DeepFake Audio Forensics Testbed

Development of a testbed for detecting counterfeit multimedia content generated by DeepFake methods

Master's project by Kenneth Tasie, Divjot Singh, Mateusz Koniarek, Tanvir Ahmed

---

Project Overview

With the rise of AI-generated speech, distinguishing authentic human voice from cloned, anonymized, or fully synthetic DeepFakes is critically important. This repository provides a modular, end-to-end testbed to:

1. **Generate Mel-spectrograms** and raw spectral arrays
2. **Extract audio features** (spectral centroid, PSD, spectral flatness, formants, wavelets, pitch via YIN, RMS/power, high-frequency artifacts)
3. **Quantify differences** via numerical metrics (Euclidean distance, MSE, cosine similarity)
4. **Compare images** using SSIM and MSE on spectrogram PNGs, with annotated composites
5. **Perform forensic analysis** (Pitch MAE, spectral flatness/power PSD deviations)
6. **Produce visual reports** (bar charts, KDE plots) and CSV summaries

The testbed evaluates three DeepFake categories—**cloned**, **anonymized**, and **synthetic**—against genuine human recordings, helping researchers benchmark detection methods and understand artifact characteristics.

Repo Structure

```
├─ .gitignore
├─ README.md                 # This overview
├─ mapping_template.csv      # Example file:
id,human,cloned,anonymized,synthetic
├─ spectrograms.py           # Generate spectrogram PNGs & .npy arrays
├─ forensic_analysis.py      # Core feature extraction and CSV of feature
deltas
├─ advanced_forensic_analysis.py  # Extended features: formants, wavelets,
YIN pitch
├─ numerical_analysis.py     # Compute Euclidean/MSE/cosine on spectrogram
arrays
├─ statistical_analysis.py   # Compute mean/variance/std-dev statistics
├─ comparison_detailed.py    # SSIM/MSE image comparisons with annotated
output
├─ buzzy.py                  # High-frequency “buzzy” artifact & loudness
comparisons
├─ visualize_metrics.py      # Bar-chart & KDE visualization of all metrics
├─ debug.py                  # Sanity-checks against mapping_template.csv
└─ requirements.txt          # Python dependencies
```

Quickstart

1. **Clone the repo**

```
git clone https://github.com/kentex99/deepfake-audio-forensics.git
cd deepfake-audio-forensics
```

1. **Install dependencies**

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

1. **Prepare data**

2. Organize your WAV files into folders (e.g. `data/human/`, `data/cloned/`, `data/anonymized/`, `data/synthetic/`)

3. Edit `mapping_template.csv` with headers:

   ```
   id,human_path,cloned_path,anonymized_path,synthetic_path
   ```

4. **Generate spectrograms**

```
python spectrograms.py --mapping mapping_template.csv --output output/
spectrograms
```

1. **Run forensic feature analysis**

```
python forensic_analysis.py --mapping mapping_template.csv --output output/
feature_deltas.csv
python advanced_forensic_analysis.py --mapping mapping_template.csv --output
output/feature_deltas_extended.csv
```

1. **Compute numerical & statistical metrics**

```
python numerical_analysis.py --arrays output/spectrograms --output output/
numerical_metrics.csv
```

```
python statistical_analysis.py --arrays output/spectrograms --output output/
statistical_metrics.csv
```

1. **Image-based comparisons**

```
python comparison_detailed.py --input output/spectrograms --output output/
image_comparisons
```

1. **Detect high-frequency artifacts**

```
python buzzy.py --mapping mapping_template.csv --output output/
buzzy_metrics.csv
```

1. **Visualize all metrics**

```
python visualize_metrics.py --metrics-dir output --plots output/plots
```

Future Work

- **Questionnaire Deployment**: Integrate human perception feedback via Google Forms
- **Deep Learning Integration**: Train CNNs on spectrograms using documented feature insights
- **Enhanced Forensics**: Add MFCC-based DTW distances, PESQ/STOI perceptual metrics
- **Dashboard**: Build interactive Streamlit/Dash app to explore results

Contact

Kenneth Tasie (Team Member)\ ✉️Tasyken121@gmail.com\ GitHub: kentex99