Classifying Real and Deepfaked Presidential Voices Using Signal Processing and Machine Learning Techniques

Tracy Strickel

Stakeholder - The Associated Press

AI Voice Detection to Aid Journalists with Fake News Identification

Recently, increased access to AI has made it trivially easy for everyday people to create deepfakes of politician and celebrity voices.

- ‘Artificial intelligence makes voice cloning easy and ‘the monster is already on the loose” - Fortune, Feb. 11, 2023
- ‘TikTok videos are using AI tools to turn Biden, Trump, and Obama into Discord goblins’ - Polygon, Feb. 22, 2023

Stakeholder - The Associated Press

This new technology has prompted much concern among political journalists:

- ‘New AI voice-cloning tools ‘add fuel’ to misinformation fire’ - Associated Press, Feb. 10, 2023
- ‘Political Media’s Next Big Challenge is Navigating AI Deepfakes’ - Vanity Fair, Mar. 6, 2023
- ‘AI Voices Are Hilarious, Haunting, and Possibly Politically Dangerous’ - The Atlantic, Mar. 3, 2023

The Associated Press is very focused on maintaining their news integrity.

Project Summary

- In this project, I combine signal processing techniques developed for voice analysis and transmission with statistical and machine learning models to classify real and deepfaked presidential voices.

- I use four “speaker” classes:
    - Human Donald Trump
    - Deepfaked Donald Trump
    - Human Joe Biden
    - Deepfaked Joe Biden

Data Collection

Human voices: YouTube and the UVA Presidential Speeches Archive.
AI voices: self-generated samples via ElevenLabs and audio stripped from videos, including:

- ‘Joe Biden Issues Executive Order For Creation of More Shiny Pokemon’
- ‘Donald Trump Complains About Call of Duty: Warzone’
- ‘Joe Biden Praises British Drill Rap Music’

I split each audio file into five-second clips. Before the audio clips can be analyzed by the models, they must be converted to a numeric format.
I tried two conversion methods: MFC coefficients and LPC coefficients.

Data Processing and Analysis: Mel-Frequency Cepstral Coefficients

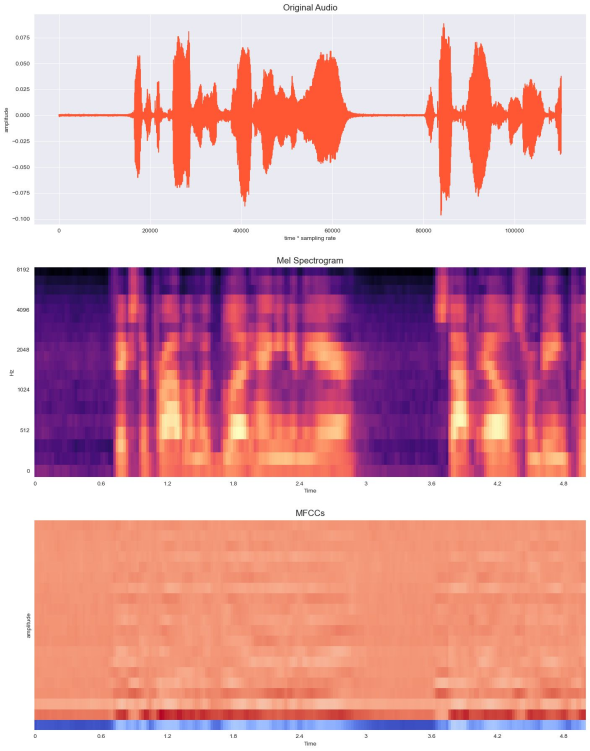
- The ‘Mel-frequency cepstrum’ is used in signal processing to convert the frequency distribution of a sound signal, mimicking the way humans hear.
- The coefficients that make up a MF cepstrum are commonly used for speech processing as they are scaled to the vocal and auditory systems of humans.
- Each coefficient (MFCC) represents how much of a speaker’s vocal output lies within a given frequency range as measured at various timeframes.

Data Processing and Analysis: Mel-Frequency Cepstral Coefficients

I used the ‘Librosa’ package for Python to extract the MFCCs for each five-second sample. This yielded 20 lists, each list representing a MFCC and containing 216 numeric time snapshots.

I took the mean of these 216 values to obtain a list of 20 numbers for each five-second sample. I then ran the resultant DataFrame representing all audio samples through three models, logistic regression, K-nearest neighbors and decision tree.

Results:
Mel-Frequency Cepstral Coefficients



Logistic regression: 4 incorrect out of 268 in test set.
ROC/AUC score: 0.999

K-nearest: 1 incorrect out of 268 in test set.
ROC/AUC score: 1.0

Decision tree: 6 incorrect out of 268 in test set.
ROC/AUC score: 0.983

Left: an original audio sample, the audio sample converted to the MF cepstrum, and the MFC coefficients for that sample

Data Processing and Analysis: Linear Predictive Coding Coefficients

‘Linear predictive coding’ is used in signal processing for speech transmission. The voice is modeled as a ‘source’, the original sound produced by the body, and a ‘filter’, the change in this sound produced by the resonance of the vocal tract.

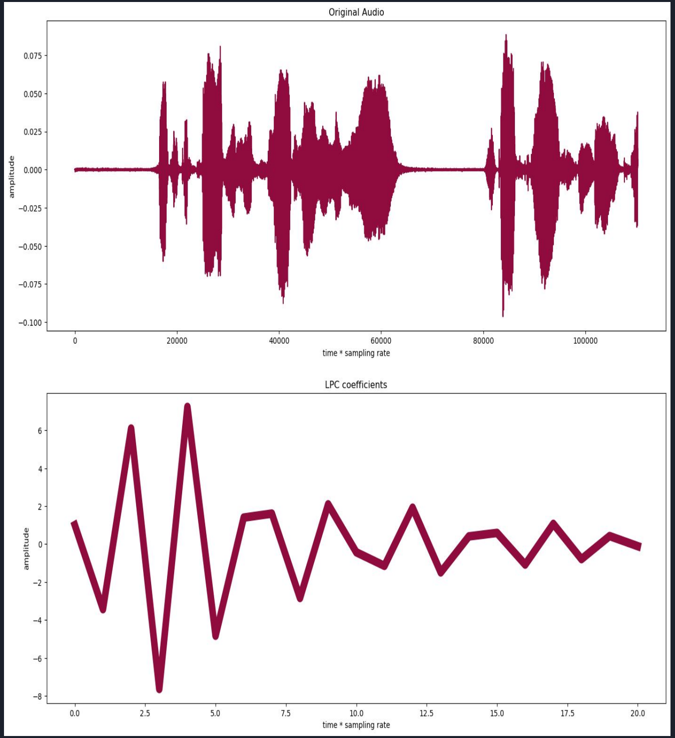
LPC separates the ‘source’, the ‘filter’ and random noise ‘error’ using a technique called autoregression - like multiple regression, but predicts future values of a variable using previous values of the same variable. Used for time series that contain repeating patterns.

Data Processing and Analysis: Linear Predictive Coding Coefficients

I used a custom Python function (Levinson-Durbin recursion, a linear algebra technique) to extract the LPC coefficients for each five-second sample.

This yielded 20 numeric LPCCs for each sample. I then ran the resultant DataFrame for all audio samples through three models, logistic regression, K-nearest neighbors, and decision tree.

Results:
Linear Predictive Coding Coefficients



Logistic regression: 6 incorrect out of 268 in test set.
ROC/AUC score: 0.998

K-nearest: 16 incorrect out of 268 in test set.
ROC/AUC score: 0.983

Decision tree: 32 incorrect out of 268 in test set.
ROC/AUC score: 0.919

LPC-KNN and LPC-DT had trouble telling real Trump from deepfake Trump.

Left: an original audio sample and the audio sample’s extracted LPC coefficients.

Potential Next Steps

Noisier audio samples - presidential audio is fairly clean, some applause and music. Some deepfake clips had background noise or music but it was minimal. Noise reduction techniques could be applied before coefficient extraction.

Using both the MFCCs and LPCCs may lead to better results in ‘trickier’ samples, as both methods perform some noise reduction themselves.

Combine this speaker recognition technique with a neural net for word recognition - could make a functioning multiple-speaker transcription program.

Thank you!

tracystrickel@gmail.com

linkedin.com/in/tracy-strickel

github.com/hypermoderndragon