Troy Sabia
C964

Task D – Quick Start Guide

System Requirements

Windows 10 or 11
Requires FFMPEG installation
Python 3.13.3
Python packages for classification only:
torch, torchaudio, os, shututil, subprocess, tempfile

additional Python packages required for training:
argparse, json, csv, math, numpy, sklearn

Installation

Unzip (or clone if using Git) files to working directory
Place MP3 files in /input_mp3
Run “python process_directory.py”
Files will be automatically sorted into the /sorted_mp3 directory

Note: some sample files have been placed in the directory for testing purposes

Training

1. Collect Music Files

Music files are collected into three directories in the project directory each with a metal and non_metal subdirectory:

/dataset/metal */dataset/non_metal*	These files are used to train the model
/valset/metal */valset/non_metal*	These are the files against which dataset is validated
/testset/metal	These are files used to evaluate the final model

/testset/non_metal

The suggested ratio is an even split between metal and non_metal in each subdirectory with an overall balance of approximately 80% dataset, 10% validation set, and 10% test set.

Current model used 82%-11%-6% (3933/569/290 = 4792).

2. Extract Features

With directories populated in the project directory, run “python save_mfcc_cache.py”. This will process all mp3s by converting them to 22.05 kHz mono WAV files and extracting three 15 second clips. These clips are then converted to MFCCs and saved in the appropriate /mfcc_cache subdirectory.

3. Modify Model

Current optimized model is located in metal_classifier_optimized.py. Make desired changes to it, e.g. alter number of parameters or change activation function. Note that this model must stay the same for all following steps as further steps will require the same structure and number of parameters to work consistently.

4. Train Model

Once all features have been extracted, run “python train_with_metrics.py” with the following parameters:

parameter	default	description
--run_name <name>	“baseline”	specify the directory it is saved in
--epochs <number>	40	the number of epochs to train
--step_size <number>	5	# epochs until learning rate is reduced
--gamma <float>	0.6	amount learning rate reduced by
--lr <float>	1e-4	learning rate

Consult train_with_metrics.py for other parameters. Model and metrics are saved in /runs/<run_name> folder.

5. Evaluate Metrics

Run “eval_test.py --ckpt /run/something/best_model.pt” (substitute location of preferred model).

This will give Accuracy, F1, Precision, Recall, and Confusion Matrix information for model evaluation.

Repeat steps 3 through 5, adding/removing parameters, changing training functionality, etc until desired metrics are achieved.