CMPE491 – Senior Project I
Spring 2021

Analysis Report

Voiceolation

Emir Yılmaz

Kemalcan Güner

Yunus Emre Günen

1. Introduction

Voiceolation is a music source separator that extracts vocals from songs. Either companies may need it for music information retrieval (MIR) and karaoke, or individuals may use it for their personal usage like mixing or editing a song.

Music source separators have become very common after the rising popularity of machine learning and neural networks. In the past, there were limited amounts of labeled datasets for training as they are difficult to label and require professional annotators. While the world is trying to overcome these hardships and gains great momentum towards the perfect source separator methods, they still need to improve, also datasets need to expand to be comprehensive that include various genres, languages, etc. We are joining the bandwagon too, as Voiceolation.

2. Proposed System

2.1 Overview

Like every neural network system, Voiceolation needs a dataset to train on. One of the biggest problems in source separation on the music, is lack of datasets. As it is not a final decision we will be planning to use MUSDB18[1,2]. There are other datasets like MedleyDB and RWC. All of them music focused with various instrumental stems and some of them have melody and pitch annotations. If investigated further, more datasets can be found, but as we researched they shifted to speech recognition rather than source separation.

To train these datasets, we decided to use the well-known U-Net[3], which is a convolutional neural network that originally developed for image segmentation. U-Net has a lot of variations and similar architectures for music source separation[4,5,6]. Even if we change the model, it will be akin to U-Net. Also, we are planning to use PSNR or VQM which will be one of the novel parts of this project.

With the usage of U-Net, Voiceolation is confirmed to work on spectrogram (frequency-domain) instead of waveform (time-domain), since U-Net architecture works on images, we will work on frequency spectrogram and Fourier transform - probably Short Time Fourier Transform (STFT) - will be used for transition to spectrogram. Both of them have their advantages and disadvantages and still explored topics in scope of source separation.

From a view of the end user, Voiceolation is a system that separates a music file into a vocal and an instrumental parts. We are planning to have a website for use by the end-user. The website allows users to upload a local sound file to the server.

Users can upload, listen and download audio files. The processes are like this; the user opens the website, the website displays a “Terms of Service” and asks users to accept it. If the user does not accept and agree, the website does not allow the user to continue any processes. After accepting “Terms of Service”, the user can upload a local sound file into the server. (the keyword “local” is important because the website will not reach any kind of URL of any other website to handle legal issues, especially copyright. The website only works with the files that are located on users’ devices.) After uploading the sound file, the user may listen to the sound file if they want.

The uploaded sound file is sent to the server, Voiceolation will do the aforementioned vocal extracting processes. After the separation of the user's music, the end user gets two playable sound files: one of them is a vocal, another one is the rest of the music which is instrumental. The last operation that users can do is downloading the new separated files, users may download both of them. After these processes, users may repeat them with different sound files.

2.2 Functional Requirements

2.2.1 User Requirements

- The user shall see the user interface clearly.
- The user shall upload a song file from local memory easily.
- The user shall get two result files one is for vocal, another one is for instrumental.
- The user shall download the files whichever they want.

2.2.2 Website Requirements

- The website shall support to upload and download audio files.
- The website shall extract the vocals from the given file.
- The website shall offer a functional music player to users.

2.3 Nonfunctional Requirements

2.3.1 Performance

- According to the uploading data (size of the file) and upload speed of the user, response time may be different in a variety of ranges.

2.3.2 Usability

- The site shall be easy to use by who needs vocal separation to whatever and has basic computer skills.

2.3.3 Scalability

- The user shall upload one song at a time
- The interface will be easy to use with understandable and very simple design.

2.3.4 Supportability

- The site shall be available on any OS which has a browser.

2.3.5 Security

- We will not ask, hold or store any kind of user information.
- We do not save uploaded sound files because it can also contain private information. The file will be processed, and then will be deleted from the server.
- The processed version will be available for only limited time which is enough for listening and downloading.
- We will not share any kind of user data to third parties.

2.4 Pseudo Requirements

- Python will be used for neural network and vocal separation parts.
- GitHub will be used for collective work and deploying.
- PyTorch will be used to create neural networks.
- MUSDB18 will be used to train and test.
- Since the system has not been installed yet, new requirements could be added.

2.5 System Models

2.5.1 Scenarios

Scenario 1: Upload Sound File
Actors : User and Server
Entry Conditions: User clicks the button “select local file”
Exit Conditions: If the file is uploaded to the server successfully
Flow Events:

- User clicks the “select local file” button
- Web browser permission for reaching the local files which are located in the users computer
- User selects the audio file
- Server takes the file successfully and process it

Scenario 2: Listen Separated Sounds
Actors : User
Entry Conditions: User clicks the button “listen”
Exit Conditions: Whenever user wants to stop or end of the sound file
Flow Events:

- Server displays two separated sound files to user
- Server displays an audio player for listening both tracks //User shall listen either vocal sound file or instrumental sound file

Scenario 3: Download Sound File
Actors: User
Entry Conditions: User clicks to “Download” button
Exit Conditions: Download finishes successfully or user cancels the download process
Flow of Events:

- User navigates and clicks to “Download” to either vocal or instrumental stem
- Depending on the browser, it will ask for download location or make the user give permission to download process
- Server will delete the file from our servers in short time

2.5.2 Use Case Model

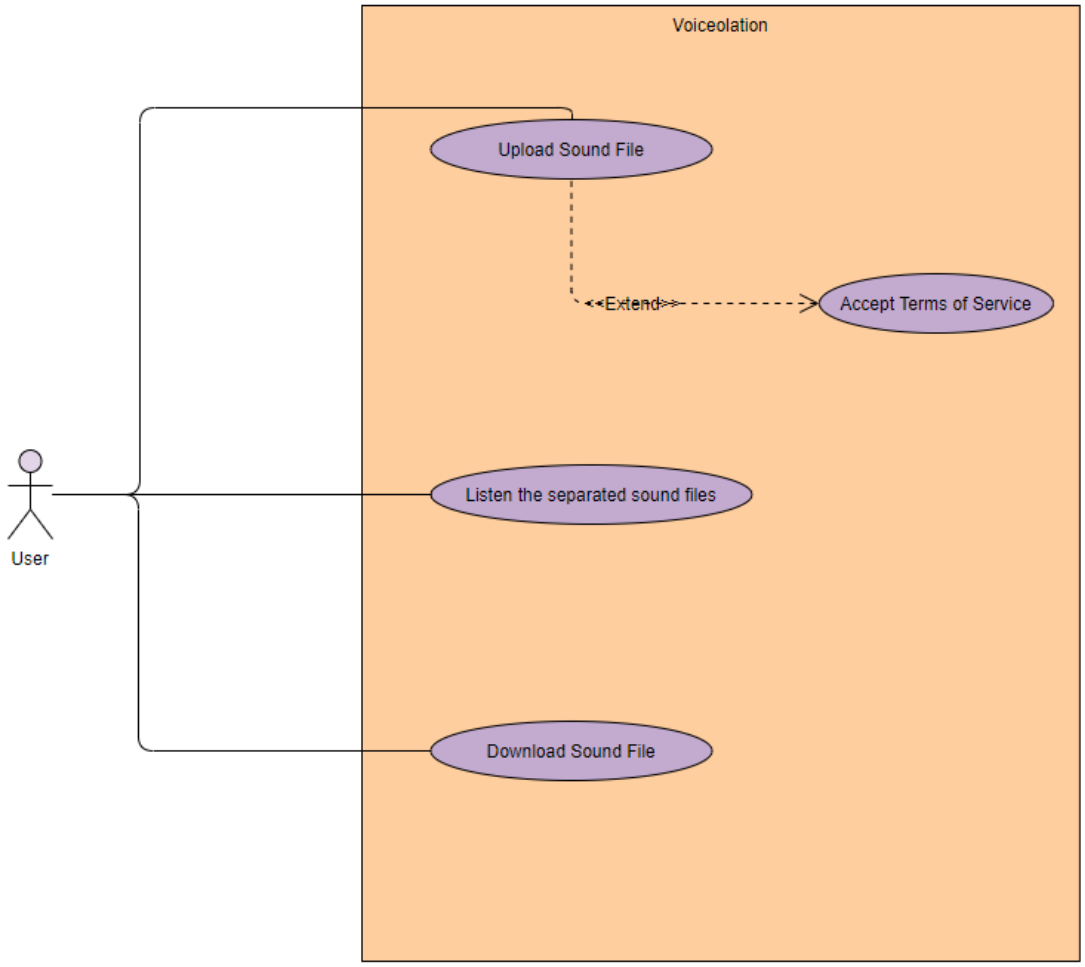

Figure 1: Use Case Diagram of the Website

2.5.3 User Interface

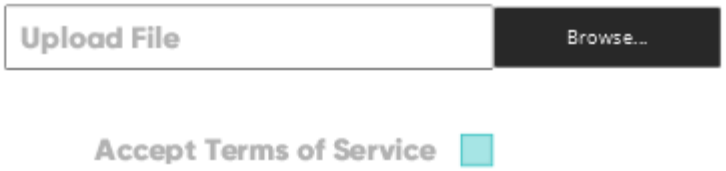

Figure 2: Audio file upload screen with simple representation

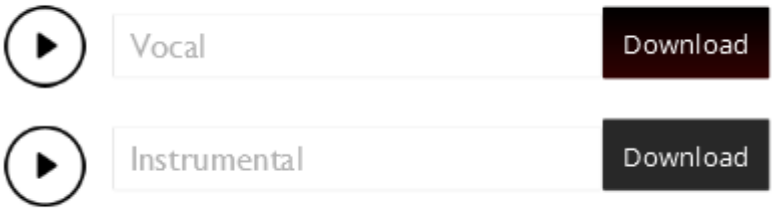

Figure 3: Listening and downloading the processed audio files screen with simple representation

3. References

1. Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., & Bittner, R. (2017, December). MUSDB18, a corpus for audio source separation. doi:10.5281/zenodo.1117372
2. Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., & Bittner, R. (2019, August). MUSDB18-HQ - an uncompressed version of musdb18. doi:10.5281/zenodo.3338373
3. Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. doi.org/10.1007/978-3-319-24574-4_28
4. Choi, W., Kim, M., Chung, J., & Jung, S. (2020). LaSAFT: Latent Source Attentive Frequency Transformation for Conditioned Source Separation. arXiv preprint arXiv:2010.11631.
5. Jansson, A., Humphrey, E., Montecchio, N., Bittner, R., Kumar, A. and Weyde, T. (2017). Singing voice separation with deep U-Net convolutional networks. Paper presented at the 18th International Society for Music Information Retrieval Conference, 23-27 Oct 2017, Suzhou, China.
6. Meseguer-Brocal, G., & Peeters, G. (2019). Conditioned-U-Net: Introducing a Control Mechanism in the U-Net for Multiple Source Separations. ArXiv, abs/1907.01277.