ELEC5305 Project Proposal

Project Title:
Voice Deepfake Detection: An Audio Anti-Spoofing Approach
Using Spectral and Neural Features

Student Name: Nasser Alameri
SID: 510044280 **GitHub Username:** nmalameri
GitHub Project Link: https://github.com/nmalameri/elec5305-project-510044280.git

1 Project Overview

Voice deepfakes are synthetic speech audio generated by advanced text-to-speech (TTS) or voice conversion systems that can closely mimic real human voices, making it difficult for listeners to tell fake from genuine speech; which poses serious security risks if used maliciously (for fraud, misinformation, or bypassing speaker authentication systems), especially with the recent rise of AI-driven voice synthesis [1].

This project aims to develop a deepfake speech detection system using Python-based tools that are relevant to signal processing and machine learning, with its main function being to identify wether an input sample is genuinely human or a spoof (AI-generated or a recording) which will serve as defense against voice spoofing attacks. The approach will combine spectral features (MFCCs, spectrograms) with lightweight neural classifiers to identify subtle artifacts left by generative models and the evaluation will be based on the ASVspoof 2019 dataset [2]; which is a benchmark corpus for audio anti-spoofing.

2 Background and Motivation

As mentioned briefly above, the main motivation behind this project is the recent advances in AI-driven speech synthesis having achieved remarkable realism as synthesized voices can be nearly indistinguishable from real voices. While this has positive uses, it also brings serious cybersecurity threats as malicious actors could misuse deepfake audio to impersonate others or spread damaging false statements; which might facilitate fraud or public disinformation, highlighting the need for reliable detection solutions [1]. In response, initiatives such as the ASVspoof challenge have been launched to develop spoofing counter [2].

Prior studies found that features capturing high-frequency spectral details, such as linear-frequency cepstral coefficients (LFCC) as opposed to traditional mel-frequency cepstral coefficients (MFCC), and dynamic speech cues are especially useful for detecting synthetic audio [3]. On the other hand, classic solutions often used statistical models, such as gaussian mixture models (GMMs); for instance, the ASVspoof 2019 baseline (LFCC features + GMM) attained promising spoof detection performance without deep learning [2]. More recently, however, lightweight neural networks have been explored; such that a shallow CNN using LFCC features was a baseline in ASVspoof 2021 [4]. Such models remain efficient while still capturing key patterns of spoofed speech.

This project will delve into and explore whether lightweight Python-based methods can effectively detect deepfake speech without heavy computation and, if successful, could be easily deployed in low-resource or real-time settings.

3 Proposed Methodology

I will be mainly using Python for this project. A quick overview of what libraries I could utilize would be librosa with the help of NumPy and SciPy for low-level signal transformations in regards to audio processing; and scikit-learn to implement GMMs, and if needed, frameworks like PyTorch or TensorFlow/Keras to design a shallow CNN (deep learning) in regards to classification. The plan as of now is to use the ASVspoof public corpus as the basis for my experiments; specifically, the logical access portion of ASVspoof 2019, which provides many examples of bona fide speech and spoofed utterances generated with the latest TTS and voice conversion techniques [2]. This dataset offers a controlled setting with ground-truth labels for each audio file. If my laptop computational capacity allows, I may also evaluate on the newer ASVspoof 2021 Deepfake (DF) evaluation set, which includes recent deepfake attack examples [4]. All audio will be standardized (resampled) for consistency in feature extraction.

Cepstral features that characterize the speech spectrum and expose artifacts of synthesis will be extracted; with the primary features being:

- LFCC (Linear Frequency Cepstral Coefficients): derived from short-term spectra using a linearly spaced filterbank and DCT to produce cepstral coefficients [2]. My thinking at the moment is to retain approximately 20 cepstral coefficients (including the 0th energy) and include delta and delta-delta features for dynamics.
- MFCC (Mel-Frequency Cepstral Coefficients): computed similarly with a mel-scale filterbank. MFCCs will also be compared against LFCC to test whether capturing higher-frequency detail (as LFCC does) improves detection.

Moving to the classifiers, two lightweight classifiers will be implemented; which are:

- Gaussian Mixture Model (GMM): separate GMMs will be trained on bona fide and spoof feature vectors, and the log-likelihood ratio will be used between the models as the detection score [2].
- Shallow Convolutional Neural Network (CNN): a small CNN (inspired by an ASVspoof 2021 baseline) operating on the extracted features will be made [4]; keeping it minimal in size to ensure efficiency.

The GMM and CNN results will then be compared to determine whether the CNN offers any accuracy advantage over the generative GMM.

The overall workflow will involve:

1. Data Preparation: Organizing the audio data into training and testing sets; and performing any pre-processing needed (dependent).
2. Feature Extraction: `librosa` to compute LFCC and MFCC features for all audio samples, and store these features for training and evaluation.
3. Model Training: train GMM (separate models for real and fake) using the training set features; and if using the CNN, train the network on the training set, reserving a portion of the data for validation to tune parameters (normal procedure).
4. Evaluation: Apply the trained models to the test set to obtain detection scores and calculate performance metrics, primarily the Equal Error Rate (EER) [4], as is standard in spoof detection, and also overall classification accuracy for interpretability.

5. Analysis: Analyze the results, including which attack types are hardest to detect and compare performances amongst models. I will also examine example audio or feature plots to illustrate the differences between bona fide and spoofed speech to make clear.

4 Expected Outcomes

- Functional Detector: A working detection system (Python code) that can label an input audio sample as real or fake with high accuracy, aiming for low error rates comparable to known baselines (say around 15% EER).
- Feature Effectiveness Insights: Confirming which features are most effective. For instance, I expect LFCC features to outperform MFCC in capturing tell-tale artifacts of fake audio [3].
- Report and Presentation: A concise project report and presentation summarizing the methodology, key results, and conclusions, including suggestions for future work.

5 Timeline (Weeks 6-13)

Week	**Task**
6	Literature review; setting up Python environment; downloading ASVspoof 2019 dataset.
7	Data familiarization; implementing MFCC/LFCC feature extraction; verifing features on examples.
8	Training initial GMM on training set; tuning mixture components on development set; obtaining baseline GMM performance (EER, accuracy).
9	Developing shallow CNN and training on training set.
10	Evaluating models on the test set; computing detection metrics (EER, accuracy); comparing GMM and CNN results; performing error analysis.
11	Documenting and analyzing results; ensuring end-to-end code functionality on git hub.
12	Finalizing report and results; cleaning up code and README on GitHub; preparing presentation slides.
13	Submitting final report and code; presenting project (demo detector and discussing results).

References

[1] L. Cuccovillo, C. Papastergiopoulos, A. Vafeiadis, A. Yaroshchuk, P. Aichroth, K. Votis, and D. Tzovaras, "Open challenges in synthetic speech detection," in *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, Dec. 2022, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/WIFS55849.2022.9975433

[2] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling, "Asvspoof 2019: A large-scale

public database of synthesized, converted and replayed speech," 2020. [Online]. Available: https://arxiv.org/abs/1911.01601

[3] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *Interspeech 2015*, 2015, pp. 2087–2091.

[4] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, and H. Delgado, "Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection," 2021. [Online]. Available: https://arxiv.org/abs/2109.00537