Outline - Check-in #2

Title: Is This Really You? Real-Time Detection of Deepfake Voices

Who: Samuel Liu (szliu); Andrew Kim (akim199); Sami Nourji (snourji); Ghali Maata (gmaata)

Meeting date: 04/18/2025, 2pm

Github link: https://github.com/saminourji/DeepfakeAudioDetection

Introduction:
Our goal is to develop a deep model that outputs a continuous probability, in real time, about whether a piece of speech audio is a deepfake. On a phone call, for instance, people may need to determine whether the person they're speaking to is actually who they claim to be, or if they're imitating someone else through deepfake technology. This needs to happen in real-time; unlike the many existing deepfake classifiers that output a label at the end of a voice clip, we want our model to grow more confident in its classification over time.

This is thus a **binary classification task** with a continuous, time-evolving confidence level.

We'll be taking inspiration from pre-existing research papers when completing this project. Tehis paper published in 2023 achieves great results (99.3% average classification accuracy) on the real-time classification task we want to tackle. While the paper provides us with a detailed approach on how they created and trained their detection model, we quickly realized that the data they evaluated their model on was generated by the authors and of poor quality (when compared to deepfakes in other datasets). However, there are two key things we took away from this article: (1) when processing the audio, we should separate ambient noise from the speaker's voice, and (2) extracted features like Mel-Frequency Cepstral Coefficient (MFCC) and Spectral Bandwidth are particularly important for deepfake classification.

Another paper published in March 2025 introduces an interesting architecture for (non-real-time) deepfake audio classification. It leverages a Siamese CNN with a new "StacLoss" function and self-attention modules to achieve 98% accuracy on the more robust AVSspoof2019 deepfake dataset. Our own architecture will most likely be modeled after this one (with some adjustments, explained later), as we want to benchmark our model's performance against the most robust dataset possible.

Related work:
The two works cited and described above are important related work we'll rely on for the majority of our project. Additionally, this article published in the ACM Digital Library along with this GitHub repository containing a list of related research papers, aggregates most academic work recently done on this topic.

Public implementations:

While there are many implementations of deepfake audio detection (e.g. https://github.com/Srujan-rai/Deepfake_voice_detection), and real-time ***video*** deepfake detection (e.g. https://github.com/Zhreyu/Realtime-Deepfake-Detection), there is no public implementation of real-time **audio** detection at this time.

Data: AVSspoof2021 Dataset
Throughout our mini-literature review for this project, we explored the different datasets used by researchers in the audio deepfake detection community. We found that the ASVspoof 2019 and 2021 datasets were among the most used to evaluate models and compare results between architectures. The latest dataset, ASVspoof 2021, contains tens of thousands of short clips (2-5 seconds) of synthetic and voice-converted speech.

While we are conscious that there exist more recent and advanced Text-To-Speech (TTS) and Voice Conversion (VC) models, we are unable to use them as part of our project for two reasons. Firstly, we do not have the capacity or time to generate our own data. There exist open-source methods that could help us generate high-quality VC data from models like Diff-VC (Google) or MetaVC (Meta), but we decided that we did not want this to be the focus of our project, making this approach infeasible. Secondly, we do not have access to any *datasets* containing State-Of-The-Art VC models, with the best alternatives to ASVspoof 2021 being singing voice conversion datasets, which are outside of our project's scope, or datasets that are limited in quality or size.

ASVspoof 2021 is thus the best option for our project, and we plan to use its "LA" and "DF" tracks, which are designed to evaluate the detection of synthetic and VC under real-world codec (i.e., audio compression and decompression) conditions.

Methodology:
Training the model -
(1) We plan to train our model by feeding in regular deepfake audio samples, but incrementally in a time series, e.g. 1 second at a time.
(2) Using a Siamese CNN, we'll input fake and real audio samples, ensuring that our distance metric outputs low distance for real-real and fake-fake pairs, but high distance for real-fake pairs.
(3) We then plan to create a more lightweight version of a Siamese CNN to lower the latency of our model, allowing for potential real-time applications.

Hardest implementation detail - the Siamese CNN. A Siamese CNN consists of two identical subnetworks (e.g. they share the same architecture and weights) that each process different inputs and then output two distinct feature embeddings, which are then compared to each other using a similarity measure (something that determines how similar the two outputs are). We anticipate the similarity function being the biggest challenge, as we need to determine the optimal distance metric for our specific task of deepfake audio detection. This will involve experimenting with many distance metrics with different combinations of extracted features.

Novel changes - light-weight optimizations. Because our task is real-time deepfake detection, we need to design a model that balances robust and reliable outputs with quick, low-latency processing. Previous

real-time detection papers used boosting and random forests as their light-weight alternatives to CNNs, which could be interesting to consider in our process.

Fallback - If the lightweight Siamese architecture is too difficult to optimize for real-time use, we'll switch to a standard CNN classifier trained on fixed-length audio chunks (e.g., 1-second windows) and treat the task as a classification problem with majority voting across time.

Metrics:
Experiments - (1) we plan to compare how our model performs when trained on sequential audio fed in through a time series versus how it performs when trained on the entire audio clip at once (still testing it against the real-time detection task).

Measuring performance - Accuracy is a good metric for our project; generally, models assess accuracy through the binary classification task, but we need to adjust this target to fit our real-time detection goal. For instance, we will likely define a confidence threshold for assigning a binary label to the sample (e.g. 70%) above which a sample is classified as deepfaked.

However, we also need to consider how quickly the model reaches that confidence threshold. Is it after 3 seconds? 5 seconds? Does the prediction flip between time steps? During training, we may adjust weight updates based on how quickly the model reaches confident decisions, e.g., applying a larger gradient update when the model confidently classifies a sample early on.

When testing and evaluating, we'll measure not only overall accuracy, but also time-to-confidence, i.e., how many seconds of audio the model needs before it consistently surpasses the confidence threshold. This will allow us to both see how good our model is at detecting deepfakes, but also how fast it can detect them.

It is also important to note that, beyond accuracy, Equal Error Rate (EER) would also be a useful metric to evaluate our model's performance. EER is commonly used in speaker verification and spoofing detection tasks; it reflects the point where the false acceptance rate equals the false rejection rate. Including EER would allow us to compare our results more directly with ASVspoof baselines and related work in the field.

Base goal - get the PyTorch code working with siamese CNN

Target goal - lightweight CNN

Stretch goal - repeatedly feeding 1s clips

Ethics:

(1) What broader societal issues are relevant to your chosen problem space?
   (a) Scams and Financial Fraud: Real-time deepfake detection could be a powerful tool for preventing scams involving impersonation over phone calls. Criminals are increasingly using AI-generated voices to mimic loved ones, coworkers, or authority figures to manipulate victims into transferring money or revealing sensitive information. Our project could help build safeguards into call verification systems, especially in high-risk sectors like banking, tech support, and elder care.
   (b) Misinformation and Political Manipulation: In a more general sense, deepfake audio has the potential to be weaponized in disinformation campaigns by faking speeches, quotes, or interviews from political candidates, journalists, or activists. These fake audios can be especially damaging during high-stakes events like elections or social movements, where public trust is fragile. This is something we already saw in the 2024 US election and in other elections around the world.

(2) (our own question) What are some concerns with data that might be used in your chosen problem space?
   (a) Privacy and Consent in Voice Use: Many voice conversion and TTS systems are trained on scraped content from YouTube videos, podcasts, or public datasets, often without the speakers' knowledge or consent. This raises serious privacy concerns, especially when models are later used to impersonate those same voices. Ethical data sourcing is really important in this space, although it is a hard balance to strike when trying to develop detection tools: identifying unethical uses of voice data often requires access to similarly problematic data and technologies trained on that data.

Division of labor:

- Samuel Liu (szliu): focus on implementing the Siamese CNN architecture and training the base model
- Andrew Kim (akim3): works with Samuel on the Siamese model, with additional focus on real-time audio preprocessing and time-series input formatting.
- Sami Nourji (snourji): manages the project timeline, selects and processes the dataset, and works on tuning the model to make it more lightweight.
- Ghali Maata (gmaata): works on model evaluation and performance metrics for accuracy, latency, and time-to-confidence.

All members contribute to writing the final report.