AI Audio Enhancement Research

Project Status & Future Roadmap

A comprehensive overview of our progress in digital audio processing and AI model development

November 2025

What is this project about

Can we use AI to upscale lossy audio files to near-lossless audio quality?

Benefits:

- Reduce streaming bandwidth if native to streaming devices
- Optimize disk space for donwloaded audio files
- Allow lossless streaming on Bluetooth devices
- Highly distributed IoT networks for audio processing

Tools

AI

- Model - 1D U-Net archtecture
  - PyTorch for ML
  - Librosa for processing
- Hosted through Kaggle

Back-End Streaming

- API:
  - FastAPI framework
  - Librosa for processing
  - Pydantic for DTO/validation
  - Uvicorn for hosting
- DB
  - MongoDb for file metadata
  - File Store for audio Files

FrontEnd

- Flutter app
  - Network packaging for API
  - Web audio playback (Audio Element)

What Went Well

AI Model Development Progress

- Advanced the AI output that has a level of **interpolation**

Streaming Platform

- Created simple streaming platform that sends File meta data and streams audio files to a client

Challenges Encountered

Dataset Limitations

- **Lack of diverse datasets** for comprehensive training
- Affecting interpolation due to lack of diverse files

Streaming

- Easily storing and streaming files was a challenge

Key Features:

- Mock Streaming service
- Audio playback from streaming service
- AI upscaling output with small level of interpolation

Future Plans & Roadmap

AI Model Enhancement

- Create/find data sets
- Optimize for native implementation on less powerful devices

Streaming Simulation

- Integration with AI that sends lossy files to "enhance"
- Integrate evaluation metrics from previous iterations

Key Learnings & Technical Insights

Digital Audio Processing Mastery

- Comprehensive understanding of **audio compression algorithms**
- In-depth knowledge of **perceptual audio coding**

Streaming output

- Easy streaming interface that will easily display differences between files

Sprint 1 overview

- Week 1: Defined **terminology** - began **audio evaluation**
- Week 2: **More terminology** - Continued evaluation - **audio I/O** - Began **AI model development**
- Week 3: ***Even more*** **terminology** - Meta-data handling. Differentiating data sets
- Week 4: **Surprise! more terminology** - Graph Construction/Modeling - **AI training**

Sprint 2 overview

- Week 5: **Data set** and **AI refactoring**
- Week 6: Began creating **DB** and **streaming API**.
- Week 7: Finished **DB and backend**. Began creating **frontend** flutter interface
- Week 8: Fulling integrating **Streaming Playback** to front end with **Meta-Data**

Metrics

- Total individual Lines of Code (LoC): ~4085
- Number of individual features completed: **5**
  - Total Features: 8
- Number of individual requirements completed: **11**
  - Total requirements 24
- Individual burndown rate (%): **45%**

Next Steps & Immediate Actions

Sprint ???

1. **Acquire MORE datasets** from diverse sources
2. Move AI model hosting to **Google Colab** for **GPU** usage
3. **Integrate API to Model for output to streaming**
4. **Integrate evaluation metrics** to the frontend
5. Create **Native instances** of AI Model

Technical Implementation Overview

Backend Architecture

- Audio file processing - chunking for streaming
- Python-based processing pipeline
- Mongo-DB integration with file-store
- API for managing files

Analysis Components

- Flutter/dart front end implementation
- Audio preprocessing and normalization both on front-end and back-end
- User visualization/listening platform