THE UNIVERSITY OF SYDNEY

ELEC5305

Project Progress Update

Instrument Family Classification in Music Recordings Using Audio Signal Processing and Machine Learning

Student:
Dawod Ghifari 520140154

Github Username:
dawodghifari

Teacher:
Craig Jin

October 12, 2025

Contents

1 Introduction

This project focuses on the automatic classification of musical instrument families from audio recordings using signal processing and machine learning techniques. The five target families are *strings*, *percussion*, *winds*, *keyboards*, and *voice*.

The primary objective is to prepare a clean, balanced, and well-labeled dataset suitable for supervised learning. This report summarizes the progress made in data acquisition, conversion, and segmentation, forming the foundation for the subsequent extraction of features and model training.

2 Data Acquisition

2.1 Audio Sources

Raw data was collected from multiple online repositories that provide royalty-free or community-licensed audio samples appropriate for academic use. Table 1 lists the sources and the type of material collected.

Source	**Type**	**Example Content**
TV & Radio Voices	Spoken voice	Male and female dialogue
Looperman	Vocal acapella	Singing and vocal phrases
Sample Focus	Instrumental	Strings, percussion, keyboards, winds
Pixabay	Mixed audio	Humming and instrument clips

Table 1: Summary of audio sources used for dataset collection.

3 Raw Data Characteristics

The original downloads were a mix of `.mp3` and `.wav` formats, with sampling rates up to 44.1 kHz and inconsistent bitrates. Audio durations varied widely, ranging from short 2–3 second clips to full multi-minute tracks.

This variability made direct use impractical for training machine learning models. Standardization was therefore required to ensure uniform audio properties across all recordings.

4 Data Conversion and Organization

4.1 Conversion Process

Data conversion was performed using a MATLAB Live Script named `file_rename_converts_00.mlx`. The script automated the following steps:

1. Conversion of all files to the `.wav` format.
2. Resampling each recording to mono, 16 kHz.
3. Renaming files using the convention `<family>_<instrument>_<index>.wav`.

4. Classifying and moving files into the appropriate family directory using keyword-based inference.

This ensured consistent sample rate, file format, and naming conventions across the dataset.

4.2 Project Directory Structure

After conversion and cleanup, the project directory was standardized as follows:

```
elec5305-project-520140154/
 Raw Data/                 % unprocessed original audio
 Data/                     % standardized 16kHz WAV files
    strings/
    percussion/
    winds/
    keyboards/
    voice/

 Segmented/                % 3-second processed clips
 Manifests/                % metadata CSV + MAT files
 Models/                   % model training outputs
 Results/                  % evaluation metrics and plots
 Scripts/                  % MATLAB codebase
 Reports/                  % LaTeX documents and notes
```

This modular structure enables reproducibility and clear separation between raw, processed, and derived data.

5 Data Segmentation and Preprocessing

5.1 Objectives

The goal of segmentation is to create a consistent, balanced dataset of short, fixed-length audio clips suitable for supervised learning and CNN-based architectures.

5.2 Segmentation Method

Segmentation was implemented in MATLAB Live Script `data_segmentation_01.mlx`. The process performs:

1. Resampling all audio to 16 kHz if required.
2. Segmenting into non-overlapping 3-second windows (`clipSec = 3`, `hopSec = 3`).
3. Zero-padding shorter clips to maintain uniform duration.
4. Removing segments below `-40 dBFS` RMS threshold.
5. Applying dynamic capping:

- Family scarcity weight ($w_{fam} = 0.6$)
- Instrument scarcity weight ($w_{instr} = 0.4$)

6. Writing each clip to:

```
Segmented/<family>/<instrument>/
```

Each segmentation run produces:

- Segmented audio clips.
- A manifest in CSV and MAT format containing file paths, labels, duration, RMS levels, and applied caps.

6 Results and Achievements

6.1 Segmentation Output Summary

The final dataset after segmentation consists of approximately 870–900 clips distributed across the five instrument families. Each clip is 3.000 s in duration, mono, 16 kHz, and free of silence. Table 2 summarizes the distribution.

Family	**Clips (approx.)**	**Mean RMS (dB)**
Strings	142	−22.4
Percussion	138	−18.7
Winds	187	−19.9
Keyboards	214	−27.4
Voice	170	−21.3

Table 2: Balanced clip distribution across instrument families.

6.2 Achievements

- All audio standardized to **16 kHz mono, 3 s duration**.
- Silence removal successfully reduced low-energy noise clips to below 4%.
- Dynamic capping provided improved balance across families and instruments.
- RMS and duration verification confirmed data consistency.
- A manifest-based structure ensures full reproducibility and traceability.

7 Next Steps

- Split the dataset into training, validation, and testing partitions (source-file grouped to prevent leakage).
- Extract spectral features (MFCC, Mel-spectrograms).
- Train baseline models (SVM, Random Forest), followed by deep learning models (CNN, VGGish).
- Store trained models in the `/Models` directory and evaluation outputs in `/Results`.

8 GitHub Page

All project files, including MATLAB Live Scripts, converted datasets, segmentation manifests, and LaTeX documentation, have been committed and pushed to a public GitHub repository for transparency, version control, and feedback.

The project is hosted under the GitHub username: `dawodghifari`

The repository can be accessed at:

`https://github.com/dawodghifari/elec5305-project-520140154`

It contains:

- Full source code for all data processing and segmentation scripts (`Scripts/` directory).
- Generated and processed datasets, manifests, and metadata files.
- The LaTeX report and supporting figures.

This repository will continue to host updates throughout the project, including feature extraction scripts, trained models, and evaluation results. Supervisors and peers are encouraged to review the repository and provide feedback through GitHub issues or pull requests.

9 Conclusion

All raw audio data has been successfully standardized, converted, organized, and segmented. The resulting dataset is clean, balanced, and labeled by family and instrument. The data preparation phase is complete, providing a robust foundation for the next phase of feature extraction and model development.