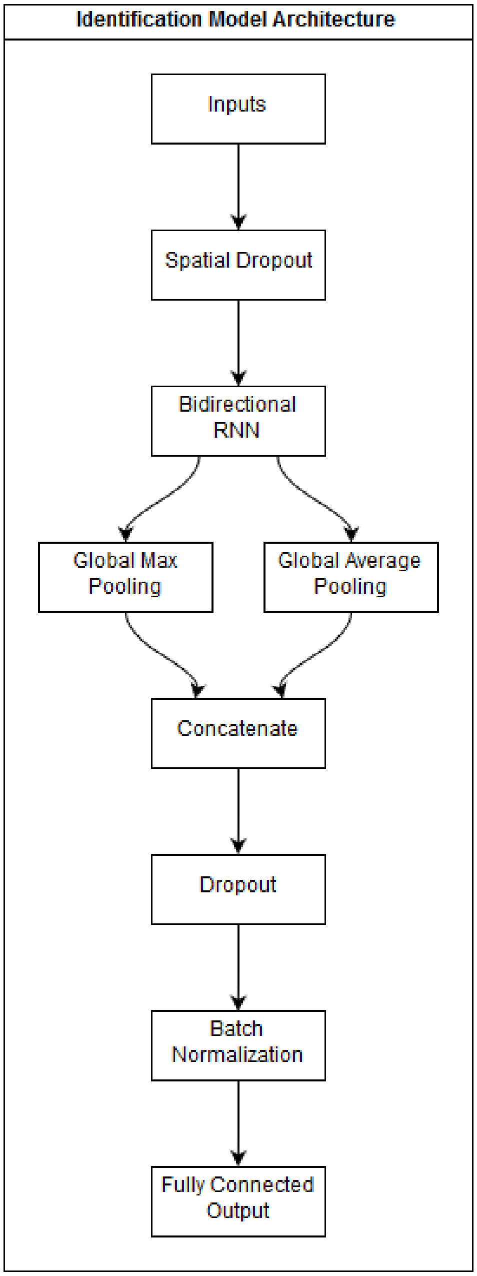
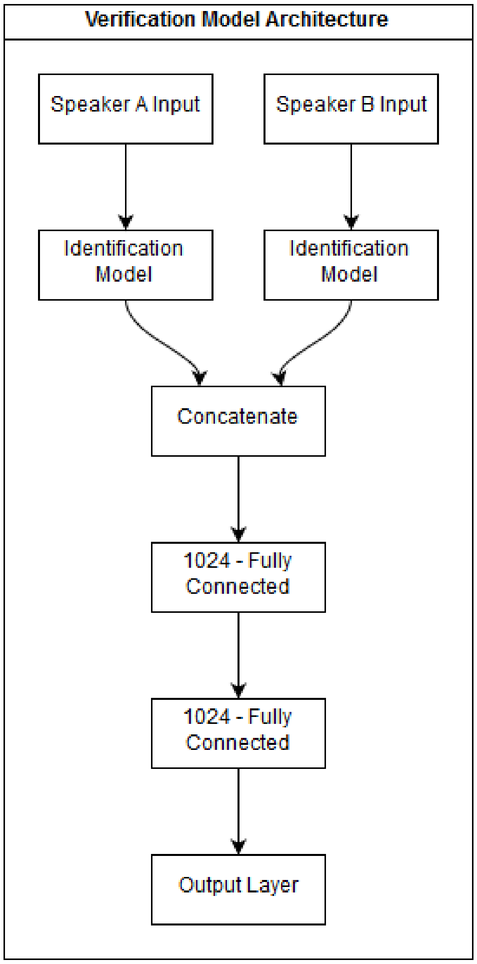


Figure 3: MFCC before and after normalization

Model	Top - 1	Top - 5
GRU - MFCC	0.56	0.76
GRU - CMVN	0.66	0.84
LSTM - CMVN	0.68	0.85

Table 2: Top-1 and Top-5 Accuracy of the identification models


Identification Model Architecture



architecture (see figure 3.2) I used ReLU activation for the middle Dense layers and Sigmoid activation for the final layer

3.3 Refinement

Moving from CNNs to RNNs gave the most dramatic improvement, beyond that table 2 show the improvements of using CMVNs instead of MFCCs and the marginal improvement of using LSTM over GRU

Model	EER
GRU-based siamese network	0.15
LSTM-based siamese network	0.14
using ensemble blend of both	0.13

Table 3: results of the verification models

4 Results

4.1 Model Evaluation and Validation

The final model results can be seen in table 3 since the test set contains no overlap in terms of video or speaker with the training set, I would say that the model is generalizes well with new data and that these results can be trusted.

4.2 Justification

the final solution is described in full in section 3.2 The final results, as seen in table 3 are better than 2 of the 3 benchmarks chosen for comparison. the model is robust enough for the uses proposed in section 1.2 given more data as input, using longer utterances for example will reduce the likelihood of miss-classification exponentially.

5 Conclusion

In this project, a recurrent neural network that determines whether two audio segments come from the same speaker has been developed. As it currently stands the model can still be used in production as mentioned in the previous section. The model exceeded 2 of the 3 proposed benchmarks which means that while the model has preformed well, there's still room for improvement.

5.1 Improvement

due to time and computational resource constraints, I've not been able to experiment with some of the approaches I've planned on trying out.

Possible venues of improvement include:

- instead of building the training set randomly, consider using hard negative sampling for a more robust model.
- use segments longer than 3 seconds, or use multiple overlapping segments from each utterance
- invistigate different possible feature representations. such as i-vector, d-vectors and full spectograms
- revisit CNNs, I feel like I'm missing something here that will become clear with more research

References

[1] François Chollet et al. Keras, 2015.

[2] James Lyons. Mel frequency cepstral coefficient (mfcc) tutorial, 2013.

[3] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[4] Receiver operating characteristic. Receiver operating characteristic — Wikipedia, the free encyclopedia, 2018. [accessed 15-March-2018].