Sampling Vocals Using Deep Learning Methods to Isolate Human Voices

ENSC 429 D100

Group 3

Darren Fok – 301461164 – dyf@sfu.ca

Chris Sim – 301448214 – wms8@sfu.ca

Tony Chen – 301458138 – dec8@sfu.ca

Nick Lagasse – 301451354 – nla78@sfu.ca

Introduction:

In certain genres of music, sampling is the process of re-using a portion of audio from an external source to use in music production. A vocal snippet is one of the potential candidates for such sampling. Often however, if sampling from another pre-existing song, these vocals will be accompanied by background noise and instruments, which can conflict with the musical integrity of the song being produced. In this project, we will develop a tool that can isolate vocals from any background noise or instrumentals. Using deep learning and training a model with digital signal processing methods such as filtering, we can produce a clean isolated vocal sound file from an input audio file.

Vocal Isolation is the process of removing unwanted audio and obtaining a clean audio snippet purely consisting of a human voice and is a classic problem in digital signal processing (DSP) with common areas of application including music production, hearing aids, and speech enhancement. Traditional techniques include spectral subtraction, and time-frequency masking, both of which fall short due to relying on written rules and assumptions about signal structures.

However, recent developments in deep learning methods have enabled even more accurate vocal separations. A strong example of this is Spleeter, a software developed by Deezer Research, which utilizes convolutional neural networks (CNNs) trained on spectrograms to perform vocal separation [1]. Another example is Demucs, which improves on existing developments by operating in the time domain and utilizing hybrid CNNs to directly extract vocals from the raw waveform. This method results in an improved signal-to-distortion ratio and higher quality output [2].

Project Description:

In our design, we will attempt to isolate vocals from any unwanted background and foreground audio. To verify our results, we will attempt our process on songs that have official instrumentals and acapella tracks and compare the results to the official files. If the result is not satisfactory, we will tweak our trained model and our algorithm to continually iterate our results. Furthermore, regarding the model, we will aim for a 70% accuracy when identifying whether an audio file is an isolated vocal, background noise or instrumental, or both simultaneously. When comparing waveforms of our output and the official isolated vocals, we will attempt a difference in waveforms as close to 0 as possible.

Plan:

We will begin by sourcing an audio dataset from online sites such as Kaggle and then split it into a training (70%) and test (30%) set. Each clip from the dataset will be annotated as

either containing vocals, background noise or both. We will then convert the audio clips into a spectrogram by utilizing a Short-Time Fourier Transform (STFT), via a script that will process all files within a folder.

A CNN will then be designed and implemented to be trained on features of spectrograms and then evaluated on a separated and unique test set. With a script that utilizes the trained CNN, it will then output a spectrogram, which will have a reverse STFT performed on it to obtain an audio output. A UI will be developed that utilizes the trained model, with an optional goal of implementing real-time vocal isolation if time permits.

The work is being divided based on individual strengths and prior experience. Darren will handle data annotation and play a keep role in training and implementing the CNN model due to his prior experience in CMPT 419 (Machine Learning), with assistance from Chris. Nick and Chris will focus on preprocessing, while Tony will be focusing on postprocessing, with Chris and Tony's experience from CMPT361 (Visual Computing) with neural networks and deep learning playing a key role. Lastly, the UI will be handled by Tony and Nick.

Preliminary Work:

So far, we have gathered various stem datasets (which include vocals and instrumentals) from Kaggle, and we have decided we will be using PyTorch in Python due to previous experience with those tools.

Feasibility Analysis:

Given the pure software nature of this project and the previous experience of our groupmates, we should be able to at least finish a rudimentary demo that can isolate vocals from random background noise. Additionally, this technology has been developed in the past, and we may be able to draw from open-source code online. The most variable aspect would be the quality of the vocal isolation, as it would depend heavily on our model and the actual implementation of filtering and frequency separation.

References:

[1] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a Fast and Efficient Music Source Separation Tool with Pre-Trained Models." Accessed: Jun. 20, 2025. [Online]. Available: https://research.deezer.com/publication/2020/06/24/releasing-spleeter.html

[2] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music Source Separation in the Waveform Domain," Apr. 29, 2021, *arXiv*: arXiv:1911.13254. doi: 10.48550/arXiv.1911.13254.