TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Major Project Report

On

Data-Driven Approach in Isolating Vocals and Instruments from Music

Submitted By:

Anish Dahal (Exam Roll No.: 25354)

Prajwol Pakka (Exam Roll No.: 25367)

Sujal Subedi (Exam Roll No.: 25388)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

March, 2022

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Major Project Report

On

Data-Driven Approach in Isolating Vocals and Instruments from Music

Submitted By:

Anish Dahal (Exam Roll No.: 25354)

Prajwol Pakka (Exam Roll No.: 25367)

Sujal Subedi (Exam Roll No.: 25388)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor's Degree in Electronics and Communication Engineering.

Under the Supervision of

Er. Dinesh Baniya Kshatri

March, 2022

DECLARATION

We hereby declare that the report of the project entitled **"Data-Driven Approach in Isolating Vocals and Instruments from Music"** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics and Communication Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Anish Dahal (THA074BEX004) _____

Prajwol Pakka (THA074BEX022) _____

Sujal Subedi (THA074BEX043) _____

Date: March, 2022

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a major project work entitled "**Data-Driven Approach in Isolating Vocals and Instruments from Music**" submitted by **Anish Dahal, Prajwol Pakka,** and **Sujal Subedi** in partial fulfillment for the award of Bachelor's Degree in Electronics and Communication Engineering. The project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled, and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of the Bachelor's degree in Electronics and Communication Engineering.

_____

Project Supervisor
Er. Dinesh Baniya Kshatri
Department of Electronics and Computer Engineering, Thapathali Campus

_____

External Examiner
Dr. Pradip Paudyal
Nepal Telecommunications Authority, Kantipath, Kathmandu

_____

Project Coordinator
Er. Umesh Kanta Ghimire
Department of Electronics and Computer Engineering, Thapathali Campus

_____

Head of Department
Er. Kiran Chandra Dahal
Department of Electronics and Computer Engineering, Thapathali Campus

March, 2022

COPYRIGHT

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards the Institute of Engineering, Tribhuvan University for the inclusion of the major project in the course of Bachelors in Electronics and Communication Engineering. We are also thankful to our Supervisor Er. Dinesh Baniya Kshatri and the Department of Electronics and Computer Engineering, Thapathali Campus for providing us with the resources and support which is needed for this project.

Anish Dahal (THA074BEX004)

Prajwol Pakka (THA074BEX022)

Sujal Subedi (THA074BEX043)

ABSTRACT

This project involves the separation of vocals, instrumentals, drums, and bass stems from songs. For the separation, two approaches are used. Initially, the 2DFT approach, which only separates vocal and, instrumental is used, which leverages the fact that instrumentals have some amount of periodic repetition, while the vocals are relatively aperiodic. In the machine learning approach, convolution-based architecture, “U-Net” is used. This provides instrumental, vocal, drum, or bass, whichever is to be extracted. Finally, the instrumental and vocal results from both approaches are compared.

Keywords: 2DFT, Convolution, U-Net

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2DFT	2-Dimensional Fourier Transform
ABI	Artificial Biological Intelligence
ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DLM	Deep Learning Method
DNN	Deep Neural Network
FT	Fourier Transform
GPU	Graphics Processing Unit
GUI	Graphical User Interface
IFT	Inverse Fourier Transform
LSTM	Long Short-Term Memory
MSS	Musical Source Separation
NMF	Non-Negative Matrix Factorization
SAR	Signal to Artifact Ratio
SDR	Signal to Distortion Ratio
SNFM	Sparse Non-Negative Matrix Factorization
SNR	Signal to Noise Ratio
STFT	Short Time Fourier Transformation
SVM	Support Vector Machine

1. INTRODUCTION

Musical Source Separation (MSS) is the process of isolating individual sounds in an auditory mixture of multiple sounds. It is the opposite of the mixing process used by audio engineers and musicians to create songs. Music Separation is a sub-domain of Source Separation, which is the process of isolating at least the musical background (instrumentals) or the vocals of the singer from a musical mixture (song).

1.1 Motivation

While listening to the instrumentals is pleasing, the creation of those is not. The team came across this reality when one of the team members tried to make his own instrumental by using software known as Audacity but failed to do so. It was a simple method that used the concept of phase inversion. The method was quick, but the output was unsatisfactory. In addition, the team members used an online separating program. Although it generated an output, there were still detectable ranges of vocals present in instrumental music. Hence, the output found was still unsatisfactory. Therefore, a question arose – "Why not create a better approach for this task?"

1.2 Problem Definition

Songs are primarily a mixture of vocals and instrumentals. However, artists rarely compose them separately. But, they are useful for different applications. This requires the use of third-party software which might be proprietary or open-source. Such software can be expensive, cumbersome to use, and their performance might not be up to par, to be used at the professional level. Thus, this project helps to facilitate singers, musical engineers, and the general population to separate vocals and instrumentals by providing an open-source program that is easy to use and delivers satisfactory performance.

1.3 Project Objectives

The specific objectives of this project are listed below**:**

- To separate vocals and instrumentals from songs
- To isolate drum and bass stems from instrumentals

1.4 Project Applications

There are many applications of separating vocal and instrumental from a song. Some of the prominent real-life applications of the project are listed below.

- Karaoke – It is a form of entertainment whereby people take turns singing songs into a microphone over prerecorded background tracks. The extracted instrumentals from the project can be used as background tracks.

- Audio Mixing – The drum, bass, vocal, and instrumental stems extracted from a song can be used to produce new audio mixtures. This is very useful for audio engineers and DJs.

- Lyrics Extraction – The drawing out of lyrics from a song is a challenging task due to the presence of music. The extracted vocals from the project can be used as input for a lyrics withdrawal system.

- Singer Identification – This project can be enhanced to identify singers, the number of singers, and their gender. In addition, identifying singers can help with copyright issues for someone claiming others' work as their own.

- Noise Reduction – The quality of video and audio calls suffer due to the presence of background noise. Such noise can be reduced with the concepts used in this project.

1.5 Scope of Project

In this project, the vocal, instrumental, drum, and bass stems can be separated from a song. It has some limitations as well. Since the dataset is limited to songs in the English language and has few instruments only, the system will not work for songs of every language and it cannot extract every instrument in a song.

1.6 Report Organization

This report is divided into 9 chapters. Each chapter discusses different issues related to this project. The outline of each chapter is stated below. A basic introduction about the separation of vocals and instrumentals has been described in chapter 1. The project statement, objectives, and applications of the project have also been described in this chapter. Chapter 2 covers the important background information and history about the different techniques used regarding the separation of vocals and music. Chapter 3 gives information about different software components and the evaluation metrics required for the completion of the project. The analysis of the used dataset is explained in chapter 4. Methodology and working principles of the separation processes are described in Chapter 5. The block diagram of the system as well as the different processes used for separation are also described here. In chapter 6, the implementation of the project is defined. In chapter 7, the outputs are illustrated. The future enhancements which can be made in this project are described in chapter 8. Finally, the conclusion drawn from this project, by the team members, is covered in chapter 9.

2. LITERATURE REVIEW

The separations of audio from a piece of music, isolating the vocals and the instrumentals have been studied for many years. The isolation of the vocals and the instrumentals are possible since both of them have distinguishing features. MSS is carried out by utilizing those features [1]. Different techniques are used for this isolation process. One of the methods is by exploiting both channels of the stereo mix, where the dual channels are inverted and mixed to reduce the vocals [2]. Although this method cancels out most of the vocals, some residues of the vocals are still left.

Music has repeating patterns, where the melody repeats after a small-time reforming the waveforms formed earlier. The melodic components in music audio signals were enhanced and melody trackers were used to track the repetition of the melodies [3]. The separation of the music is done by identifying the period of the repeating structure. Then the whole music is segmented on that period and the segments are averaged out creating a repetitive segment [4].

The separation of instrumentals and vocals can be performed by using the 2DFT (2-Dimensional Fourier Transform) approach [5]. Here the 2DFT is applied to the magnitude spectrogram of the song to detect and extract the temporal repetitions. The peaks of the 2DFT spectrogram are masked. These masked peaks represent the periodic music which are the instrumentals and the unmarked are the vocals. The results from the 2DFT algorithm depended upon the genre of the song. As different genre songs have different repetition pattern, the results depended on it.

Another method by which the voice is separated and extracted is by using a Tandem Algorithm which estimates the singing pitch and separates the singing voice jointly and iteratively [6]. To enhance the performance of the tandem algorithm for dealing with musical recordings, a trend estimation algorithm is used to detect the pitch ranges of a singing voice in each time frame and a Support Vector Machine (SVM) for pitch estimation and voice extraction from music accompaniments [7]. While using the SVM, the dataset should be small in size, as a large dataset has more noise which degrades the performance of SMV.

These techniques are used for the extraction of vocals and instrumentals if the music is dual audio, where both the channels are present and cannot be used for single-channel music.

The extraction of the vocals and the instrumentals was a difficult job for mono-channel music due to the audio being channeled through a single channel. After the involvement of the machine learning methods, this problem has been simplified. Machine learning is used to directly learn a mapping between the mixture and the constitutive sources [8]. Such a strategy recently introduced a breakthrough compared to everything that was done before. The voice separation algorithm based on the Non-Negative Matrix Factorization (NMF) method can effectively extract the singing voice from mono-channel music [9]. NMF has a distinctive ability to decompose the spectrum of mixtures and after the decomposition of the mixture, the voice and the instrumentals are extracted.

The NMF was further revolutionized into Sparse Non-Negative Matrix Factorization (SNFM) and with SNFM and low-rank modeling methods, the separation of the vocals and instrumental music was performed [10]. The SNFM divides the audio file into small fragments and the low-rank modeling method solves the problem by representing those small audio segments as low-rank matrices. Although these, NFM algorithms gave results the results received were only capable of separating the voice and music of one instrument. If more than one instrument were used these algorithms did not give accurate results. Hence different audio filtering algorithms were developed.

The vocals and instrumentals can be separated by using a median filtering-based algorithm [11]. In this technique, median filtering is used on spectrograms to separate harmonic and percussive components in audio signals. It suppresses the percussive events and enhances the harmonics components. Using median filtering along with different factorization techniques, more accurate results were obtained [12]. Although these algorithms could separate instrumentals and vocals, they cannot separate different stems of instruments. This limitation is addressed by using a Deep Neural Network (DNN).

Deep learning utilizes both the ANN and the feature learning methods for generating the mask. It estimates the masks by training multiple deep neural networks to separate different entities of the mixture [13]. The DNN is used in the frequency domain to estimate a mask target instrument from the mixture which allows DNNs to reduce the noise in the targeted output. Although all the noises are reduced, the final output is distorted from the targeted output. Hence DNN is used for single-channel source separation [14].

When using DNNs the estimated output separated entity was distorted from the targeted output so different filtering methods are used to filter the distortion created on the output. The distortion can be filtered out by using a feed-forward architecture and a bidirectional LSTM network [15]. In feed-forward architecture, the information only moves forward whereas bidirectional LSTM allows information to flow in both forward and backward directions.

DLM includes CNNs where each unit neuron in one layer is connected to all the neurons in another layer. The convolutional neural networks are trained by the real-world music of different vocals and instrumental music for identifying and separating them from a mixture of both [16]. Different CNN architectures can be used for the MSS process and among those one of them is U-net convolution network architecture [17].In U-net architecture, each layer is stacked upon another for both up-sampling and down-sampling.

Another CNN architecture in MSS is the hourglass module which captures information at every scale [18]. It was originally designed for estimating the poses of humans but is used in MSS as well. The network learns features from a spectrogram image across multiple scales and generates masks. The estimated mask is refined as it passes over stacked hourglass modules [19].

Supervised deep learning is very effective against problems where there is a huge amount of data. The performance of the deep learning method can be boosted significantly by applying transfer learning [20]. Transfer learning helps in the MSS of instruments whose datasets are not available through the existing datasets.

3. REQUIREMENT ANALYSIS

The music source separation is a signal analysis and source separation problem, which is time and cost-efficient to be solved with programming skills. Hence, the requirements for the project are software tools only.

3.1 Google Colaboratory

Google Colaboratory is an online Jupyter Notebook environment. It is well suited for learning purposes and the platform is easy to collaborate with other team members. It provides free access to computing resources including GPU. While the resources are free, they are limited. For this project, the 2DFT approach is implemented in the Google Colaboratory platform.

3.2 Google Drive

Google Drive is a cloud-based storage solution, that allows to save files online and access them anywhere. It makes it easy for the team members to edit and collaborate on files. Google drive is used along with google colaboratory to save and load data required for the project.

3.3 Kaggle

Kaggle also provides an online Jupyter Notebook environment. Besides providing a coding platform, it also hosts the dataset in its server, which can be shared among the team members. It provides free access to resources including RAM, CPU, and GPU. For this project, the quick prototyping of the machine learning approach is implemented in the Kaggle platform.

3.4 Google Cloud

Google Cloud is a cloud computing platform. It provides many services such as computing and database storage. However, for this project, only computing power is used. Google cloud platform provides free 300$ credit, which is used to purchase the virtual machine of required configuration.

3.5 Programming Language and Libraries

The programming language used in this project is Python programming language. The libraries used for the project are mentioned below:

- Librosa – It is used for loading audio signals. It provides functions for performing signal processing and generating spectrograms.

- SciPy – It is used for determining the maximum and minimum value over a matrix.

- Museval – It provides the functions for the calculation of different evaluation metrics such as SAR and SDR.

- Matplotlib – It is used for creating different plots and graphs for the project.

- NumPy – It is used for working with arrays.

- TensorFlow – It is used for building and deploying the machine learning model.

- Scaper – This library is used for audio augmentation.

3.6 Evaluation Metrics

Evaluation Metrics are required to measure the performance of the project. Some of the evaluation metrics used in the project are as follows:

- Signal to Noise Ratio (SNR)

  $$SNR = 10\log_{10}\left(\frac{||\, s_{target}\,||^2}{||\, s_{target} - \hat{s}\,||^2}\right) \quad 3.1$$

  Where $\hat{s}$ is the estimate of s_{target}.

  The value of SNR is in dB. The larger the value of SNR, the better is the estimation of the signal.

- Cosine Similarity

$$Cosine\ Similarity(A,B) = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad 3.2$$

The cosine similarity of 0 indicates that the signals A and B are dissimilar and the cosine similarity of 1 indicates that the signals A and B are the same.

- Signal to Distortion Ratio (SDR)

$$SDR = 10\ \log_{10}\left(\frac{||\ s_{target}\ ||^2}{\left\|e_{interf} + e_{noise} + e_{artif}\right\|^2}\right) \quad 3.3$$

Where e_{interf} is the error terms for interference

e_{noise} is the error terms for noise

e_{artif} is the error terms for artifact

SDR is usually considered to be an overall measure of how good a source sound. Its value is in dB.

- Signal to Artifact Ratio (SAR)

$$SAR = 10\ \log_{10}\left(\frac{\left\|s_{target} + e_{interf} + e_{noise}\right\|^2}{\left\|e_{artif}\right\|^2}\right) \quad 3.4$$

SAR is usually interpreted as the number of unwanted artifacts a source estimate has with relation to the true source. Its value is also in dB.

4. DATASET ANALYSIS

Some of the datasets available for music source separation projects are listed in the following table:

Table 4-1: Open-Source Datasets

S.N	Dataset	Year	Instrument Categories	Tracks	Average Duration (sec)
1.	MASS	2008	N/A	9	16 ± 7
2.	ccMixter	2014	N/A	50	231 ± 77
3.	MedleyDB	2014	82	63	206 ± 121
4.	DSD100	2015	4	100	251 ± 60
5.	MUSDB18	2017	4	150	236 ± 95
6.	Slakh2100	2019	34	2100	249

The columns of the table indicate the key characteristics to consider when choosing a dataset for music source separation. Generally, having a greater number of data is better for the project. But quantity isn't enough, quality and variability are of great importance as well. Some datasets in the past provided only small excerpts of the songs, however, new datasets are providing data of full songs which is richer.

Among all the datasets, the most suitable dataset for the project is Demixing Secrets Dataset 100 (DSD100) [21] and MUSDB18 [22]. This is because the size of the dataset is reasonable and contains songs and stems from various genres. The detail of these datasets is given in table 4-2 and table 4-3 respectively.

Table 4-2: Details of DSD100 Data Set

S.N	Track Genre	Number of tracks	Encoded Frequency (KHz)
1.	Pop/Rock	73	44.1
2.	Heavy Metal	11	
3.	Electronic	6	
4.	Rap	5	
5.	Reggae	2	
6.	Jazz	2	
7.	Country	1	

The DSD100 dataset consists of 100 full-track songs of different genres. For every song in the dataset, it provides its vocal, instrumental, bass, and drum stems. The time duration of each stem is equal to that of the original song.

The total time duration of songs in the dataset is 6 hours 56 minutes and 40 seconds. This dataset also provides a small subset of testing data consisting of only 4 tracks having a duration of 30 seconds each. All the music is in the extension of Waveform Audio File Format (.wav).

Along with the DSD100 dataset, the MUSDB18 dataset is also used in the project, which contains a total of 150 different songs from different genres. MUSDB18 dataset is formed by combining songs from different datasets. Among the 150 tracks in the MUSDB18 dataset, 100 tracks are taken from the DSD100 dataset, 46 tracks are taken from the MedleyDB, 2 tracks were provided by Native Instruments and 2 tracks are from the Canadian rock band - The Easton Ellises. The detail of this dataset is stated in table 4-3.

Table 4-3: Details of MUSDB18 Data Set

S.N.	Track Genre	Number of Tracks
1.	Country	3
2.	Electronic	8
3	Heavy Metal	12
4.	Jazz	3
5.	Pop/Rock	11
6.	Pop	72
7.	Rap	8
8.	Reggae	2
9.	Rock	17
10.	Singer/ Songwriter	14

The total time duration of the tracks is about 10 hours. Similar to the DSD100 dataset, the MUSDB18 dataset also provides separate vocal, instrumental, drum, and bass stems of the track, and its duration is the same as of the song with which it is associated to. All sound files are in the extension Waveform Audio File Format (.wav) and are encoded at 44.1 KHz.

5. SYSTEM ARCHITECTURE AND METHODOLOGY

This project has been attempted using two approaches, initially using the 2DFT approach and then the machine learning approach. This section describes the theoretical considerations and the system block diagram required for those approaches.

5.1 Theoretical Considerations

The process of extracting vocals and instrumentals is done by changing the audio signal from time to frequency domain. This results in an image representation of the audio signal known as a spectrogram.

5.1.1 Short Time Fourier Transform

In STFT, longer time signals are broken down into shorter segments of equal length and then the Fourier transform is computed separately on each segment. The function to be transformed is multiplied by a window that is nonzero for only a short period.

The STFT of an audio signal x(t) is calculated as follows:

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) w(t-\tau) e^{-j\omega t}\, dt \qquad 5.1$$

To convert the frequency domain signal back to the time domain, inverse-STFT is used. It is calculated as follows:

$$x(t) = \int_{-\infty}^{\infty} \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} X(\tau, \omega) e^{j\omega t}\, d\omega \right] d\tau \qquad 5.2$$

Where, $x(t)$ = Audio signal

$w(t)$ = Window Function

5.1.2 Window Functions

Window functions are those functions in which the amplitude decreases gradually and smoothly toward zero at the edges. They can be used in two ways, overlapping and non-overlapping.

When non-overlapping windows are used, the data near and at the edges of the window are lost as the value of the window towards the edge is zero. This loss of data causes irregularities in the spectrum.

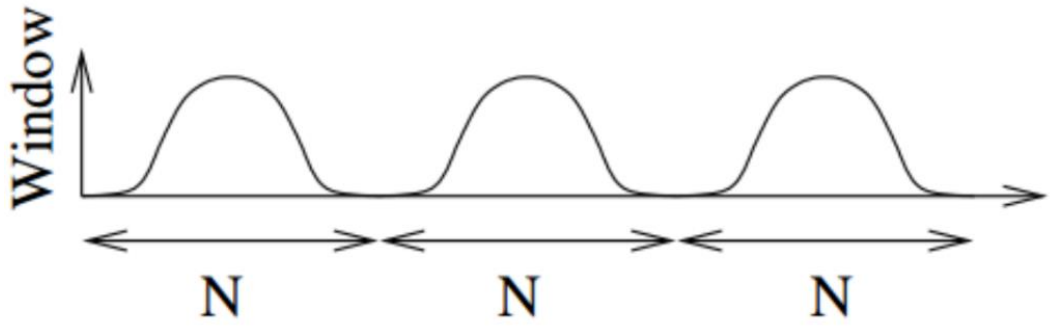


Figure 5-1: Non-Overlapping Windows

Since non-overlapping windows cause the loss in the data, to mitigate these losses certain overlapping is done in between the windows.

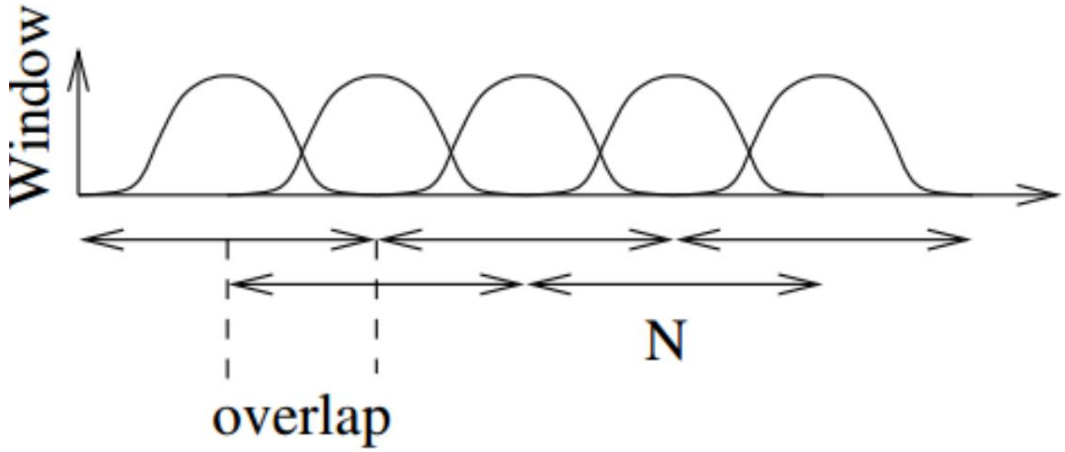


Figure 5-2: Overlapping Windows

The time taken for the start of the next window from the start of the previous window is called the hop length. In the overlapping case, the hop length is always less than the window length (N). Whereas for non-overlapping cases the hop length is equal to or greater than the window length (N).

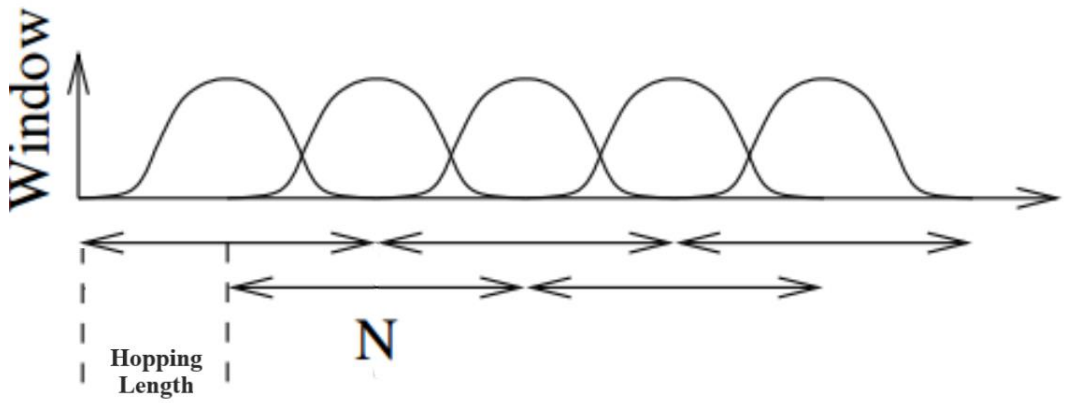


Figure 5-3: Hopping in Overlapping Window

Three types of windows are used for the project with each having different characteristics and functions.

5.1.2.1 Hanning Window

The Hanning window has a shape similar to that of half a cycle of a cosine wave. The following equation defines the Hanning window:

$$w(t) = 0.5 - 0.5 \times \cos\left(\frac{2\pi t}{T}\right) \qquad 5.3$$

Where, $w(t) =$ Hanning window function

T = Time period of the window

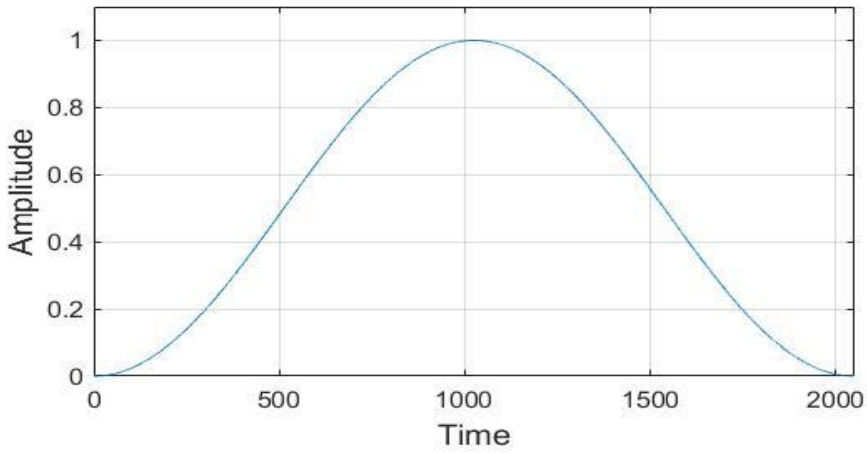


Figure 5-4: Hanning Window Waveform

5.1.2.2 Sqrt-Hanning Window

The Sqrt-Hanning window is obtained by taking the square root of the Hanning window. The following equation defines the Sqrt-Hanning window:

$$w(t) = \sqrt{0.5 - 0.5 \times \cos\left(\frac{2\pi t}{T}\right)} \quad 5.4$$

Where, $w(t)$ = Sqrt-Hanning window function.

T = Time period of window

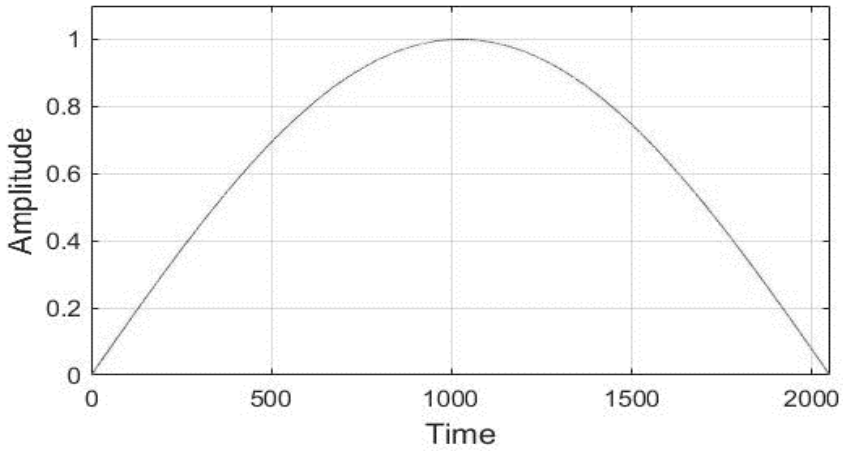


Figure 5-5: Sqrt-Hanning Window Waveform

5.1.2.3 Blackman Window

The Blackman window is given by a sum of cosine terms. By varying the number and coefficients of the terms different characteristics can be optimized. The following equation defines the Blackman window with 3 terms:

$$w(t) = 0.42 - 0.5 \times \cos\left(\frac{2\pi t}{T}\right) + 0.08 \times \cos\left(\frac{4\pi t}{T}\right) \quad 5.5$$

Where, $w(t)$ = Blackman window function.

T = Time period of the window

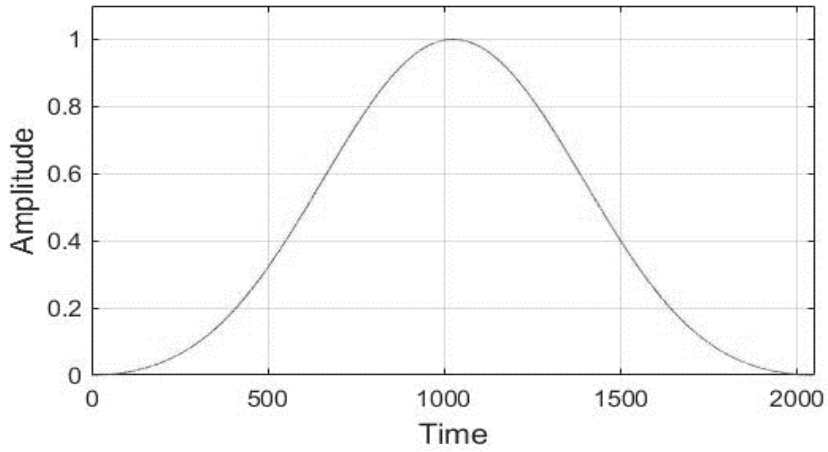


Figure 5-6: Blackman Window Waveform

5.1.3 Characteristics of Windows

The Fourier Transform gives the frequency response of the window signal which contains different lobes of magnitude as shown in figure 5-7. These lobes define the characteristics of window functions.

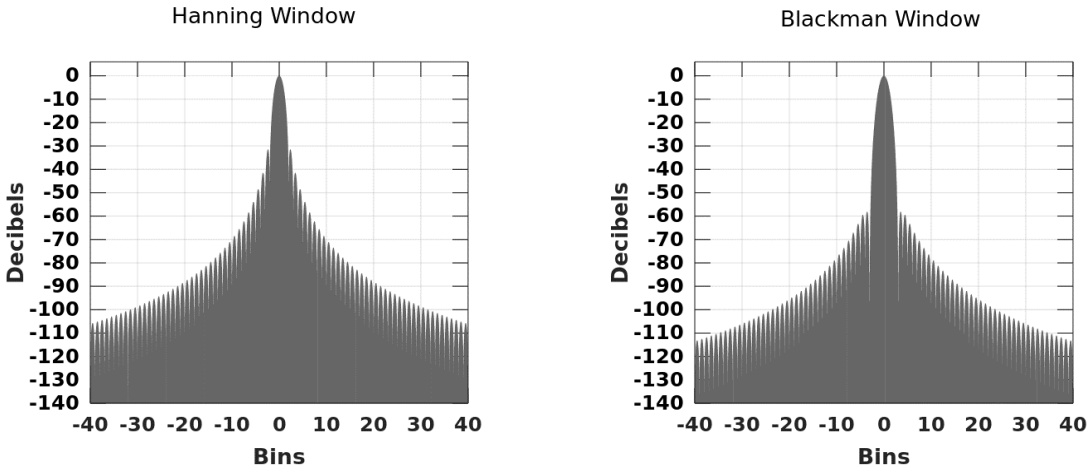


Figure 5-7: Frequency Response of a Window Function

Different windows have distinct characteristics such as first side lobe level, side lobe roll-off rate, the width of the main lobe, and the ratio of main-lobe energy to the total energy. In table 5-1, ‘N’ represents the number of samples.

Table 5-1: Characteristics of Window Functions [23]

S.N	Window	First Side-Lobe Level (dB)	Side lobe roll-off rate (dB per octave)	Width of the main lobe	Ratio of Main-Lobe Energy to Total Energy
1.	Hanning	-31.47	-18	$\frac{8\pi}{N}$	0.999485
2.	Blackman	-58.21	-18	$\frac{12\pi}{N}$	0.999990

5.1.4 Two-Dimensional Fourier Transform

Two-Dimensional Fourier Transform can be implemented as a sequence of One-Dimensional Fourier Transform operations, performed independently along the two axes. Generally, it is calculated by taking the Fourier transform along the row then along the column.

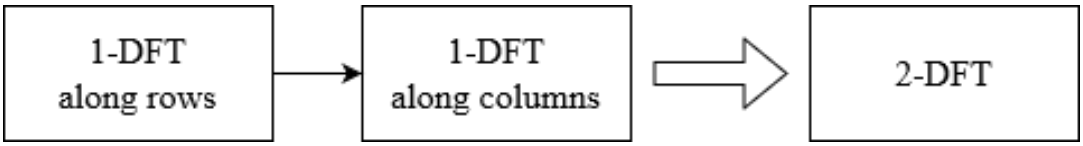


Figure 5-8: 2DFT Computation

The 2DFT and its inverse are calculated as follows:

$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-j(\omega_x x + \omega_y y)} dx dy \quad 5.6$$

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega_x, \omega_y) e^{j(\omega_x x + \omega_y y)} d\omega_x d\omega_y \quad 5.7$$

Where, $f(x,y)$ = Spectrogram of the audio signal

$F(\omega_x, \omega_y)$ = 2DFT of the spectrogram

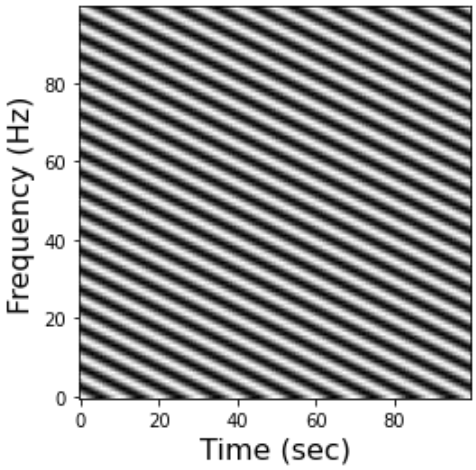


Figure 5-9: Periodic Signal

Figure 5-9 shows a periodic signal whose repetition can be seen as peaks in the 2DFT (scale-rate) plot shown in figure 5-10. For the 2DFT approach, the instrumentals are identified using this property of 2DFT, which converts repetition in the frequency domain to peaks in the scale-rate domain.

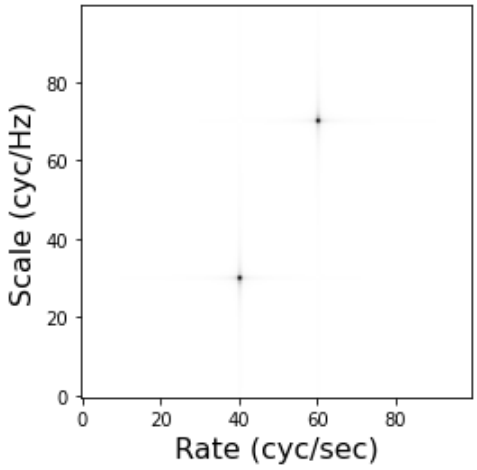


Figure 5-10: Scale Rate Plot of Periodic Signal

5.1.5 U-Net

U-Nets input a spectrogram and perform a series of 2D convolutions, each of which produces an encoding of a smaller and smaller representation of the input. The small representation at the center is then scaled back up by decoding with the same number of 2D deconvolutional layers (sometimes called transpose convolution), each of which

corresponds to the shape of one of the convolutional encoding layers. Each of the encoding layers is concatenated to the corresponding decoding layers.

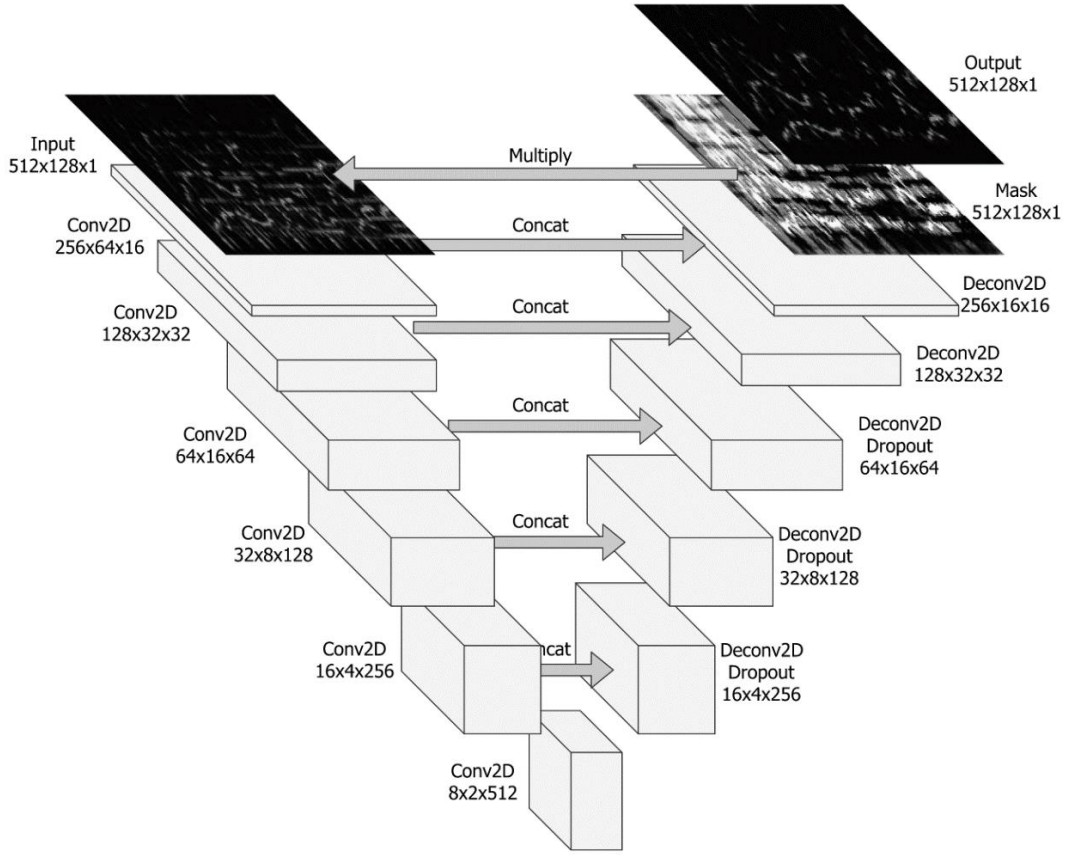


Figure 5-11: U-Net Architecture [24]

5.1.6 Convolution Operation

Convolution is a process where a small matrix of numbers (called kernel or filter) is taken, which transforms the input image based on the filter values. Subsequent feature map values are calculated according to the following formula, where the input image is denoted by “f” and kernel by “h”. The indexes of rows and columns of the result matrix are marked with “m” and “n” respectively.

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k] \quad 5.8$$

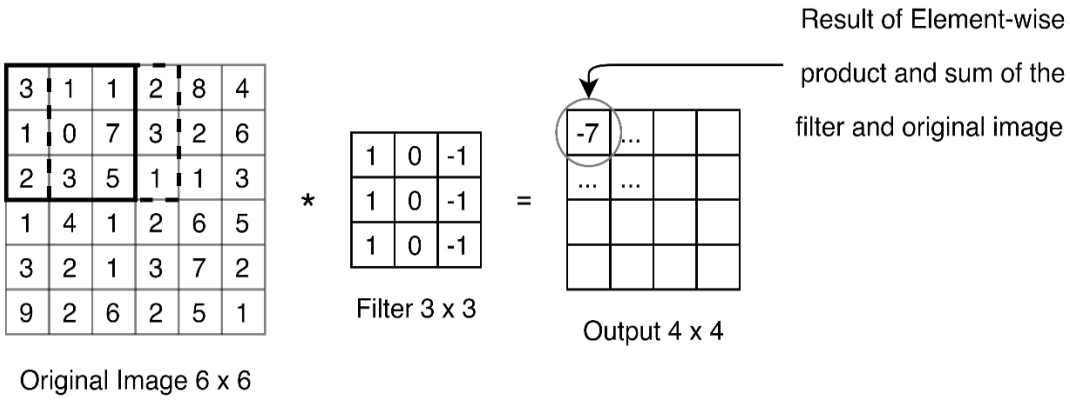


Figure 5-12: Convolution Process

The output size from convolution is affected by two parameters namely padding and stride.

5.1.6.1 Stride

Stride is the number of pixel shifts over the entire input matrix. When the stride is 1, the filters shift 1 pixel at a time. When the stride is 2, the filters shift 2 pixels at a time.

5.1.6.2 Padding

Padding is the number of pixels added to the image boundary. This is done to preserve the input dimension as repeated convolution can remove important features. There are many types of padding. Some of them are Zero padding, which adds zero in the image boundary, and Valid padding which drops the part of the image where the filter did not fit the image and keeps only a valid part of the image.

The formula for calculating the output size for any given convolution layer is:

$$O = \frac{(W-K+2P)}{S} + 1 \qquad 5.9$$

where O is the output height/length,

W is the input height/length,

K is the filter size,

P is the padding,

and S is the stride.

5.1.7 Transposed Convolution Operation

The transposed convolutional layer, unlike the convolutional layer, is up-sampling in nature. Transposed convolutions are usually used in a network that must reconstruct an image. In a transposed convolution, instead of the input being larger than the output, the output is larger.

5.1.8 Activation Function

The activation function decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron. A neural network without an activation function is essentially just a linear regression model. The activation function does the non-linear transformation to the input making it capable to learn and perform more complex tasks. There are many activation functions. Some of them are the Sigmoid Function, Hyperbolic Tangent Function (Tanh), Identity Function, and Rectified Linear Unit (ReLU) Function.

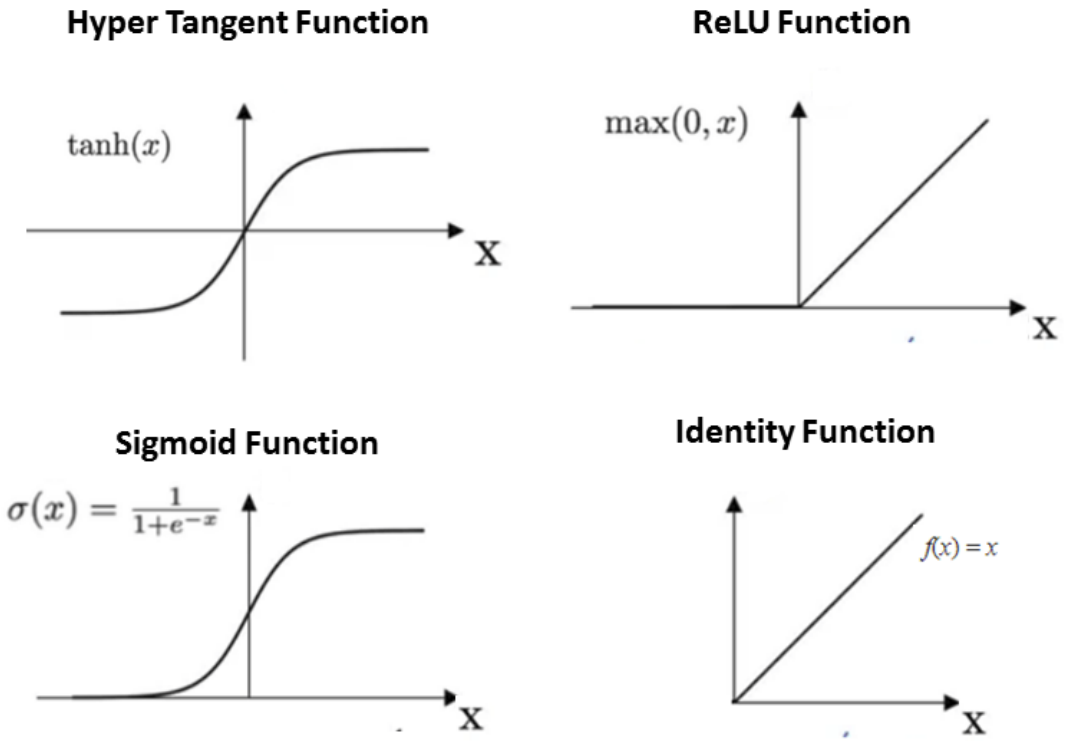


Figure 5-13: Activation Functions [25]

5.1.9 Data Augmentation

Most of the time, the data in the dataset is not enough for training the machine learning model. So, data augmentation is used to generate more data from the existing data. In the project as well, data augmentation is performed. In the augmentation of music signals, there are two types of mixing.

5.1.9.1 Coherent mixing

In coherent mixing, a piece of new music is created by mixing stems from the same track. In this type of mixture, only one track is used. While using coherent mixing, only the pitch value, SAR, SNR, SAR, or the speed of the song is changed, so the new music created is similar to the original track. Due to this reason, the variation required to train the model cannot be achieved.

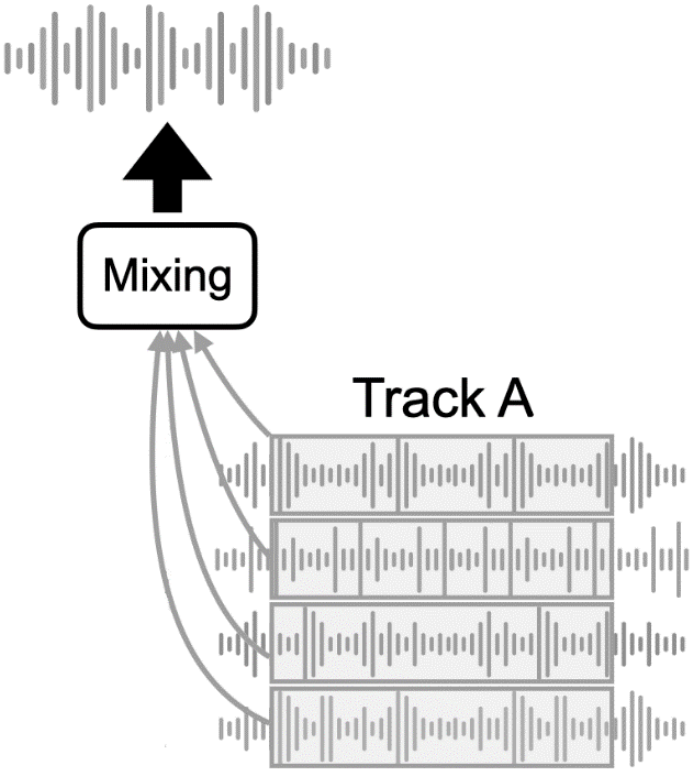


Figure 5-14: Coherent Mixing [24]

5.1.9.2 Incoherent mixing

In incoherent mixing, a piece of new music is created by mixing stems from different tracks. In this type of mixing two or more tracks are used. While using incoherent mixing, the new music is always a piece of unique music. Due to this reason, infinite

types of music can be generated. It also provides more variation to the dataset which in turn helps in better training of the model.

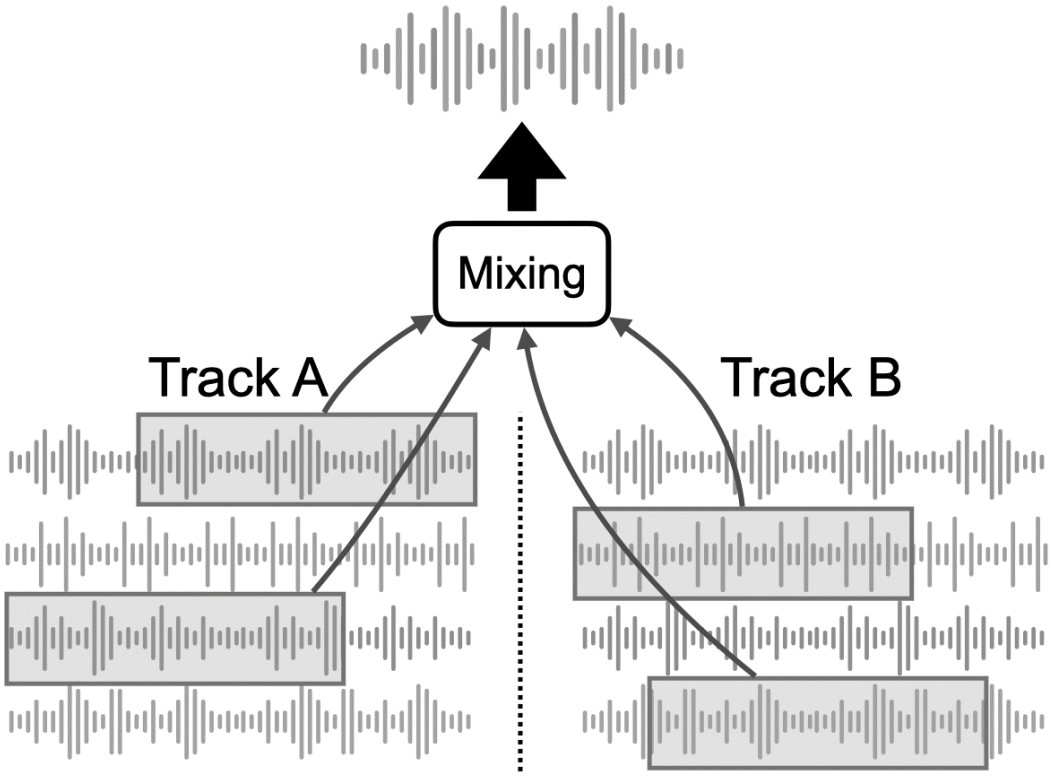


Figure 5-15: Incoherent Mixing [24]

5.2 System Block Diagram

For the separation process, two separate approaches are used. Initially, the project was attempted using the 2DFT approach and then the machine learning approach was used.

5.2.1 2DFT Approach

The audio signals taken as input are songs in the time domain. The mixture is separated into two entities, instrumentals and vocals through masking in the frequency domain. The final output of the system is also in the time domain.

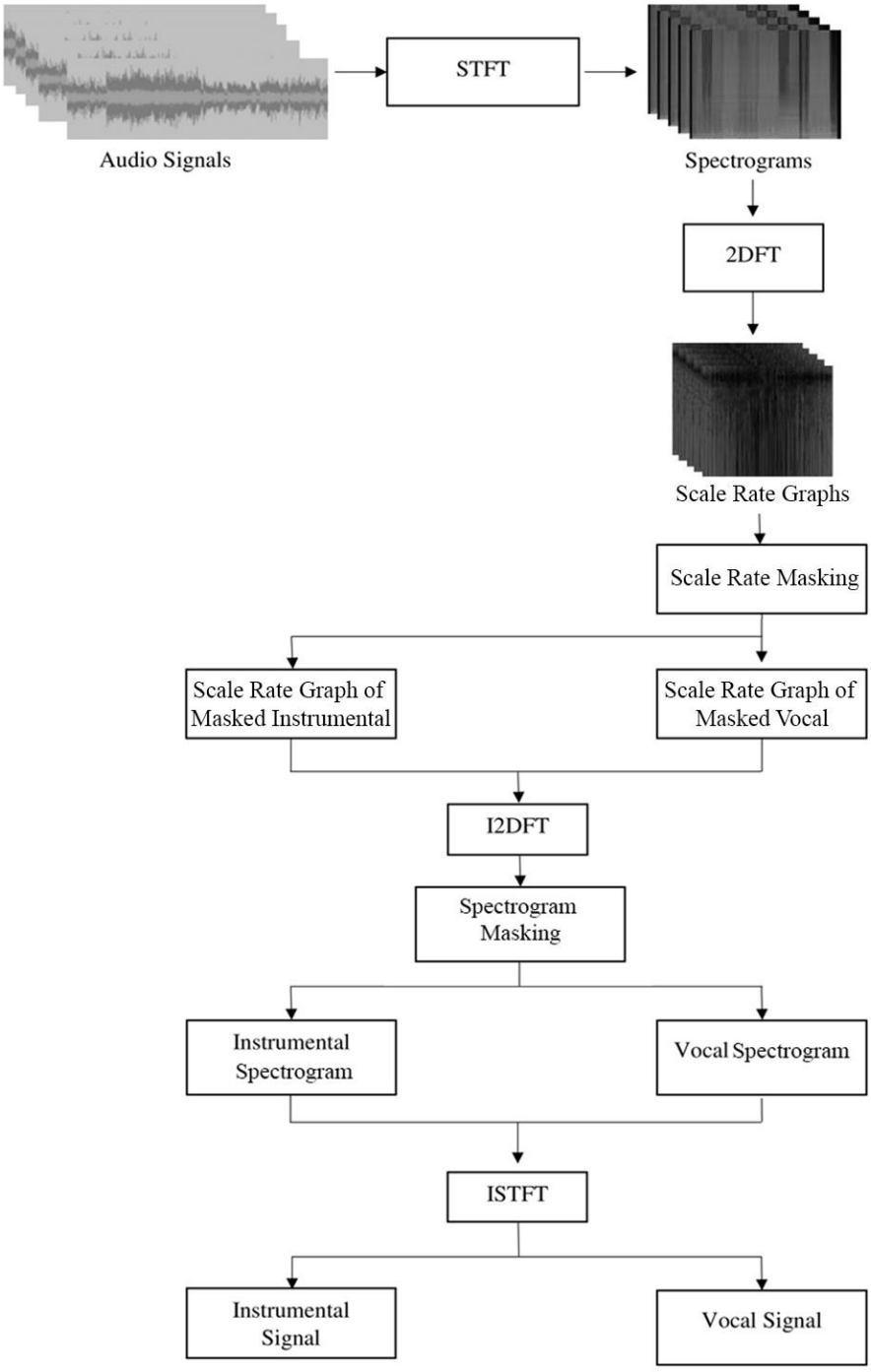


Figure 5-16: Block Diagram of 2DFT Approach

5.2.1.1 Audio Signals

The audio signals are songs that are a mixture of both instrumentals and vocals, that are to be separated. It contains songs from the dataset or the audio signals that come from the augmentation process.

5.2.1.2 Time to Frequency Domain

The time-domain signal is converted into the frequency domain via STFT. The audio signals in the frequency domain are known as spectrograms. The spectrograms are supplied to the 2DFT block.

5.2.1.3 Scale Rate Graph

The spectrograms are then converted to a scale-rate graph using 2DFT. The obtained result is in complex form and thus only the magnitude is retained.

5.2.1.4 Masking and Spectrogram Separation

The binary mask of instrumentals and vocals is generated from the scale rate graphs. These masks are then applied to the mixture in the scale-rate form to get the scale-rate graphs of instrumental and vocal separately. Then inverse-2DFT is applied to the separated graphs to obtain two spectrograms. Those two spectrograms are compared with one another to form a binary mask for the instrumental (background) signal. The mask for the vocal (foreground) signal is obtained by inverting the mask of the instrumental signal. These masks are applied to the mixture spectrogram to obtain separate spectrograms of instrumental and vocal.

5.2.1.5 Output Generation

The output of the system is obtained by applying the inverse-STFT to the separated instrumental and vocal spectrograms. This gives the time domain signal of the instrumental and vocal signal.

5.2.2 Machine Learning Approach

The machine learning approach is performed using a CNN model, i.e., U-Net model. The block diagram shown in the figure 5-17, receives the audio signal as the input. The audio signals are the songs from the dataset of the data after the augmentation process. The output of the model is vocal, instrumental, drum, and bass stems.

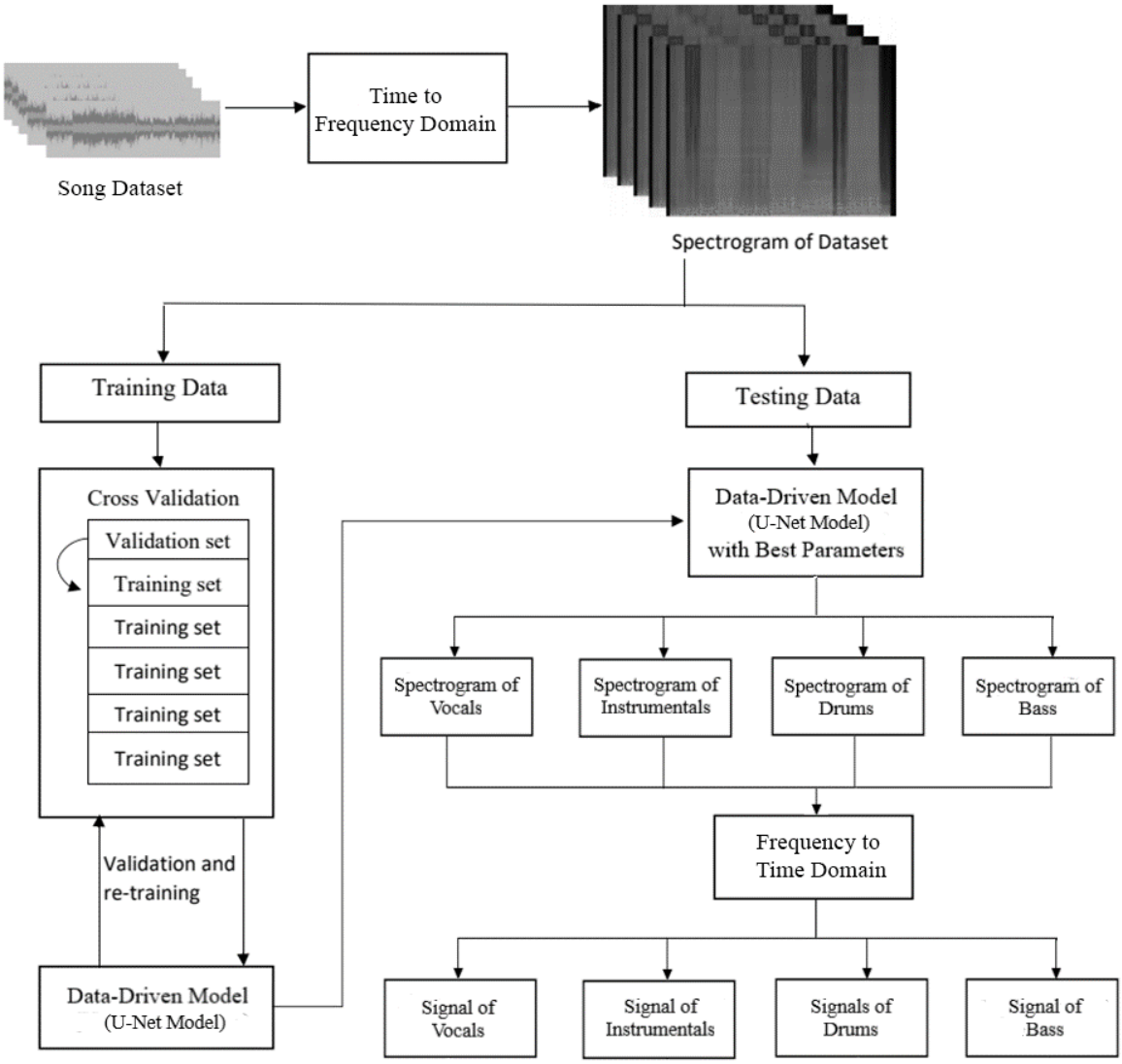


Figure 5-17: Block Diagram of Machine Learning Approach

5.2.2.1 Songs Dataset

The songs dataset contains all the songs along with their separate vocal, instrumental, drum, and bass stems. The songs are converted to frequency domain which is then split into train and test datasets.

5.2.2.2 Time to Frequency Domain

The time to frequency domain conversion process is the same as that in the 2DFT approach. The obtained spectrogram is sent to the model for either training, validation,

or testing. The training spectrogram will go towards the cross-validation block and the testing spectrogram will go towards the testing of the trained model.

5.2.2.3 Cross-Validation

Cross-validation splits the training data into many groups. Those grouped data are divided into training and validation sets which are then provided for the training phase and the validation phase. After completion of one phase, a new group of data is selected as a validation set and other sets are used as the training set.

5.2.2.4 Data-Driven Model

The model architecture used for the project is U-Net. It receives training data from the dataset, which is used for training. During the training of the model, weights and biases are tuned. A mask is created from the model and is convoluted with the spectrogram to produce an initial result. From this result, a loss is calculated and the tuning process is adjusted. After the training is completed the validation phase starts.

In the validation phase, the hyper-parameters like learning rate, the batch size are tuned and the model becomes refined. The model undergoes re-training until the loss from the model is satisfactory. The testing of the model is started when the validation produces satisfactory results. The trained model is then used to predict the output from the test dataset and the evaluation metrics are calculated.

5.2.2.5 Frequency to Time Domain

The output from the machine learning model is the spectrogram of the vocal, instrumental, drum, and bass stems. These stems are converted back to the time domain signal by using the inverse-STFT.

5.2.2.6 Separated Signal

These are the isolated vocal, instrumental, drum, and bass stems of the songs, which is the output of the project.

6. IMPLEMENTATION DETAILS

This section deals with the specifics related to the 2DFT approach and the machine learning approach.

6.1 2DFT Approach

The 2DFT approach works on the principle of repetition, which is seen on the scale rate graph, which is obtained by taking the 2DFT of the spectrogram. The repetition is due to the periodic signal, which is the instrumental stem. The vocal signal is aperiodic.

6.1.1 Windowing Parameters

While performing the STFT of a signal, window functions of finite length are used to divide the signal into different segments. For this project, the window length taken is 1024 samples and the hopping length taken is 690 samples. This results in the spectrogram of (512,128) for a song of 2-second length.

6.1.2 Spectrogram Formation

The song is a time-domain signal, which is the mixture of both instrumental and vocal signals, which are to be separated. Short-time Fourier Transform of the time domain signal $x(t)$ is taken and is denoted by $X(\omega, t)$.

$$X(\omega, t) = STFT[x(t)] \tag{6.1}$$

Here, $X(\omega, t)$ is in the complex form, so the magnitude of Fourier transformed signal $X(\omega, t)$ is computed which gives the spectrogram of the signal.

6.1.3 2DFT of Spectrogram

The 2DFT of the spectrogram is taken, which is represented by $\tilde{X}(s, r)$ where 's' and 'r' represent the scale (vertical) and rate (horizontal) axes. The terms are borrowed from the studies of the auditory system in mammals [5]. Scale denotes the repetition in the frequency domain and rate denotes temporal repetitions.

$$\tilde{X}(s, r) = FT_{2D}\{|X(\omega, t)|\} \tag{6.2}$$

6.1.4 Determination of Neighborhood Size

The peaks are located by computing the range between the maximum and minimum values of the magnitude of $\tilde{X}(s,r)$ i.e., $\left|\tilde{X}(s,r)\right|$ over a neighborhood. The neighborhood is chosen to be a rectangle on the scale-rate graph domain with center at $C(s_c, r_c)$ and the neighborhood surrounding this point is denoted by $N(c)$.

The size of the neighborhood along the scale axis is always 1 whereas along the rate axis of the neighborhood varies from 15 to 100 frames. The above selection is done because the repetition is only found along the rate axis.

6.1.5 Formation of Scale-Rate Masks

For the spectrogram of instrumental signal, a background mask $M_{bg}(s,r)$ is created which takes binary values.

If $\alpha_c > \gamma$ and $\underset{N(c)}{max}\,|x(s,r)| = |x(s,r)|$, then $M_{bg}(s,r) = 1$, otherwise $M_{bg}(s,r) = 0$.

Where

$$\gamma = standard\ deviation(\left|\tilde{X}(s,r)\right|)$$

$$\alpha_c = \underset{N(c)}{max}\left|\tilde{X}(s,r)\right| - \underset{N(c)}{min}\left|\tilde{X}(s,r)\right|$$

Similarly for vocal spectrogram, a foreground mask $M_{fg}\,(s,r)$ is created as:

$$M_{fg}(s,r) = 1 - M_{bg}(s,r) \qquad 6.3$$

6.1.6 Spectrogram Masking

The scale-rate graphs are element-wise multiplied by the scale-rate masks. The separate magnitude spectrogram of both vocals, $\left|X_{fg}(\omega,t)\right|$ and instrumentals $\left|X_{bg}(\omega,t)\right|$ are generated by taking the inverse 2DFT of both foreground and background masked and is given by:

$$\left|X_{fg}(\omega,t)\right| = IFT_{2D}\{X(s,r) \odot M_{bg}(s,r)\} \qquad 6.4$$

$$\left|X_{bg}(\omega,t)\right| = IFT_{2D}\{X(s,r) \odot M_{fg}(s,r)\} \quad 6.5$$

For the separation of instrumentals signal, a background mask $M_{bg}(\omega,t)$ is created which takes 0 or 1 values.

If $\left|X_{bg}(\omega,t)\right| > \left|X_{fg}(\omega,t)\right|$, then $M_{bg}(\omega,t) = 1$, otherwise $M_{bg}(\omega,t) = 0$.

Similarly for vocal signals, a foreground mask $M_{fg}(\omega,t)$ is created as:

$$M_{fg}(\omega,t) = 1 - M_{bg}(\omega,t) \quad 6.6$$

6.1.7 Retrieval of Signals

The time-domain vocal and instrumentals are extracted from their respective separated spectrograms by taking the inverse STFT. The vocal signal and instrumental signal are denoted by $x_{fg}(t)$ and $x_{bg}(t)$ and are given as:

$$x_{bg}(t) = ISTFT\{X(\omega,t) \odot M_{bg}(\omega,t)\} \quad 6.7$$

$$x_{fg}(t) = ISTFT\{X(\omega,t) \odot M_{fg}(\omega,t)\} \quad 6.8$$

6.2 Machine Learning Approach

The machine learning approach works on a model of U-Net architecture. The model is trained, verified, and tested to separate the stems from a song.

6.2.1 Dataset

The separation model is assessed using different evaluation metrics using the MUSDB18 dataset. The songs in the dataset are broken down into segments of 2 seconds.

6.2.2 Dataset Splitting

The overall dataset is divided into the training dataset and the test dataset. The ratio of the train to test set is 3:2. The training set undergoes 5-fold cross-validation, from which four set is used for training and one set is used for validation. The advantage of keeping a test set that the model hasn't seen during the training is to avoid over-fitting the model.

6.2.3 U-Net Architecture

The input to the U-Net architecture used in the project is a (512,128) sized spectrogram. The U-Net is designed with six 2D convolutional layers with 5x5 kernel sizes and strides of 2. After each convolutional layer, batch normalization is used which is followed by a ReLU activation. A Dropout of 50% is applied to the first three convolutional layers between batch normalization and activation function.

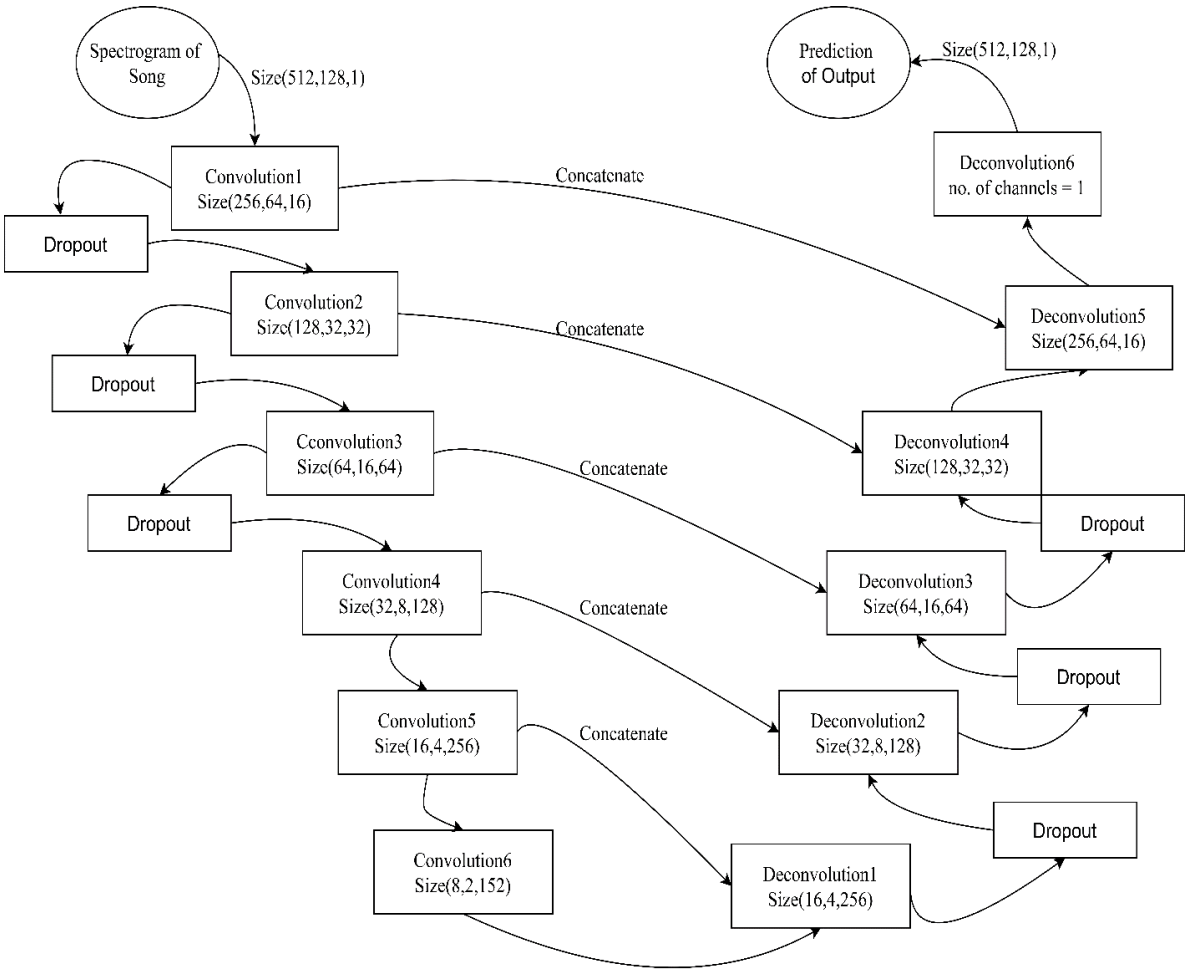


Figure 6-1: Designed U-Net Architecture

After the 6th convolutional layer, 5 transposed convolutional layers with the same kernel and stride sizes are used. Similar to the convolutional layers, after each transposed convolutional layer, there is a batch normalization followed by ReLU activations. The first 3 layers of transposed convolutional layers contain a dropout of 50%, between batch normalization and ReLU activation. The final layer of the model has a sigmoid activation function that makes a mask.

The final mask is multiplied with the input mixture and the loss is taken between the ground truth source spectrogram and mixture spectrogram with the estimated mask applied.

6.2.4 Batch Normalization

Batch normalization is a technique for improving the performance and stability of neural networks. Batch normalization depends upon the activation function, as it regulates the values going into each activation function. It stabilizes the learning process and dramatically reduces the number of training epochs required to train deep networks.

6.2.5 Loss Calculation

The loss function calculates how far the predicted value is from the actual value. This defines the amount of time the model is to be trained. The loss function used in our project is the mean squared error.

$$Mean\ Squared\ Error = \frac{1}{m}\sum_i abs(y_i - f(x))^2 \quad 6.9$$

Where y_i = Actual Value

$f(x)$= Predicted Value

m = Number of Samples

6.2.6 Optimizers

Optimizers are algorithms or methods used to change the attributes of a neural network such as weights and learning rate to reduce the losses. Optimization algorithms are responsible for reducing the losses and providing the most accurate results possible. There are many optimizers, some of them are ADAM, Gradient Descent, Momentum. Among these optimizers, for this project, the ADAM optimizer was used as it works well with large data sets and large parameters.

Adaptive Moment Estimation (ADAM) is an algorithm for optimization technique for gradient descent. The method is efficient when working with large problems involving

a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm.

- Momentum

  This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the 'exponentially weighted average' of the gradients. Using averages makes the algorithm converge towards the minima at a faster pace.

$$\omega_{t+1} = \omega_t - \alpha m_t \tag{6.10}$$

  Where,

$$m_t = \beta m_{t-1} + (1-\beta)\frac{\delta L}{\delta \omega_t} \tag{6.11}$$

  Here, m_t = aggregate of the gradient at the current time

  m_{t-1} = aggregate of the gradient at the previous time

  ω_t = weight at the current time

  ω_{t+1}= weight of next instant of time

  α = learning rate at the current time

  δL = Derivative of the loss function

  $\delta \omega_t$ = Derivative of wright at time "t"

  β = Moving average parameter

- Root Mean Square Propagation (RMSP)

  Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve AdaGrad. Instead of taking the cumulative sum of squared gradients like in AdaGrad, it takes the 'exponential moving average'.

$$\omega_{t+1} = \omega_t - \frac{\alpha_t}{(v_t + \epsilon)^{1/2}} \times \frac{\delta L}{\delta \omega_t} \quad 6.12$$

Where,

$$v_t = \beta v_{t-1} + (1-\beta)\left[\frac{\delta L}{\delta \omega_t}\right]^2 \quad 6.13$$

Here, ω_t = weight at the current time

ω_{t+1}= weight of next instant of time

α_t= learning rate at time "t"

δL = Derivative of the loss function

$\delta \omega_t$ = Derivative of wright at time "t"

v_t = Sum of the square of past gradients

β = Moving average parameter

ϵ = A small positive constant

6.2.7 Hyper-parameter Tuning

Hyper-parameter tuning is choosing a set of optimal hyper-parameters for a learning algorithm. A hyper-parameter is a model argument whose value is set before the learning process begins. The hyperparameters in our project are, the learning rate is 1e^-2, the number of epochs is 100 and the activation function used is ReLU and Sigmoid.

7. RESULTS AND ANALYSIS

The result from all the steps explained in the implementation details of the project is explained here. For the result analysis of the 2DFT and machine learning approach, a random song from the dataset is chosen. Furthermore, the evaluation metrics such as SNR, SDR, SAR, and cosine similarity have been calculated and compared using songs from every genre present in the dataset.

7.1 Result from 2DFT Approach

The song “Arise - Run Run Run” of the reggae genre is chosen for the result analysis of the 2DFT approach. The waveform of the song is shown in figure 7-1.

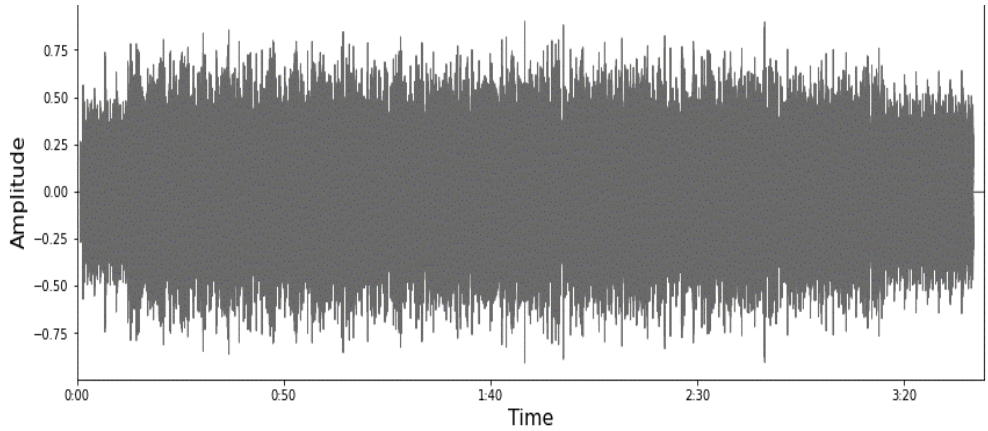


Figure 7-1: Waveform of Run Run Run Song

7.1.1 Formation of Spectrogram

The spectrogram of the song is generated by taking its STFT. The window function used is the Hanning Window. The unit is on the Decibel scale.

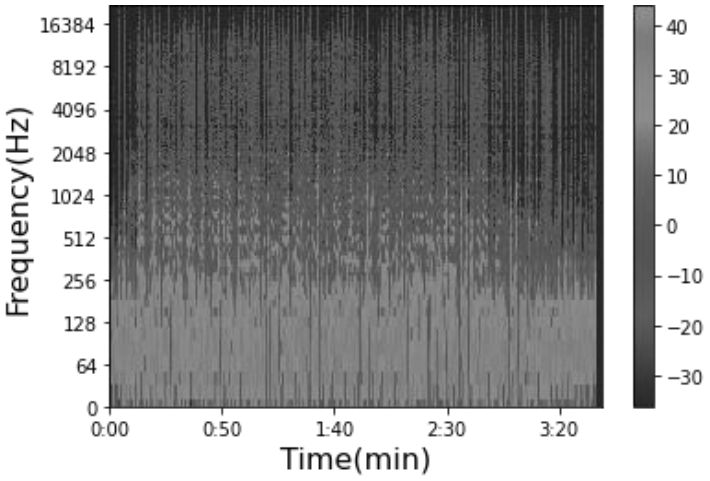


Figure 7-2: Spectrogram of Run Run Run Song

7.1.2 Scale-Rate Graph of Song

The generated spectrogram is in complex form, so 2DFT of the magnitude of the spectrogram is taken, which is in terms of scale and rate. It is shown in figure 7-3.

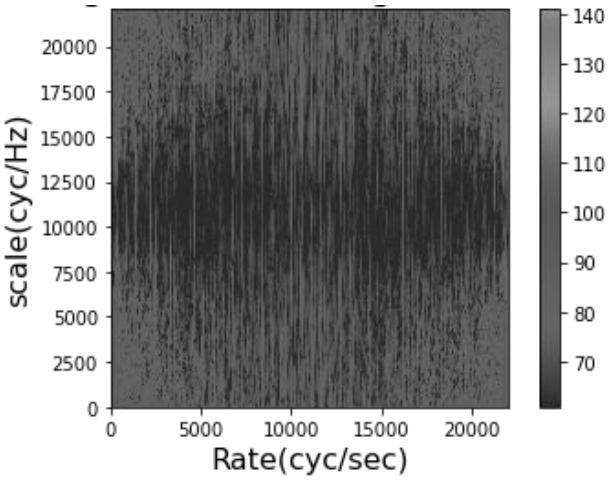


Figure 7-3: Scale Rate Graph of Song

7.1.3 Mask of Scale Rate Graph

The masks created in the scale rate domain are shown in Figures 7-4 and 7-5. For the mask, ‘1’ represents the black color and ‘0’ represents the white color.

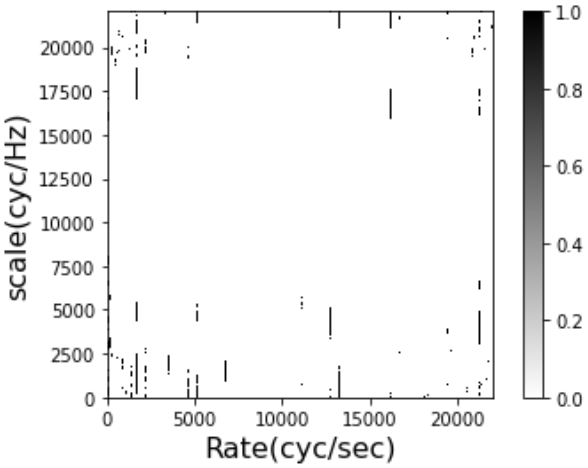


Figure 7-4: Scale Rate Mask for Instrumental

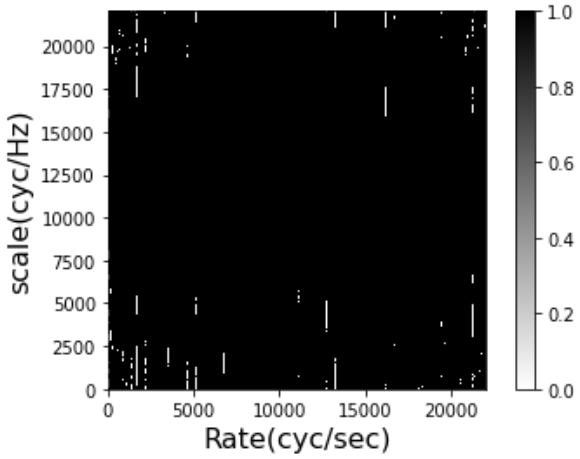


Figure 7-5: Scale Rate Mask for Vocal

7.1.4 Masked Scale Rate Graphs

The scale rate graph of figure 7-3 is element-wise multiplied with the scale rate masks of instrumental and vocal (figure 7-4 and 7-5), the results obtained are shown below:

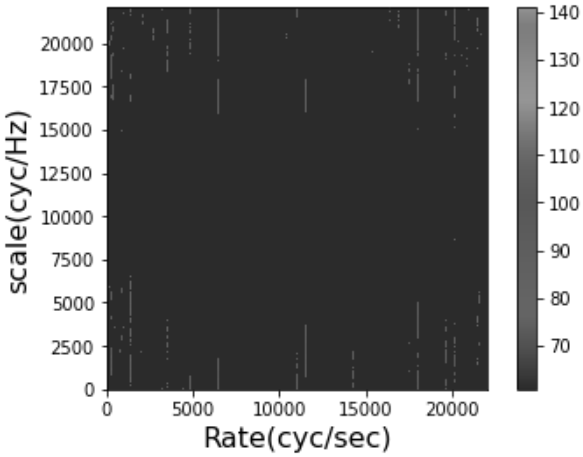


Figure 7-6: Masked Scale Rate Graph of Instrumental

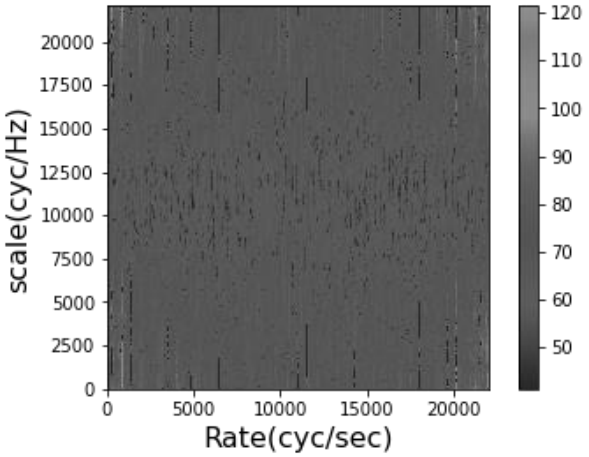


Figure 7-7: Masked Scale Rate Graph of Vocal

7.1.5 Spectrogram Mask

The inverse-2DFT of the masked scale rate graphs are taken, which generates the separate magnitude spectrogram of instrumental and vocal. These separated spectrograms of instrumental and vocal are shown in figures 7-8 and 7-9 respectively.

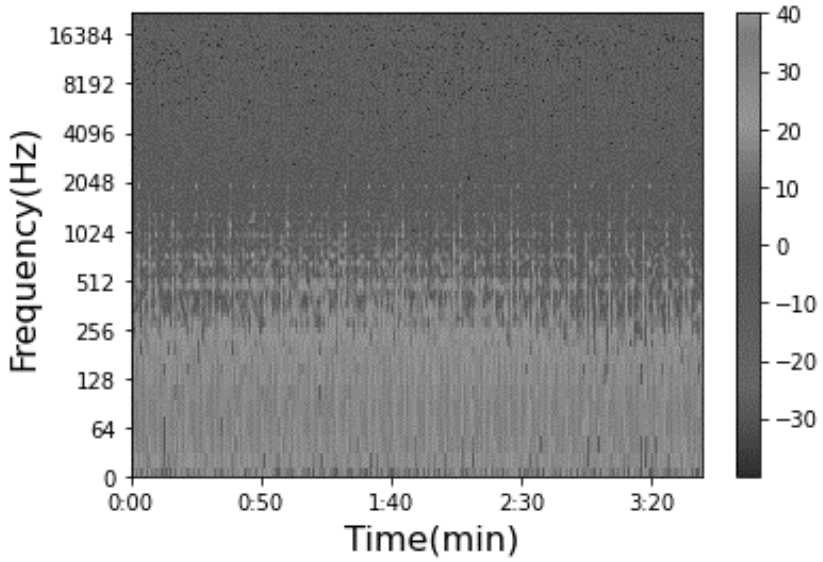


Figure 7-8: Inverse-2DFT of Masked Instrumental Scale Rate Graph

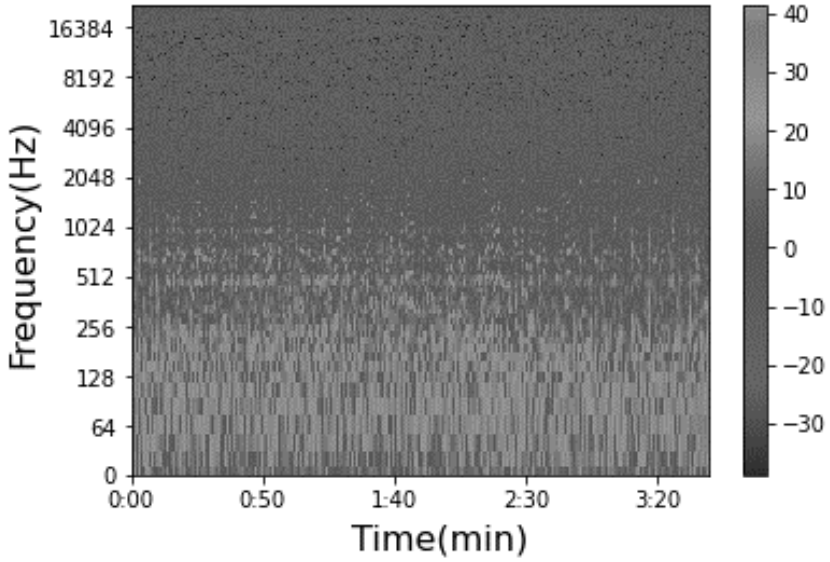


Figure 7-9: Inverse-2DFT of Masked Vocal Scale Rate Graph

A binary instrumental mask and vocal mask are generated, which are used in masking the spectrograms formed by taking inverse-2DFT of the scale rate graphs. The masks

for instrumental spectrogram and vocal spectrograms are shown in figures 7-10 and 7-11.

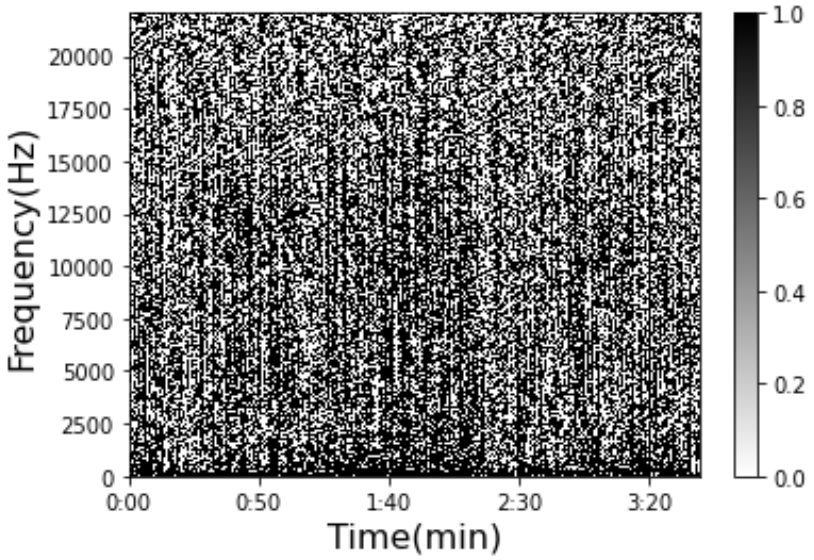


Figure 7-10: Mask for Instrumental Spectrogram

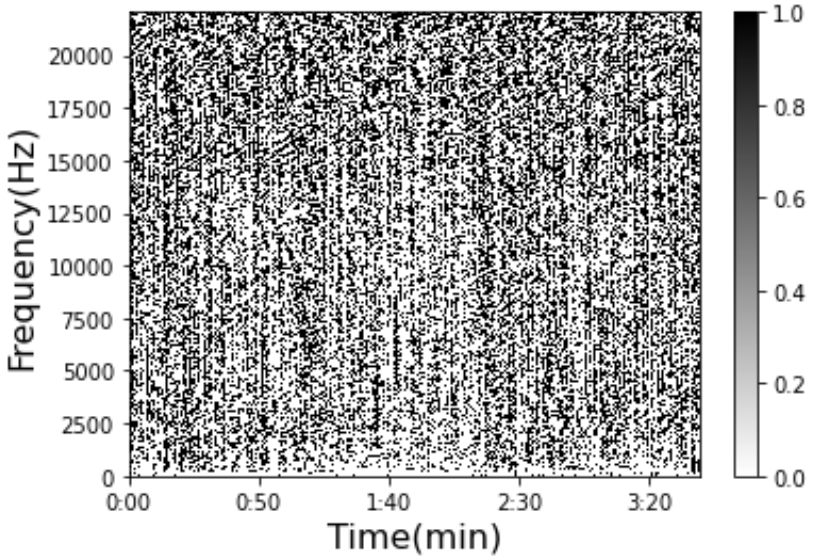


Figure 7-11: Mask for Vocal Spectrogram

The masks for vocal and instrumental are applied with the spectrogram of the song (figure 7-2), to get the separate stems. Figures 7-12 and 7-13 show the extracted instrumental and vocal spectrogram respectively.

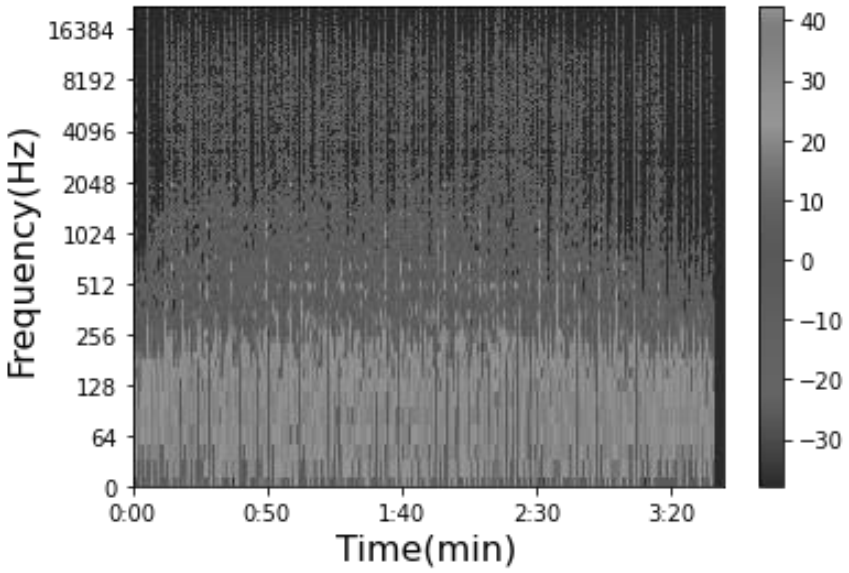


Figure 7-12: Extracted Instrumental Spectrogram

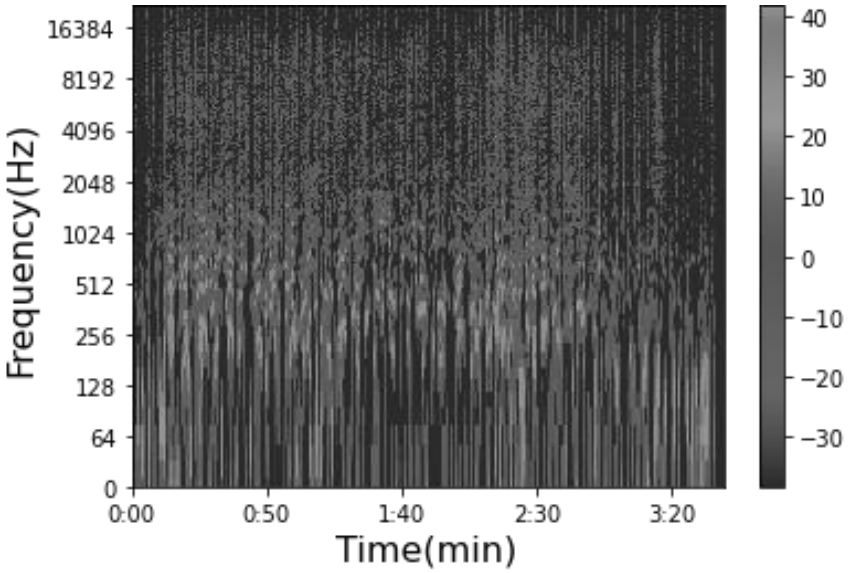


Figure 7-13: Extracted Vocal Spectrogram

7.1.6 Separated Vocal and Instrumental Signals

The inverse-STFT of the respective masked spectrograms of instrumental and vocal are taken, which provides the separate time-domain signal of vocal and instrumental.

The used dataset also provides the separate vocal and instrumental of the mixture song. The figures shown below are the original and extracted waveform of separated vocal and instrumental of the song “Arise - Run Run Run” from the Reggae genre.

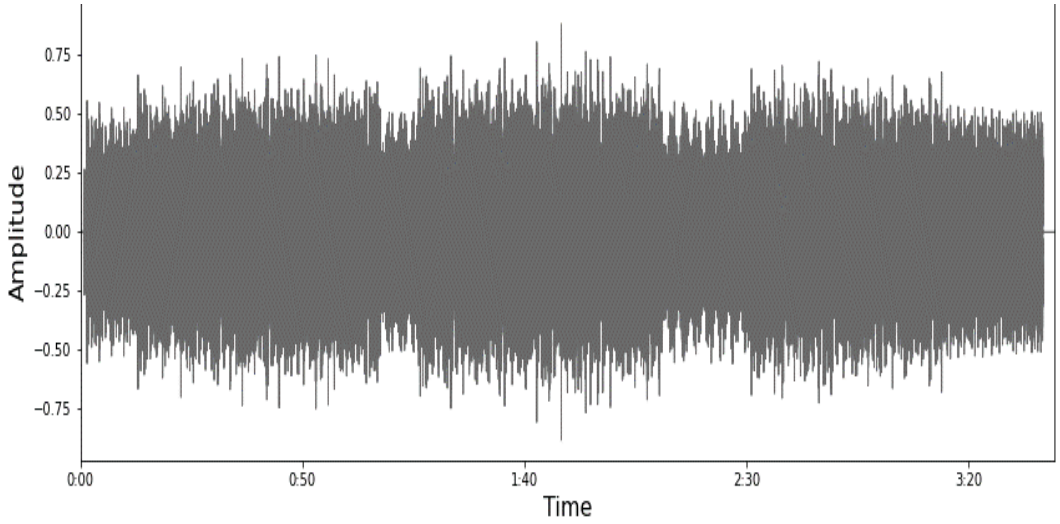


Figure 7-14: Separated Instrumental Waveform of “Run Run Run” Song

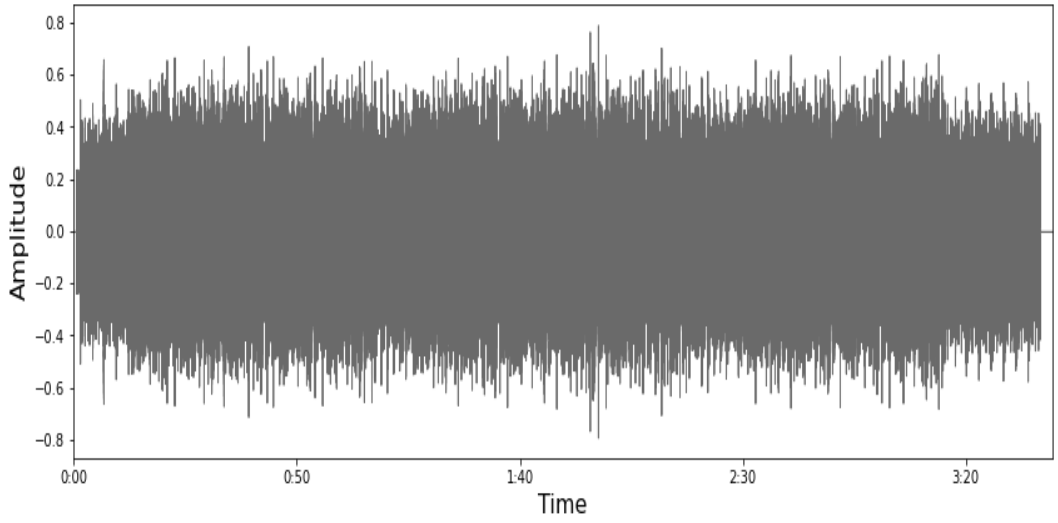


Figure 7-15: Targeted Instrumental Waveform of “Run Run Run” Song

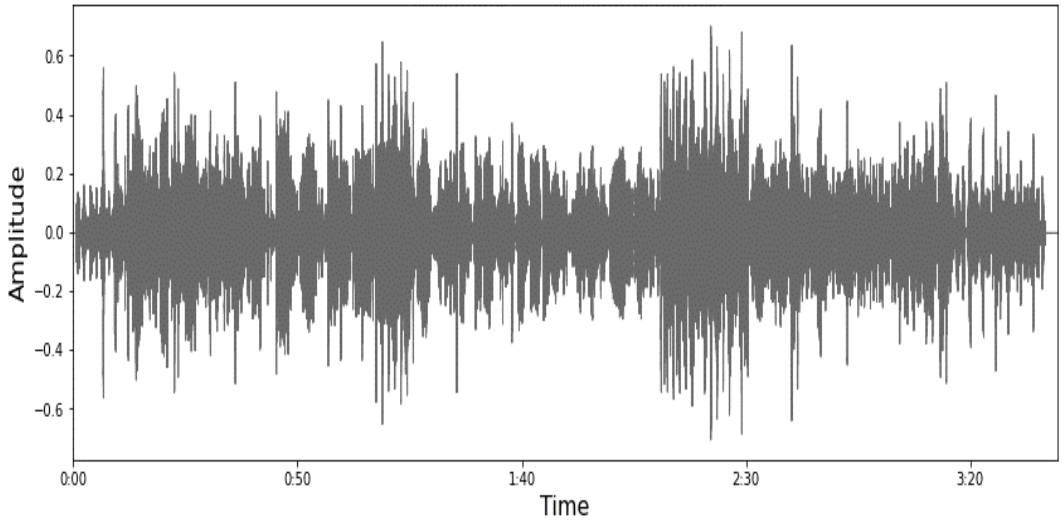


Figure 7-16: Separated Vocal Waveform of “Run Run Run” Song

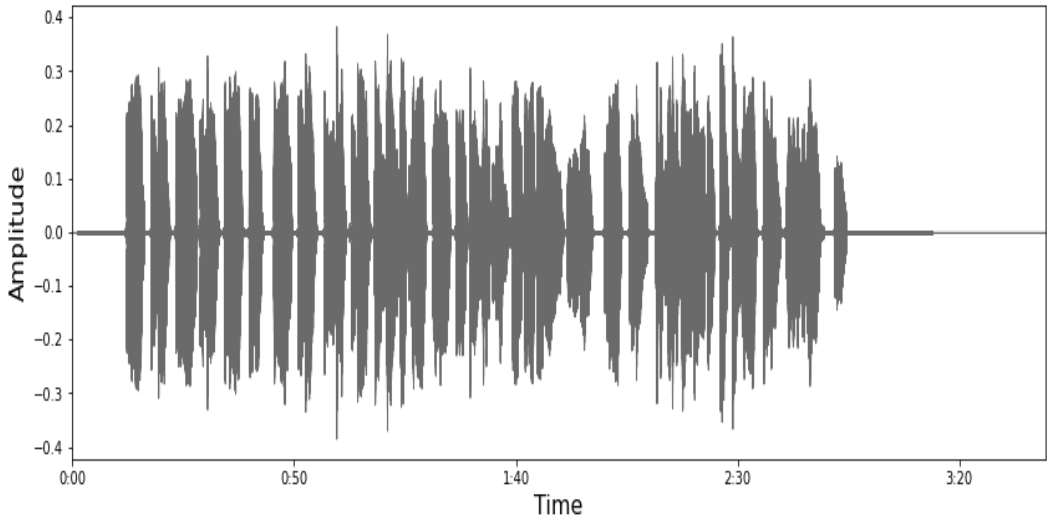


Figure 7-17: Targeted Vocal Waveform of “Run Run Run” Song

7.2 Result from Machine Learning Approach

For explaining the result from the machine learning approach, a song “Left Behind by Hollow Ground” which is of the heavy metal genre is chosen. The waveform of the song is shown in figure 7-18.

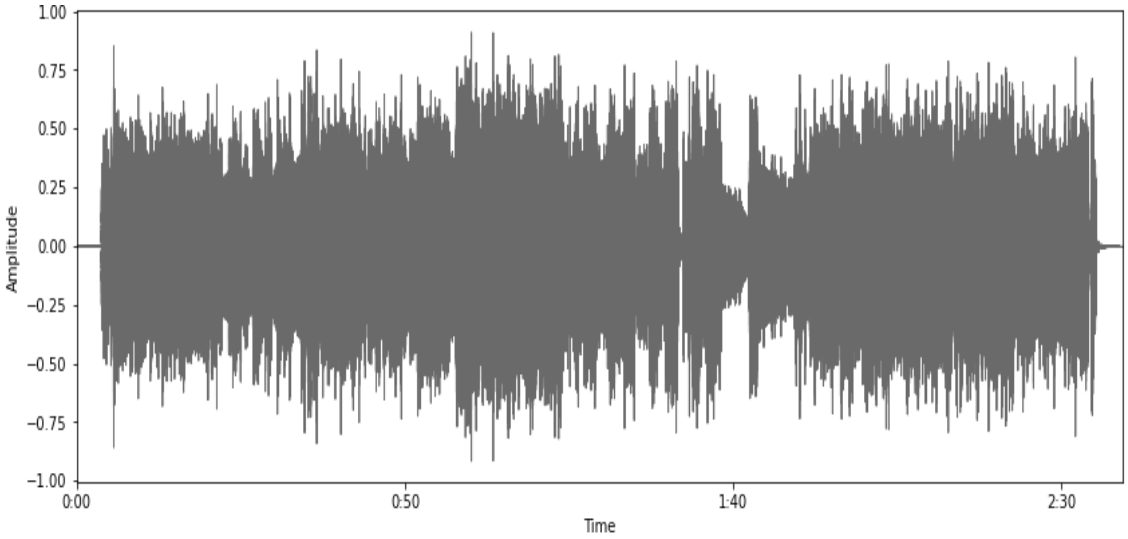


Figure 7-18: Waveform of “Left Behind” Song

7.2.1 Formation of Spectrogram

The spectrogram of the song is generated using the STFT and the spectrogram of the song is shown in figure 7-19.

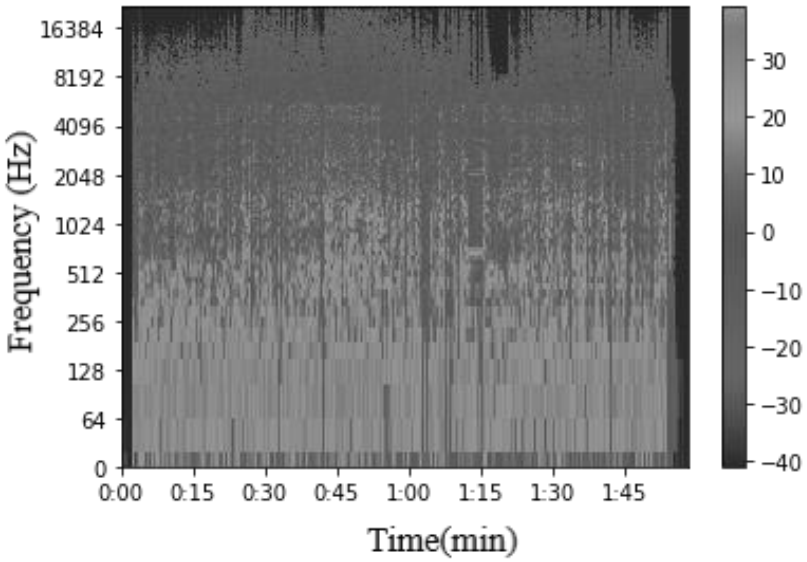


Figure 7-19: Spectrogram of “Left Behind” Song

7.2.2 Spectrograms of Isolated Stems

The trained model predicts the spectrograms of the stems which can be vocal, instrumental, drum, or bass. The predicted spectrograms are shown in Figures 7-20, 7-21, 7-22, and 7-23.

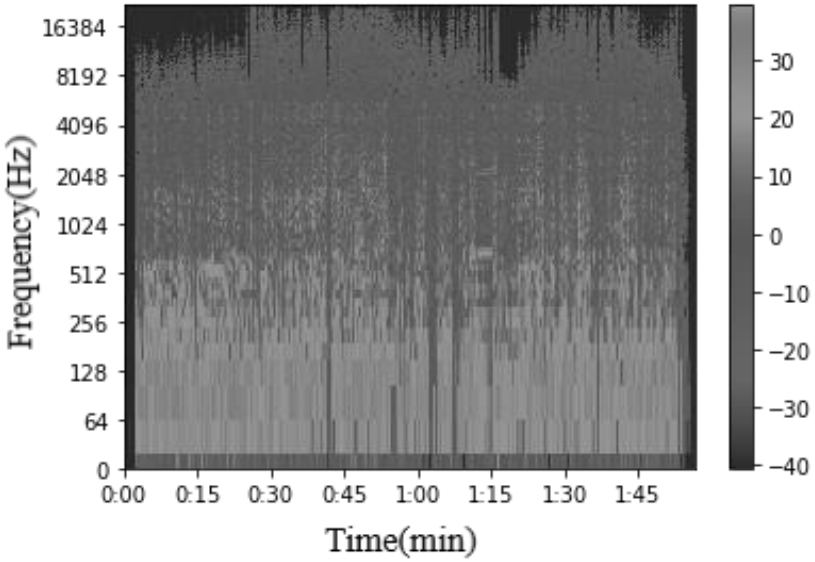


Figure 7-20: Extracted Instrumental Spectrogram

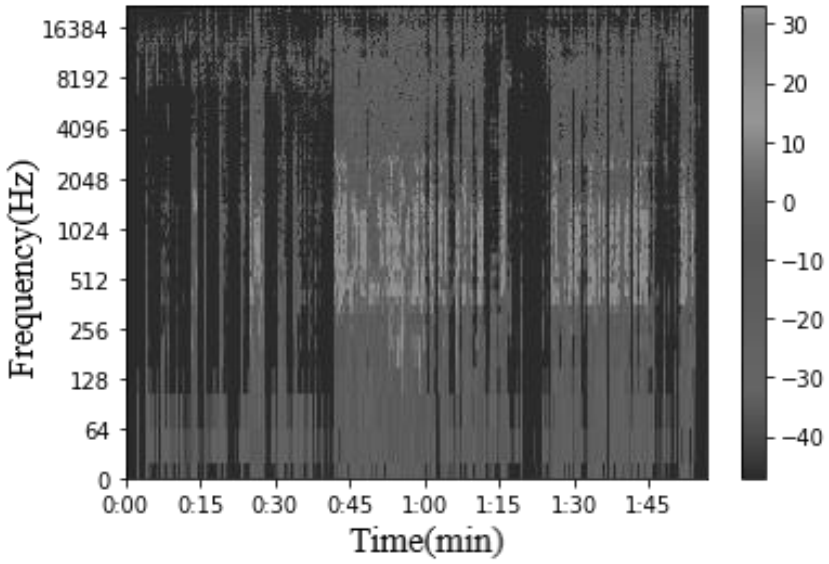


Figure 7-21: Extracted Vocal Spectrogram

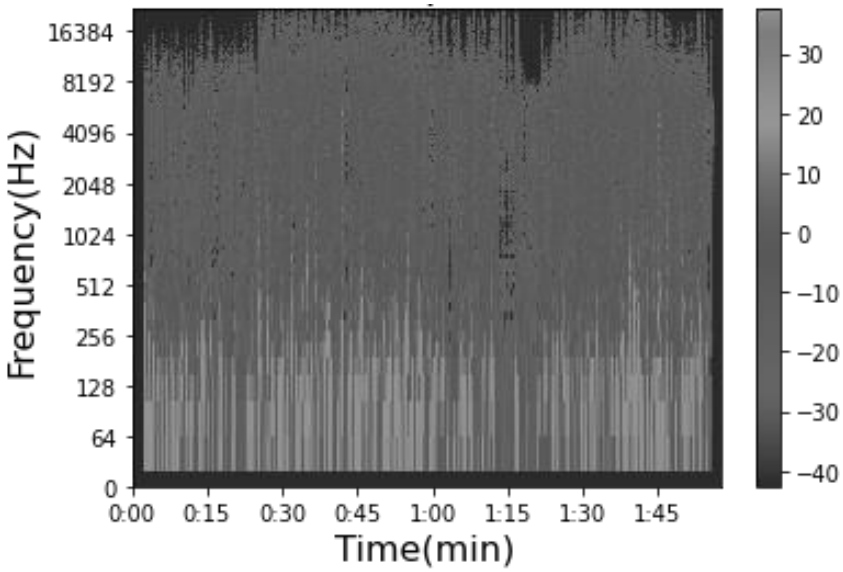


Figure 7-22: Extracted Drum Spectrogram

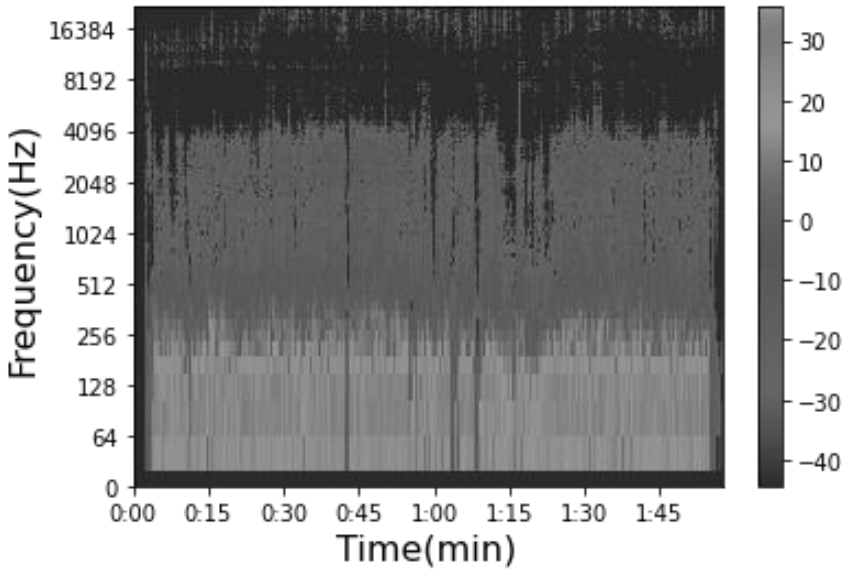


Figure 7-23: Extracted Bass Spectrogram

7.2.3 Separated Stems

The predicted spectrograms are converted back to the time domain signal by taking their Inverse-Short Time Fourier Transform. This process gives the final output where the instrumentals, vocals, drum, and bass are separated. All the signals are in .wav format. The waveform of the separated stems and their respective original stems are shown in the following figures.

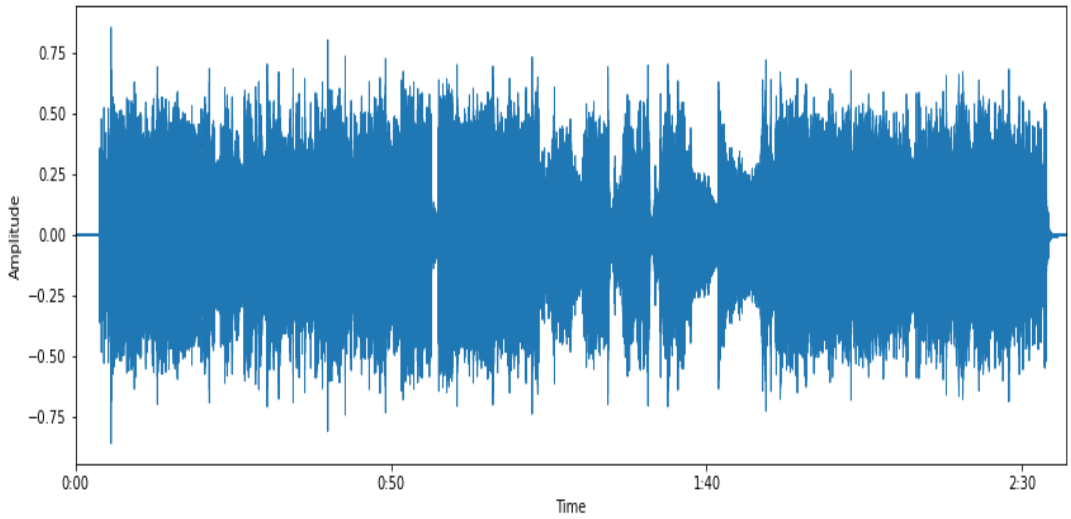


Figure 7-24: Separated Instrumental Waveform of “Left Behind” Song

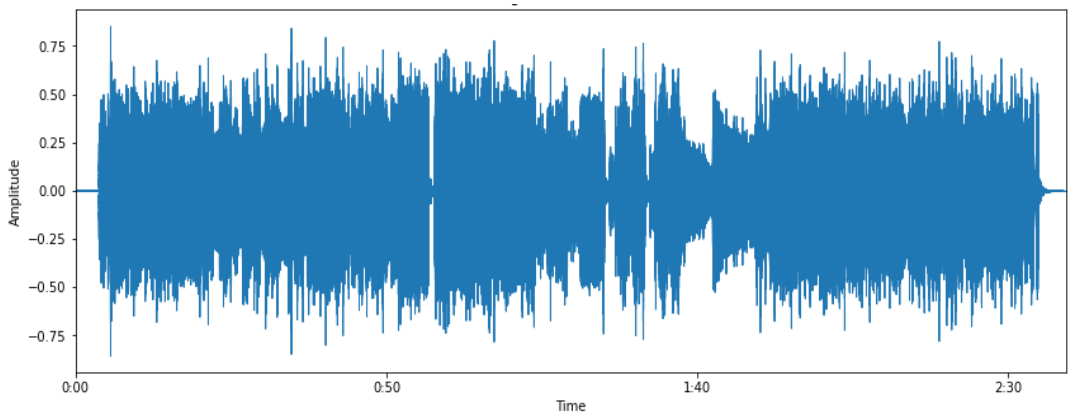


Figure 7-25: Targeted Instrumental Waveform of “Left Behind” Song

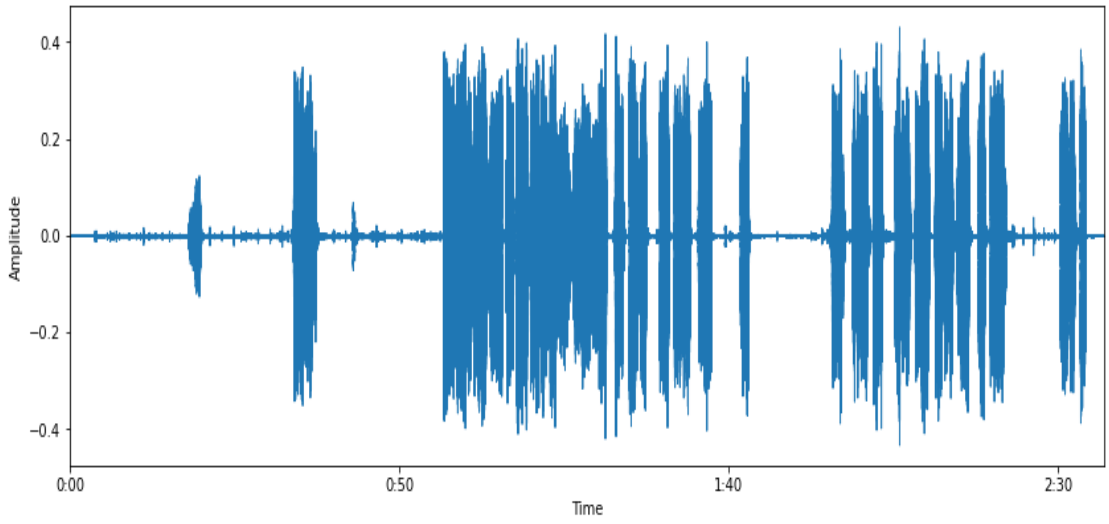


Figure 7-26: Separated Vocal Waveform of “Left Behind” Song

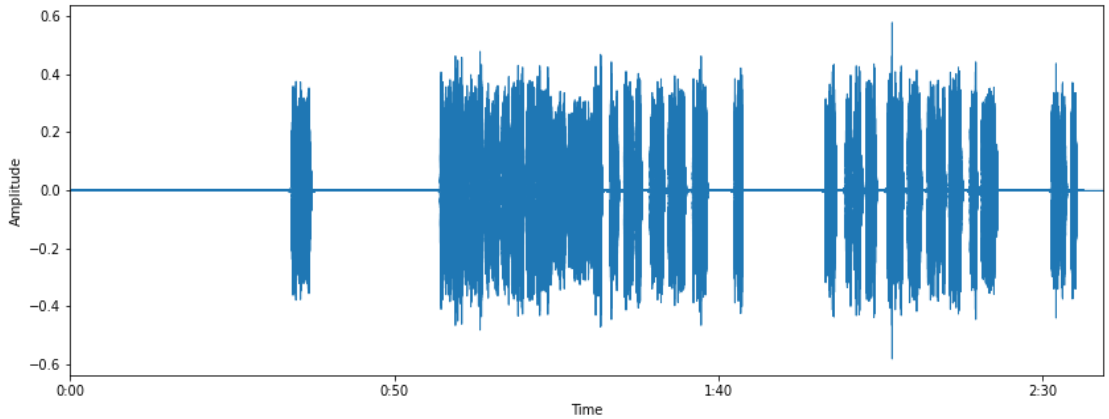


Figure 7-27: Targeted Vocal Waveform of “Left Behind” Song

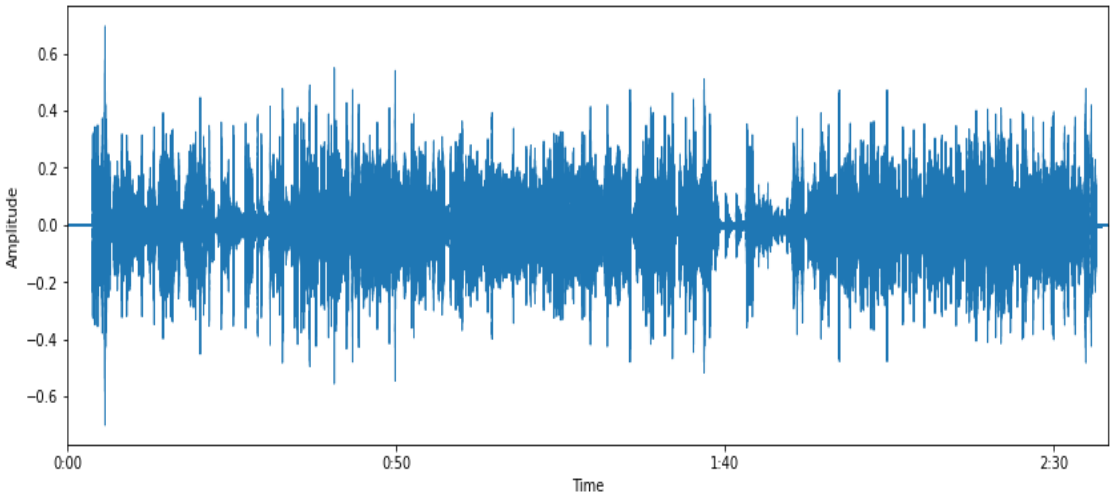


Figure 7-28: Separated Drum Waveform of “Left Behind” Song

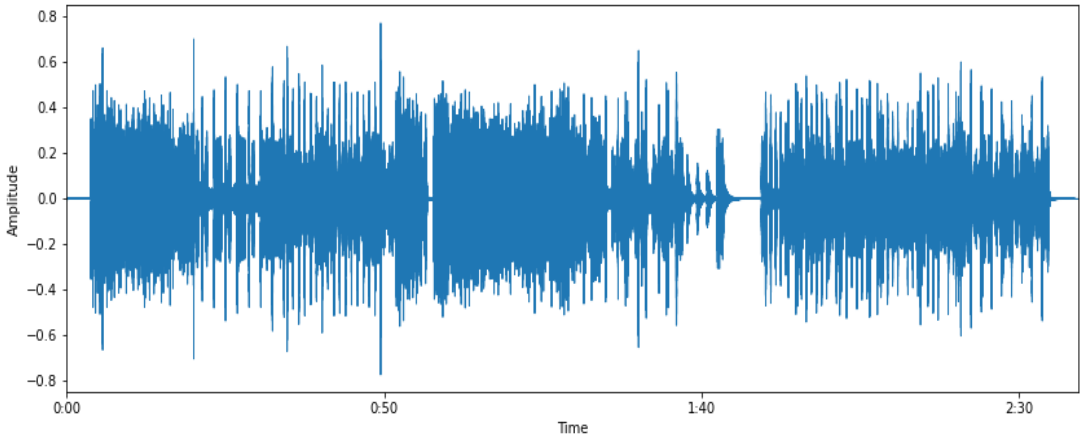


Figure 7-29: Targeted Drum Signal of “Left Behind” Song

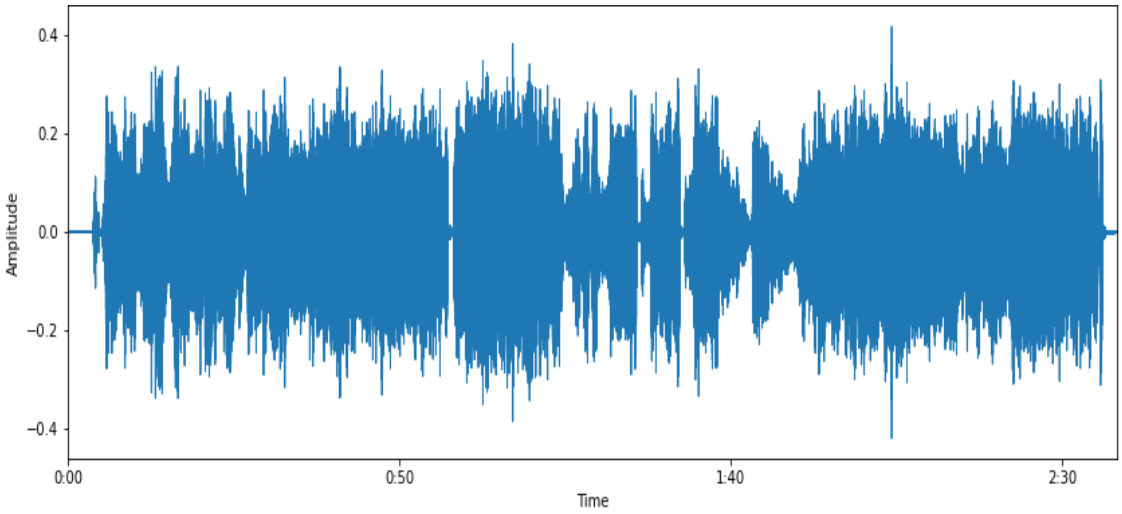


Figure 7-30: Separated Bass Signal of “Left Behind” Song

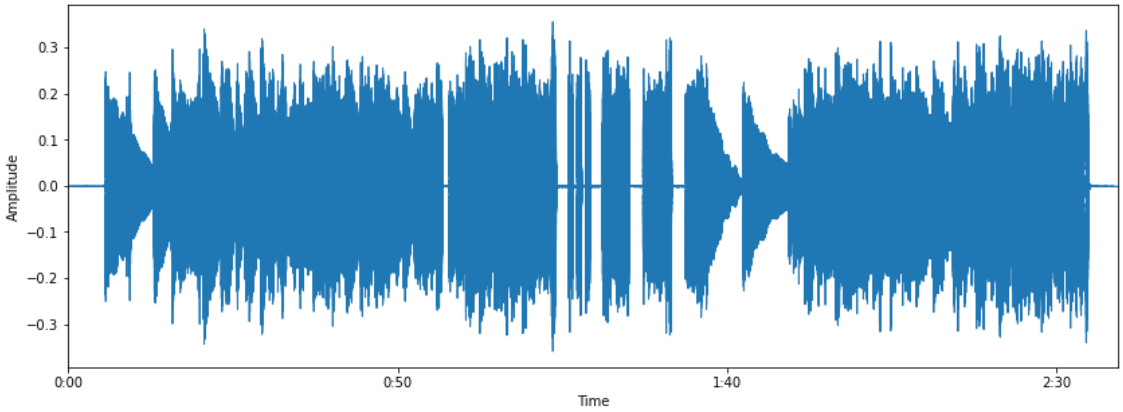


Figure 7-31: Targeted Bass Signal of "Left Behind" Song

7.3 Song Samples for Further Analysis

For further analysis of the results, one song from each genre in the dataset was chosen. The title of the song along with its artist is mentioned in table 7-1.

Table 7-1: Song Samples from Dataset

S.N.	Genre	Artist	Song Name
1	Pop/Rock	AM Contra	Heart Peripheral
2	Country	Angela Thomas Wade	Milk Cow Blues
3	Rap	ANiMAL	Rockshow
4	Jazz	Patrick Talbot	A Reason to Leave
5	Reggae	Arise	Run Run Run
6	Electronic	Giselle	Moss
7	Heavy Metal	Hollow Ground	Left Blind

7.4 Evaluation of Result of 2DFT Approach

The evaluation metrics are calculated for both the separated instrumental and vocal signals of all the chosen songs from each genre. Also, these metrics are calculated for the results obtained by changing the window function while taking the STFT.

7.4.1 SNR of Extracted Output Signals

For the instrumental music signal, the residue of the vocal is noise whereas for vocal it is vice versa. The residue is obtained from the difference between the original and extracted signal. The SNR of instrumental and vocal for a song from each genre is shown in table 7-2.

Table 7-2: SNR of 2DFT Approach Output Signals

S. N	Genre	SNR of Instrumental (dB)	SNR of Vocal (dB)
1	Pop/Rock	6.8457	2.1018
2	Country	5.5380	0.9763
3	Rap	4.6192	-2.4698
4	Jazz	4.4232	-0.1313
5	Reggae	10.3305	2.4286
6	Electronic	3.6487	-9.3916
7	Heavy Metal	7.4202	-0.7201

The negative values in the table signify that the output signal has more noise than the required signal i.e., the ratio of signal to noise is less than 1. The higher the SNR of the output signal, the more accurate it is to the required original signal.

7.4.2 Cosine Similarity of Output Signals

The dataset provides separate instrument music and vocal of the songs. For the calculation of the cosine similarity, the output instrumental and vocal of the song is related to the provided separated instrumental music and vocal.

Table 7-3: Cosine Similarity of 2DFT Approach Output Signals

S.N	Genre	Cosine Similarity of Instrumental	Cosine Similarity of Vocal
1	Pop/Rock	0.8728	0.6786
2	Country	0.8062	0.6629
3	Rap	0.7743	0.4968
4	Jazz	0.7416	0.6023
5	Reggae	0.9440	0.6544
6	Electronic	0.7237	0.2512
7	Heavy Metal	0.8900	0.5293

In table 7-3, the values close to 1 signify that the extracted output and the original signals are similar and as the value tends to 0, the extracted output and original signals are different from each other.

7.4.3 SDR of Extracted Output Signals

SDR is a measurement to look at the degradation of the output signal by unwanted signals - in particular distortion. In this project SDR measures the amount of distortion measured in the output signal from the original signals. The SDR of instrumental and vocal for songs from the song samples are shown in table 7-4.

Table 7-4: SDR of 2DFT Approach Output Signals

S.N	Genre	SDR of Instrumental (dB)	SDR of Vocals (dB)
1	Pop/Rock	6.1578	0.8527
2	Country	4.5305	0.2435
3	Rap	3.8704	-3.211
4	Jazz	3.4528	-0.6493
5	Reggae	9.4276	1.1246
6	Electronic	3.0422	-10.0340
7	Heavy Metal	6.6965	-1.4521

The larger values of the SDR in table 7-4 indicates that the separated output have less distortion and the output waveform is closer to the original waveforms. While the lower values signify that the separated output is distorted.

7.4.4 SAR of Extracted Output Signals

The artifacts are the error which are generated during the conversion processes. These errors are produced when the signal is converted into spectrograms and changing to scale rate graphs.

Also, artifacts are produced while taking the inverse-2DFT and inverse-STFT. The SAR of instrumental and vocal for a song from each genre is calculated in table 7-5.

Table 7-5: SAR of 2DFT Approach Output Signals

S.N	Genre	SAR of Instrumental (dB)	SAR of Vocals (dB)
1	Pop/Rock	5.2404	-0.6405
2	Country	3.5657	-0.9569
3	Rap	2.1449	-4.7934
4	Jazz	1.4462	-2.2790
5	Reggae	9.1015	-0.8136
6	Electronic	0.8474	-11.6256
7	Heavy Metal	6.1805	-4.0395

7.4.5 SNR vs. Rate

The rate values can be changed from 15 to 100 frames. With the change in rate, the SNR of instrumental music and vocal changes. For both signals, the SNR decreases as the rate increases.

Taking a higher value of rate creates a higher range which corresponds to an increase in the neighboring size. Hence, SNR decreases as the rate increases. The relationship is plotted between SNR vs Rate for Reggae and Electronic genre song as they gave the best and worst SNR respectively.

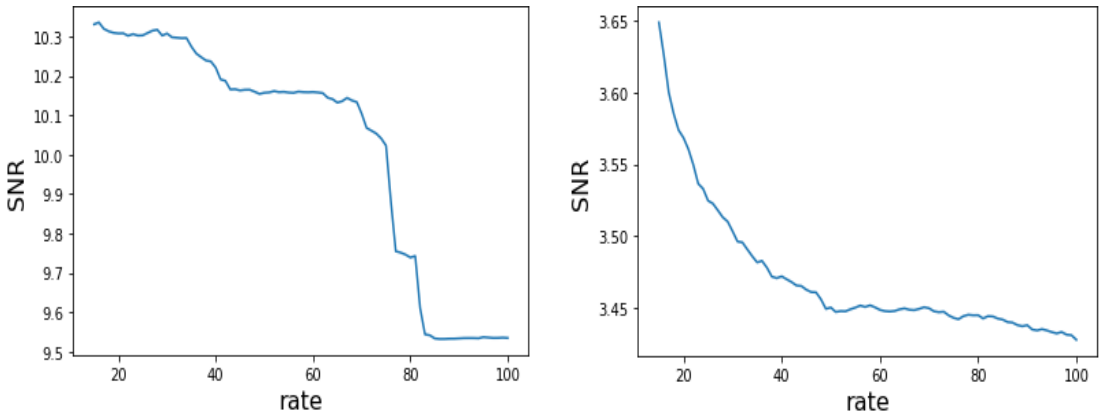


Figure 7-32: SNR vs. Rate of Instrumental Signal

In figure 7-32, the relationship is plotted between SNR vs. Rate for the instrumental music of Reggae and Electronic genre song respectively.

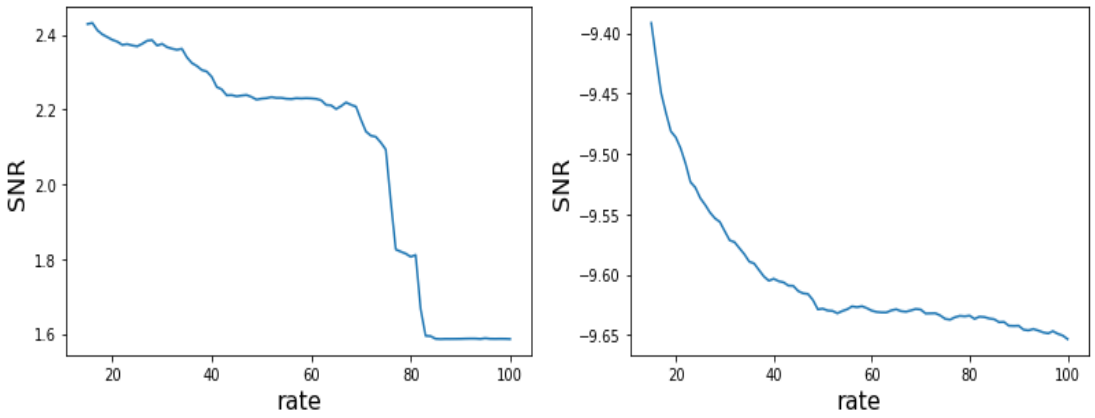


Figure 7-33: SNR vs. Rate of Vocal Signal

In figure 7-33, the relationship is plotted between SNR vs. Rate for output vocal signal of Reggae and Electronic genre song respectively.

7.4.6 Cosine Similarity vs. Rate

The graph of cosine Similarity vs. rate was plotted. Here as well, the graphs were plotted for Reggae and Electronic genre, as they gave the best and worst SNR respectively.

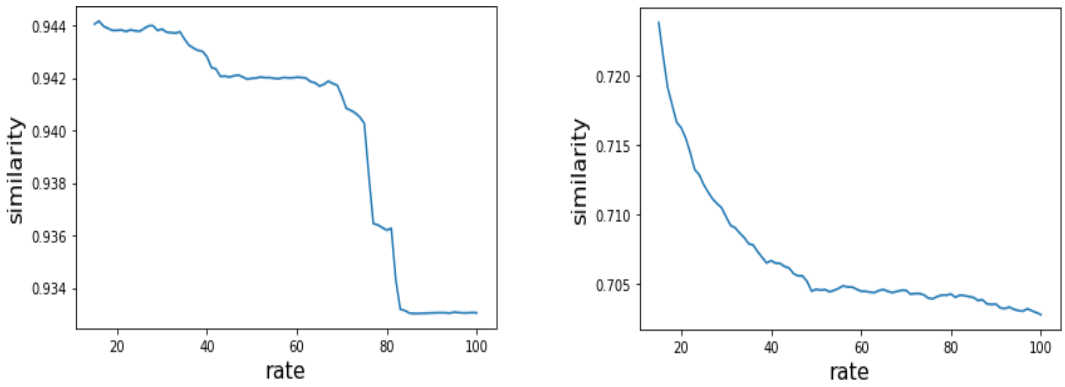


Figure 7-34: Similarity vs. Rate of Instrumental Signal

In figure 7-34, the relationship is plotted between Similarity vs Rate for output instrumental music signal of Reggae and Electronic genre song respectively.

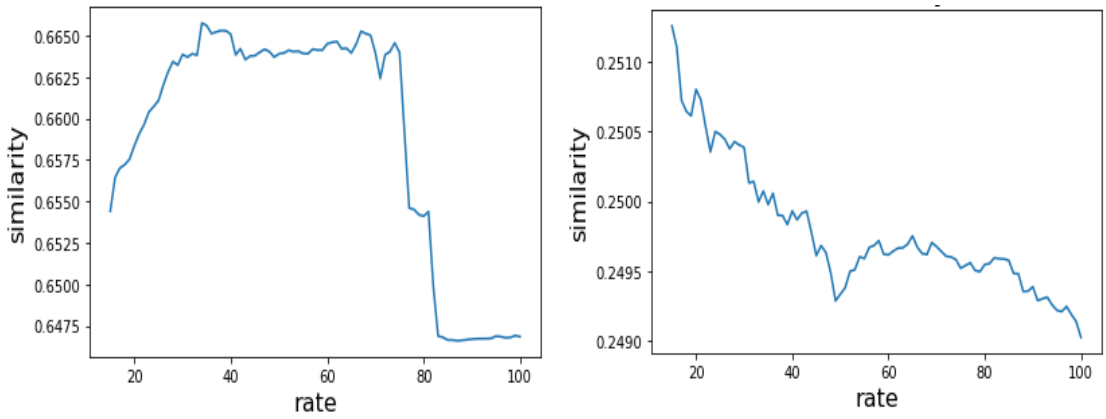


Figure 7-35: Similarity vs. Rate of Vocal Signal

In figure 7-35, the relationship is plotted between Similarity vs. Rate for output instrumental music signal of Reggae and Electronic genre song respectively. Here for the reggae signal the similarity increases as the rate increases towards 40 and starts to decrease as the rate reaches 75.

7.4.7 SDR vs. Rate

Here the SDR vs the rate graphs were plotted. The graphs which were plotted were of the reggae and the electronic genre song. These two genres' songs were chosen since they gave the best and the worst SDR respectively.

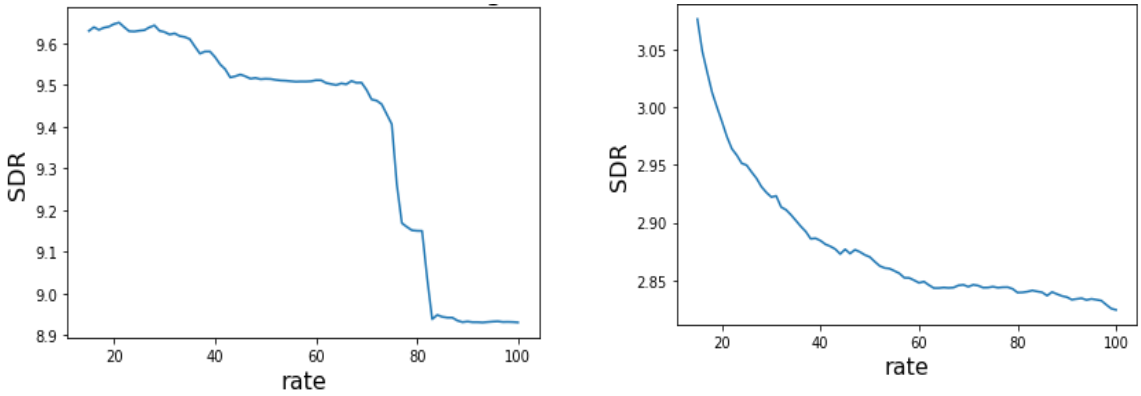


Figure 7-36: SDR vs. Rate of Instrumental Signal

In figure 7-36, the relationship is plotted between SDR vs. Rate for output instrumental music signal of Reggae and Electronic genre song respectively.

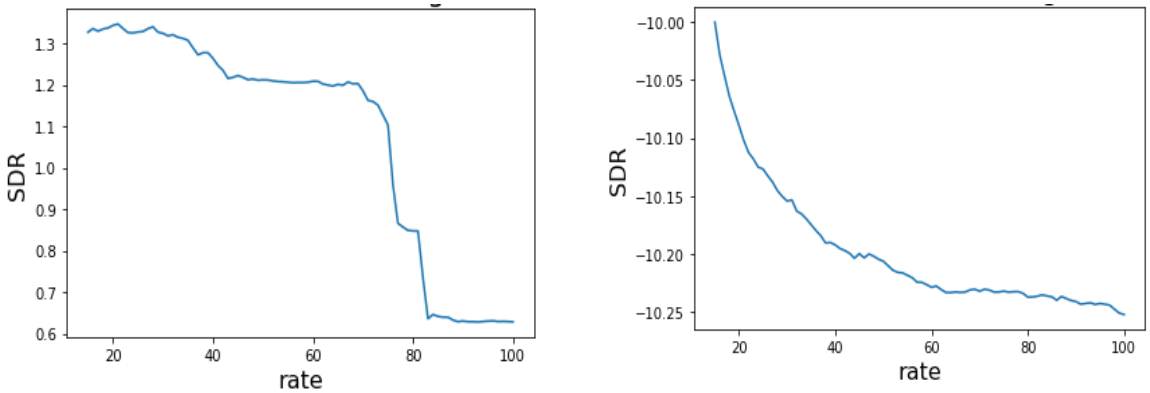


Figure 7-37: SDR vs. Rate for Vocal Signal

In figure 7-37, the relationship is plotted between SDR vs. Rate for output instrumental music signal of Reggae and Electronic genre song respectively. For the reggae signals, the distortions gradually increase whereas the distortion increases sharply as the rate is increased in electronic genre output signals.

7.4.8 SAR vs. Rate

The SAR of the output signals were plotted by varying the rate between 15 to 100. The graphs which were plotted are too of the reggae and electronic genre songs.

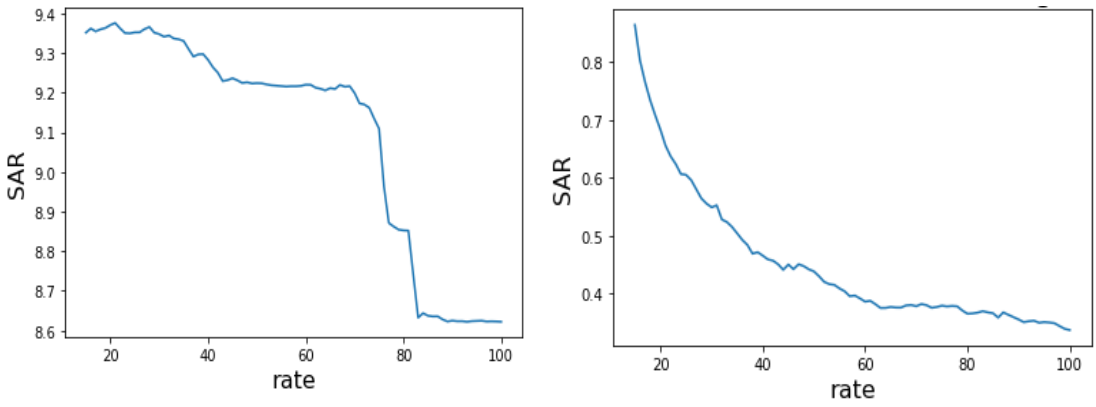


Figure 7-38: SAR vs. Rate for Instrumental Signal

In figure 7-38, the relationship is plotted between SAR vs. Rate for output instrumental music signal of Reggae and Electronic genre song respectively.

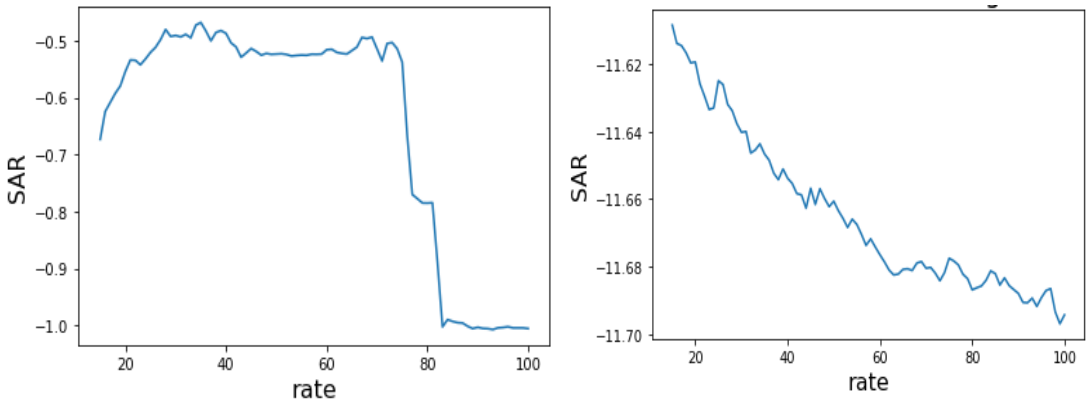


Figure 7-39: SAR vs. Rate for Vocal Signal

In figure 7-39, the relationship is plotted between SAR vs. Rate for output vocal signal of Reggae and Electronic genre song respectively. Here for the reggae signal the artifact decreases as the rate increases towards 40 and starts to decrease as the rate reaches 75.

7.4.9 Variation in Window Function

The above results were obtained by using the Hanning window while taking STFT of the different genre songs. Changing the window functions gave different results which are tabulated below.

7.4.9.1 Outputs using Sqrt-Hanning Window

While using the Square Hanning window, the best instrumental and vocals were obtained for reggae genre song and the worst for electronic genre song as mentioned in table 7-6.

Table 7-6: SNR using Sqrt-Hanning Window

S.N	Genre	SNR of Instrumental (dB)	SNR of vocals (dB)
1	Pop/Rock	6.6004	1.8140
2	Country	5.4170	0.8293
3	Rap	4.5812	-2.4902
4	Jazz	4.3355	-0.2362
5	Reggae	10.1694	2.2107
6	Electronic	3.5665	-9.5145
7	Heavy Metal	7.0955	-1.0904

Similarly, to SNR, the best cosine similarity of instrumental and vocals were obtained for reggae genre song and the worst for electronic genre song as shown in table 7-7.

Table 7-7: Cosine Similarity using Sqrt-Hanning Window

S.N	Genre	Cosine Similarity of instrumental	Cosine Similarity of Vocals
1	Pop/Rock	0.8644	0.6720
2	Country	0.8003	0.6627
3	Rap	0.7743	0.4958
4	Jazz	0.7352	0.6045
5	Reggae	0.9416	0.6543
6	Electronic	0.7137	0.2512
7	Heavy Metal	0.8799	0.5135

Using the Sqrt-Hanning window, the least number of distortions for both instrumental and vocal was observed for Reggae signal while the greatest number of distortions was detected for Electronic genre song as seen in table 7-8.

Table 7-8: SDR using Sqrt-Hanning Window

S.N	Genre	SDR of Instrumental (dB)	SDR of Vocals (dB)
1	Pop/Rock	5.9181	0.6132
2	Country	4.4416	0.1546
3	Rap	3.7647	-3.3170
4	Jazz	3.3836	-0.7188
5	Reggae	9.6352	1.3323
6	Electronic	2.9615	-10.1146
7	Heavy Metal	6.3811	-1.7672

Similarly, to SDR, the best SAR of instrumental and vocals were obtained for reggae genre song and the worst for electronic genre song as shown in table 7-9.

Table 7-9: SAR using Sqrt-Hanning Window

S.N	Genre	SAR of Instrumental (dB)	SAR of Vocals (dB)
1	Pop/Rock	4.927	-0.8148
2	Country	3.4886	-0.9817
3	Rap	1.9544	-4.8991
4	Jazz	1.3233	-2.2768
5	Reggae	9.3543	-0.6187
6	Electronic	0.6683	-11.6279
7	Heavy Metal	5.7344	-4.3510

7.4.9.2 Outputs using Blackman Window

Among the three windows used, the best instrumental and vocal signal was observed while using the Blackman Window.

Table 7-10: SNR using Blackman Window

S. N	Genre	SNR of Instrumental (dB)	SNR of Vocals (dB)
1	Pop/Rock	6.9352	2.2183
2	Country	5.6809	1.1511
3	Rap	4.6743	-2.4263
4	Jazz	4.4518	-0.1097
5	Reggae	10.3603	2.4434
6	Electronic	3.7196	-9.2965
7	Heavy Metal	7.5652	-0.5566

While using the Blackman Window, the best cosine similarity of instrumental and vocals were obtained for reggae genre song and the worst for electronic genre song as shown in table 7-10.

Table 7-11: Cosine Similarity using Blackman Window

S.N	Genre	Cosine Similarity of Instrumental	Cosine Similarity of Vocal
1	Pop/Rock	0.8753	0.6768
2	Country	0.8128	0.6629
3	Rap	0.7758	0.4967
4	Jazz	0.7426	0.5966
5	Reggae	0.9442	0.6464
6	Electronic	0.7307	0.2507
7	Heavy Metal	0.8937	0.5348

While using the Blackman window, the observed output signal which had the least amount of distortion was the reggae genre song and the output signal which had the most amount of distortion was of the electronic genre as shown in table 7-12.

Table 7-12: SDR using Blackman Window

S.N	Genre	SDR of Instrumental (dB)	SDR of Vocals (dB)
1	Pop/Rock	5.7088	0.9041
2	Country	4.4064	1.0263
3	Rap	3.8983	-3.1833
4	Jazz	3.4320	-0.6701
5	Reggae	9.6294	1.3265
6	Electronic	3.0762	-10.0001
7	Heavy Metal	6.8187	-1.3298

Similarly, to SDR, the least amount of loss in the signal for instrumental and vocals were obtained for reggae genre song and the worst for electronic genre song as shown in table 7-13.

Table 7-13: SAR using Blackman Window

S.N	Genre	SAR of Instrumental (dB)	SAR of Vocals (dB)
1	Pop/Rock	4.6341	-0.6358
2	Country	3.5780	-0.9576
3	Rap	2.1582	-4.7605
4	Jazz	1.3981	-2.3466
5	Reggae	9.3515	-0.673
6	Electronic	0.8646	-11.6084
7	Heavy Metal	6.3278	-3.9302

7.5 Evaluation of Result of Machine Learning Approach

Similar to the calculation of the evaluation metrics from the 2DFT approach, the same evaluation metrics are calculated for all vocal, instrumental, bass, and drum stems separated from the machine learning approach, for the chosen songs from each genre.

7.5.1 SNR of Extracted Output Signals

The noise is the unwanted signals contained in the output. For the instrumentals, the residue of vocals is noise and vice versa. For the drum and bass, the signal other than itself is the noise. The SNR of instrumental, vocal, drum and bass for the selected songs is shown in table 7-14.

Table 7-14: SNR of Output Signals from Machine Learning Approach

Genre	SNR of Vocal (dB)	SNR of Instrumental (dB)	SNR of Drum (dB)	SNR of Bass (dB)
Pop/Rock	6.3181	9.1852	4.2249	-3.2061
Country	5.1654	8.6397	3.0561	2.0732
Rap	3.3399	9.7534	2.3053	3.9658
Reggae	4.1843	8.6592	-2.5902	1.2126
Electronic	5.3954	8.6700	2.1252	4.0147
Heavy Metal	4.4968	9.2185	0.4338	2.7370
Jazz	7.6181	7.3812	6.0837	3.5682

The higher the SNR of the resulting signal, the more precise it is to the expected signal.

The negative qualities imply that the resulting signal has more commotion than the expected signal.

7.5.2 Cosine Similarity of Output Signals

The output waveform should be similar to the separated waveform provided from the dataset. The similarity is calculated by using cosine similarity, which provides a real number between ‘0’ and ‘1’.

Table 7-15: Cosine Similarity of Output Signals from Machine Learning Approach

Genre	Cosine Similarity of Vocal	Cosine Similarity of Instrumental	Cosine Similarity of Drum	Cosine Similarity of Bass
Pop/Rock	0.8843	0.9420	0.7929	0.4160
Country	0.8572	0.9299	0.7801	0.6976
Rap	0.7581	0.9459	0.7198	0.7878
Reggae	0.7885	0.9358	0.5439	0.6268
Electronic	0.8651	0.9303	0.7150	0.7914
Heavy Metal	0.8032	0.9432	0.4495	0.7338
Jazz	0.9096	0.9046	0.8926	0.7667

In table 7-16, the values close to 1 signify that the extracted output and the original signals are similar and as the value tends to 0, the extracted output and original signals are different from each other.

7.5.3 SDR of Extracted Output Signals

The SDR of all the stems for a song from each genre is shown in table 7-17.

Table 7-16: SDR of Output Signals from Machine Learning Approach

Genre	SDR of Vocal (dB)	SDR of Instrumental (dB)	SDR of Drum (dB)	SDR of Bass (dB)
Pop/Rock	6.3181	9.1852	4.2249	-3.2061
Country	5.1654	8.6397	3.0561	2.0731
Rap	3.3399	9.7534	2.3053	3.9658
Reggae	4.1843	8.6592	-2.5902	1.2126
Electronic	5.3954	8.6700	2.1252	4.0147
Heavy Metal	4.4968	9.2185	0.4338	2.7370
Jazz	7.6181	7.3812	6.0837	3.5682

The larger values of the SDR in table 7-16 indicate that the separated output has less distortion and the output waveform is closer to the original waveforms. The lower values signify that the separated output is distorted.

7.5.4 SAR of Extracted Output Signals

The SAR of all the stems, for songs from each genre, is shown in table 7-17.

Table 7-17: SAR of Output Signals from Machine Learning Approach

Genre	SAR of Vocal (dB)	SAR of Instrumental (dB)	SAR of Drum (dB)	SAR of Bass (dB)
Pop/Rock	7.7059	9.1473	2.7190	-5.8124
Country	4.8321	9.3898	2.2697	0.0581
Rap	2.4673	9.9783	0.8206	2.7393
Reggae	3.0958	8.5258	-3.5596	-1.6525
Electronic	4.8879	9.7357	1.7054	2.3588
Heavy Metal	3.5516	9.2330	-5.5563	1.8676
Jazz	7.8850	7.1152	6.8360	2.3651

7.5.5 Epoch vs Mean Squared Error

Epoch indicates the number of passes of the entire training dataset that the machine learning algorithm has completed. Generally, when the epochs are increased, the model loss decreases. It can be seen in the figures below as well. In the epoch vs error plot below, peaks were seen in the validation curves, where the validation loss increases and decreases sharply. This is because the validation set changes at certain intervals due to the use of cross-validation in the training phase.

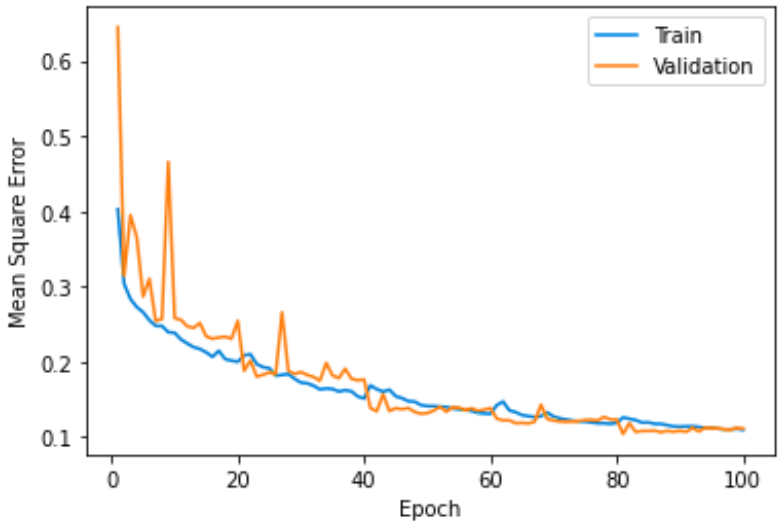


Figure 7-40: Epochs vs Mean Squared Error for Instrumentals

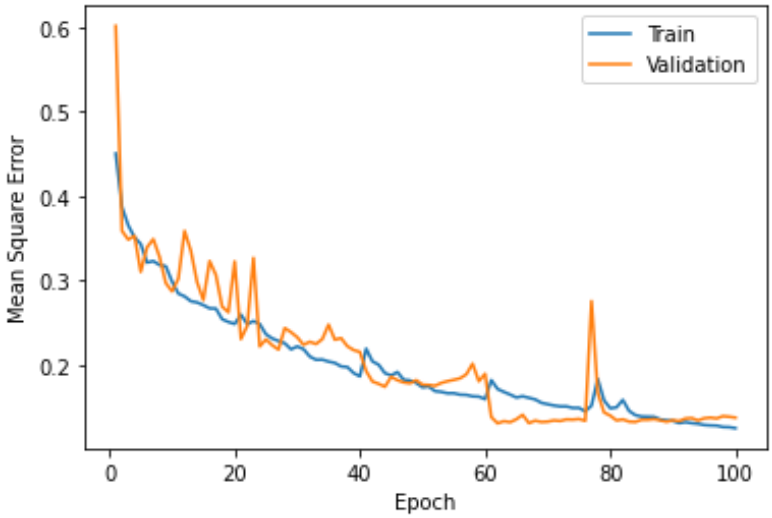


Figure 7-41: Epochs vs Mean Squared Error for Vocals

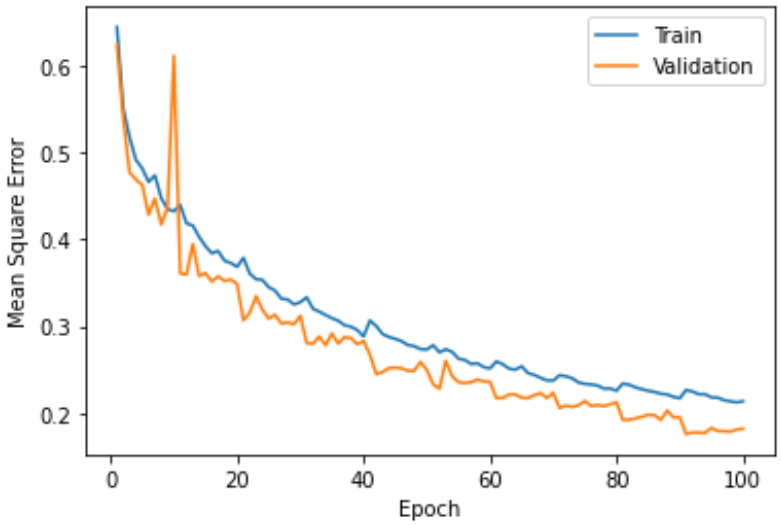


Figure 7-42: Epochs vs Mean Squared Error for Drum

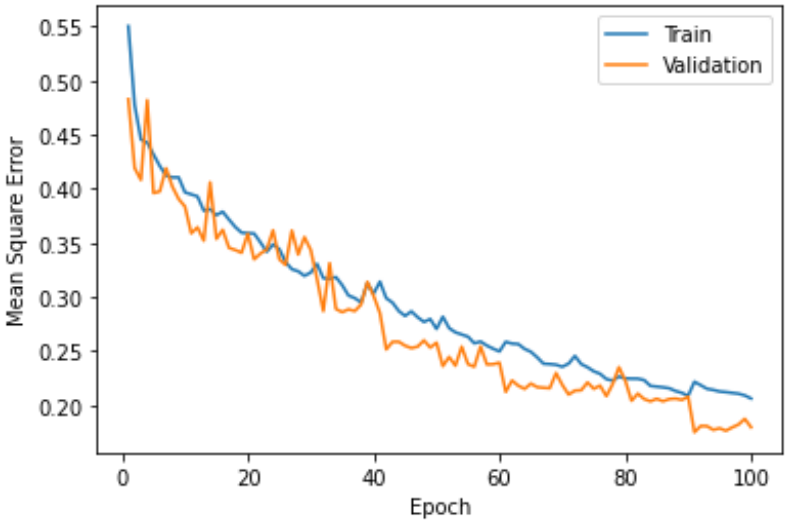


Figure 7-43: Epochs vs Mean Squared Error for Bass

7.6 Comparison of Output waveform

The output waveform from both of the approaches were compared with a same song, “Run Run Run by Arise” of Reggae genre. It was observed that the machine learning approach produced better result than the 2DFT approach. This observation was seen for both instrumentals and vocals.

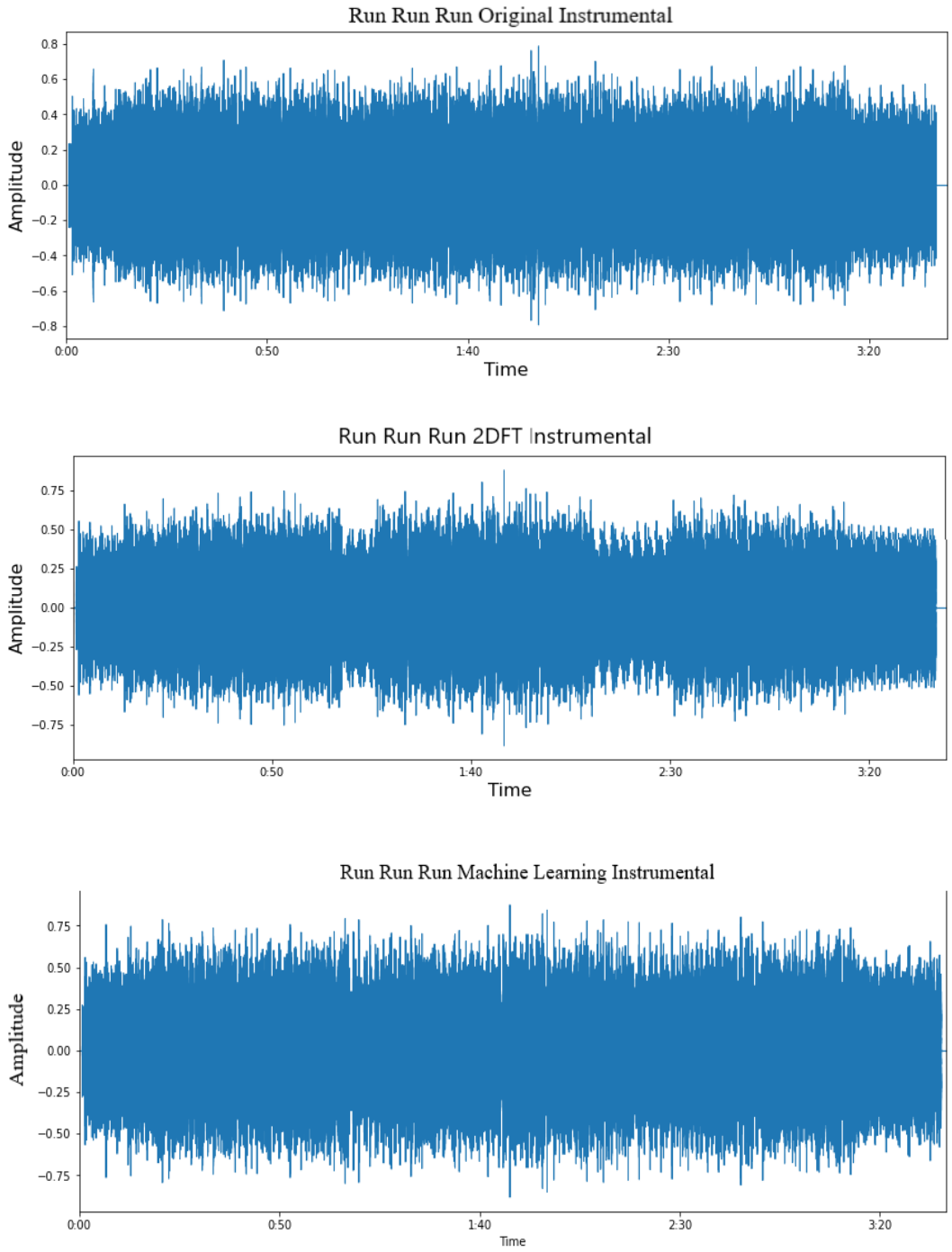


Figure 7-44: Comparison of Instrumental Waveforms

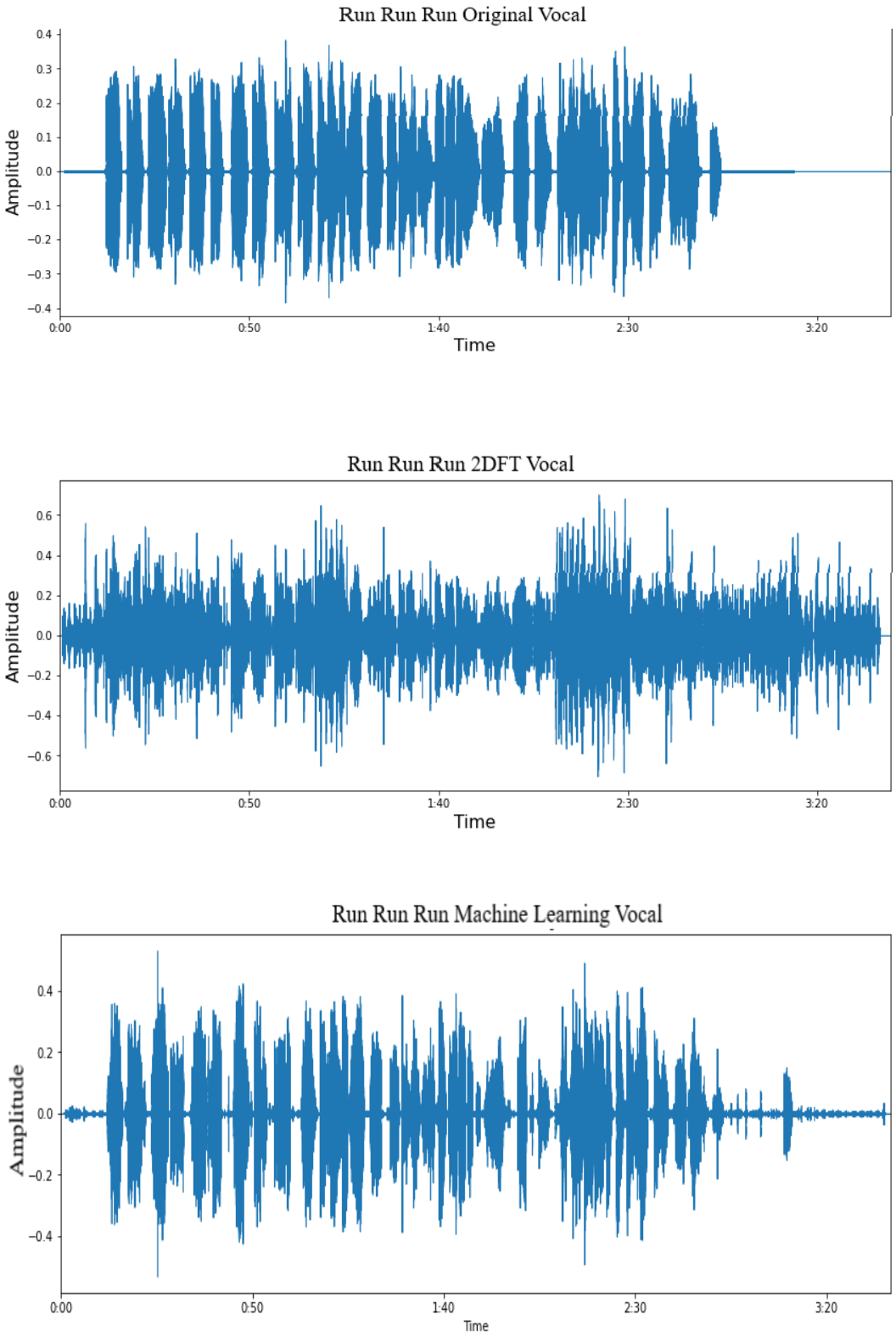


Figure 7-45: Comparison of Vocal Waveforms

7.7 Comparison with Other Related Projects

For the comparison with other related projects, only the machine learning approach from the project is used, as the machine learning approach works better than the 2DFT approach. The SDR metrics is used for the comparison purpose. It is because most of the papers use SDR metrics to compare different source separation approaches.

The following is the table of some of the similar projects for the separation of instrumentals and vocals.

Table 7-18: SDR Values of Other Similar Projects for Vocal and Instrumental [26]

Model	Vocal	Instrumental	Overall
Dedicated U-Nets (x2)	5.09	12.95	9.02
C-U-Net	4.42	12.21	8.31
UW	5.06	12.98	9.02
DWA	5.20	12.96	9.08
EBW P1	5.12	13.06	9.09
EBW InstP1	5.28	13.04	9.16
EBW P2	5.07	12.89	8.98

The following is the table of some of the similar projects for the separation of bass, drums and vocal.

Table 7-19: SDR Values of Other Similar Projects for Vocal, Drum, and Bass [27]

Model	Vocal	Drum	Bass	Overall
Wave-U-Net	3.25	4.22	3.21	3.56
Open-Unmix	5.33	5.73	5.23	5.43
Meta-Tasnet	6.40	5.91	5.58	5.96
D3Net	7.24	7.01	5.25	6.50
Demucs	6.84	6.86	7.01	6.90

The following is the table of the SDR values of the project for the vocal and instrumental, evaluated in the test dataset.

Table 7-20: SDR Values of the Project for Vocal and Instrumental

Vocal	Instrumental	Overall
5.28	11.43	8.355

Table 7-21: SDR Values of the Project for Vocal, Drum, and Bass

Vocal	Drum	Bass	Overall
5.28	4.03	3.21	3.56

In comparing to other similar projects, the results for the vocals and instrumentals are similar, but the result for the drum and bass stems are lacking. In general, the SDR value of the instrumental is high compare to that of vocal, drum, and bass. This is because the instrumental of the song is almost continuous throughout the song but vocal, drum and, bass are not. So, when there is no presence of such stems, and the model tries to predict stems in such instance, the predicted result is distorted, which in turn distorts the overall signal and reduces the SDR value.

8. FUTURE ENHANCEMENT

There is always room for improvement in every project and this project is not an exception. There can be improvements in some areas of our project. One such area is the dataset used. The dataset used in the project consists of only bass, drum, vocal, and instrumental stems, which limits the separation to only four stems. In the future, if a better dataset with more instruments could be used, a greater number of stems could be separated. The current dataset consists of only English songs; thus, it might not work best for songs in another language, thus the dataset in the local Nepali language can be created and used. Besides the dataset, improvements can be made in the model as well. Apart from the model used in this project, there are many models which can be explored.

In this project, we explored 2DFT method and machine learning method with U-net architecture. The 2DFT approach is used to find repetition in the scale-rate domain, but the machine learning approach works in the frequency domain i.e., spectrograms of the songs is used as input. In the future, the scale-rate graphs can be used as input to the machine learning method which might give better output for the vocal and instrumental signals. Besides this, we used STFT to convert time domain signal to frequency domain, however other methods like Constant Q-Transform can be used for this purpose.

9. CONCLUSION

The project was completed by using two approaches, the 2DFT approach and the machine learning approach. Initially, the 2DFT approach was carried out. It provided the instrumentals and vocals from a song. For the separation of bass and drum stems and to get better results in separating vocals and instrumentals, the machine learning approach was attempted.

During the 2DFT approach, the song was first converted into the spectrogram. The spectrogram was then converted to a scale-rate graph by taking 2DFT of the spectrogram. Masks were generated in the scale rate domain to separate the instrumental and vocal scale rate graph. The masked scale rate graphs were changed back to spectrograms via Inverse-2DFT. These results were compared to create masks for the extraction of vocals and instrumental spectrograms. The masks were then applied and finally, Inverse-STFT was applied to get separated waveforms.

After the 2DFT approach was completed, the machine learning approach was carried out. In the machine learning approach, a U-Net model was trained by using the spectrograms of the songs. The dataset used for training was the MUSDB-18 dataset which contains 150 different songs along with its separated vocals, drum, and bass.

Similarly, to the 2DFT approach, in the machine learning approach as well, the songs were converted to the spectrogram by taking the STFT. The obtained spectrograms were divided into two sets of data, training dataset and validation dataset. The training to validation dataset ratio was 4:1. After the training and validation of the model were completed, the trained model was tested. It resulted in the mask for the vocals, instrumentals, drum, and bass stems. The masks generated were multiplied with the original spectrogram of the song. Finally, Inverse-STFT was applied to separated spectrograms to get individual waveforms in .wav format.

Thus, the 2DFT approach fulfils the first objective of the project, which is to separate vocals and instrumental from a song and the machine learning approach fulfils both of the objective of the project, which is to separate vocal, instrumental, bass and drum stems from a song.

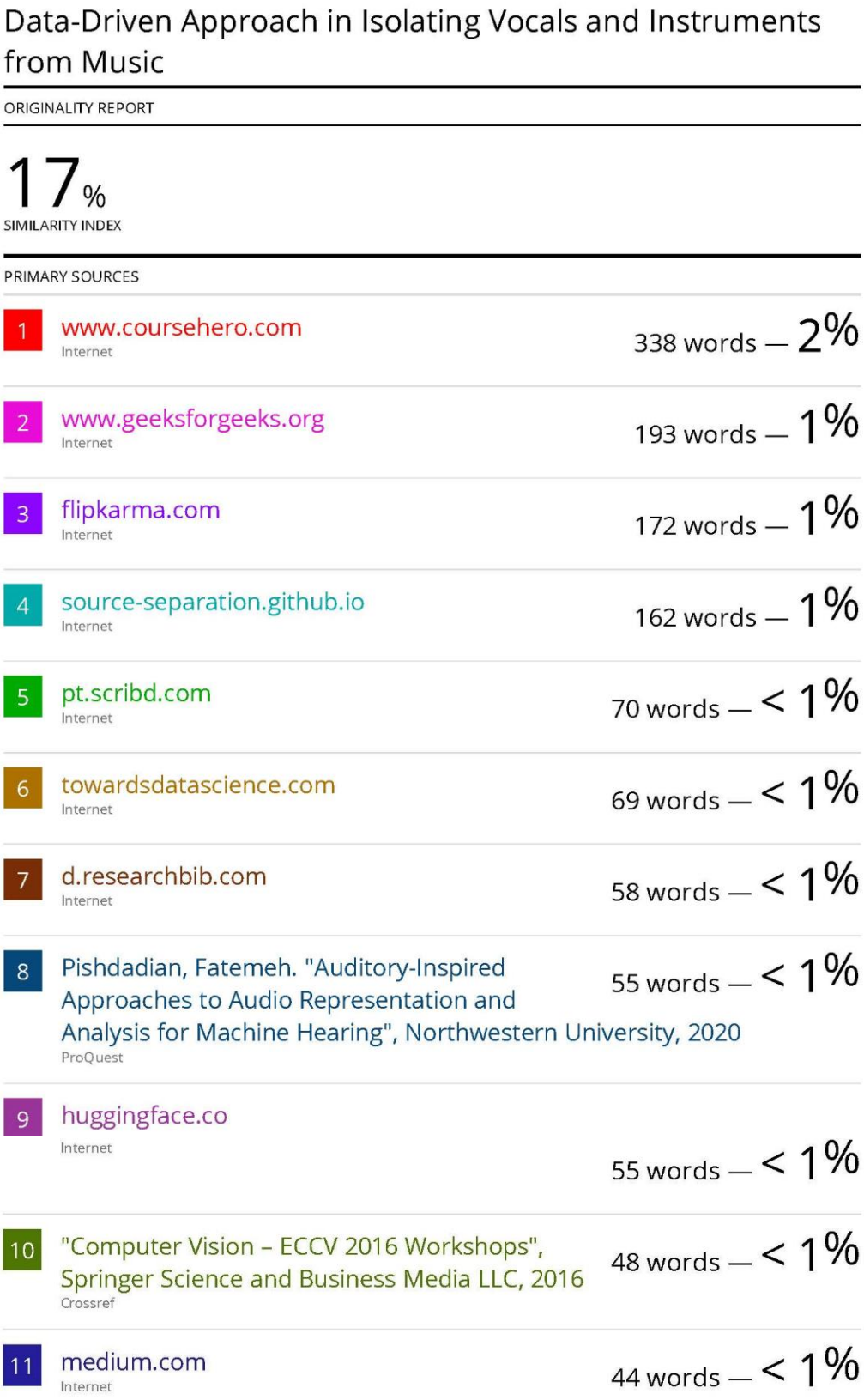
10. APPENDICES

Appendix A: Project Schedule

Table A: Gantt Chart with Project Activities and Timeline

Project Start Date	07,May,21											
Task	Progess (%)	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb
Brain Storming and Topic Discussion	100										
Documentation	100										
Machine Learning Basic course	100										
Machine Learning Advance Course	100										
Audio Dataset collection	100										
Separation with 2DFT Approach	100										
Separation with Machine Learning Approach	100										

Appendix B: Similarity Index of Report



Data-Driven Approach in Isolating Vocals and Instruments from Music

ORIGINALITY REPORT

17%
SIMILARITY INDEX

PRIMARY SOURCES

1	www.coursehero.com Internet	338 words — 2%
2	www.geeksforgeeks.org Internet	193 words — 1%
3	flipkarma.com Internet	172 words — 1%
4	source-separation.github.io Internet	162 words — 1%
5	pt.scribd.com Internet	70 words — < 1%
6	towardsdatascience.com Internet	69 words — < 1%
7	d.researchbib.com Internet	58 words — < 1%
8	Pishdadian, Fatemeh. "Auditory-Inspired Approaches to Audio Representation and Analysis for Machine Hearing", Northwestern University, 2020 ProQuest	55 words — < 1%
9	huggingface.co Internet	55 words — < 1%
10	"Computer Vision – ECCV 2016 Workshops", Springer Science and Business Media LLC, 2016 Crossref	48 words — < 1%
11	medium.com Internet	44 words — < 1%

12	arxiv.org Internet	34 words — < 1%
13	eprints.utem.edu.my Internet	32 words — < 1%
14	mind.cp.eng.chula.ac.th Internet	29 words — < 1%
15	Rudranil Das, Deepti Deshwal, Pardeep Sangwan, Neelam Nehra. "Music Source Separation: A Guide", 2021 International Conference on Industrial Electronics Research and Applications (ICIERA), 2021 Crossref	27 words — < 1%
16	tudr.thapar.edu:8080 Internet	27 words — < 1%
17	G. Albertengo, W. Hassan. "SHORT TERM URBAN TRAFFIC FORECASTING USING DEEP LEARNING", ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2018 Crossref	26 words — < 1%
18	Safayet Anowar Shurid, Khandaker Habibul Amin, Md. Shahnawaz Mirbahar, Dolan Karmaker et al. "Bangla Sign Language Recognition and Sentence Building Using Deep Learning", 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2020 Crossref	26 words — < 1%
19	repository.tudelft.nl Internet	26 words — < 1%
20	assets.kpmg.com Internet	25 words — < 1%
21	ismir2018.ircam.fr Internet	24 words — < 1%
22	Prem Seetharaman, Fatemeh Pishdadian, Bryan Pardo. "Music/Voice separation using the 2D fourier transform", 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017 Crossref	23 words — < 1%
23	apps.dtic.mil Internet	23 words — < 1%

24	Song, Lin. "Multipath Approaches to Avoiding TCP Incast.", The University of Iowa, 2019 ProQuest	20 words — < 1%
25	mafiadoc.com Internet	19 words — < 1%
26	repository.ntu.edu.sg Internet	19 words — < 1%
27	en.wikipedia.org Internet	18 words — < 1%
28	etheses.whiterose.ac.uk Internet	18 words — < 1%
29	rua.ua.es Internet	18 words — < 1%
30	www.amiproject.org Internet	18 words — < 1%
31	www.bridgeinternationalacademies.com Internet	18 words — < 1%
32	www.mdpi.com Internet	18 words — < 1%
33	www.theses.fr Internet	18 words — < 1%
34	Soha Rostaminia, Seyedeh Zohreh Homayounfar, Ali Kiaghadi, Trisha Andrew, Deepak Ganesan. "PhyMask: Robust Sensing of Brain Activity and Physiological Signals During Sleep with an All-textile Eye Mask", ACM Transactions on Computing for Healthcare, 2022 Crossref	17 words — < 1%
35	idr.nitk.ac.in Internet	17 words — < 1%
36	www.nrel.gov Internet	17 words — < 1%
37	Ng, Chong Hwa. "Control of Active Filters to Attenuate Harmonic Resonance in Power Distribution Networks.", University of Northumbria at Newcastle (United Kingdom), 2020 ProQuest	16 words — < 1%
38	portfoliodb.hslu.ch Internet	16 words — < 1%

39	Ribeiro, Alexandrine. "Study of Attention Mechanisms and Ensemble Methods for Medical Image Semantic Segmentation", Universidade do Minho (Portugal), 2021 ProQuest	15 words — < 1%
40	pure.tue.nl Internet	15 words — < 1%
41	Akis Stavropoulos, Kaushik J Lakshminarasimhan, Jean Laurens, Xaq Pitkow, Dora Angelaki. "Influence of sensory modality and control dynamics on human path integration", Cold Spring Harbor Laboratory, 2020 Crossref Posted Content	14 words — < 1%
42	Jan-Willem W. Steeb, David B. Davidson, Stefan J. Wijnholds. "Mitigation of Non-Narrowband Radio Frequency Interference Incorporating Array Imperfections", Journal of Astronomical Instrumentation, 2019 Crossref	14 words — < 1%
43	Samira Mavaddati. "A Novel Singing Voice Separation Method Based on a Learnable Decomposition Technique", Circuits, Systems, and Signal Processing, 2020 Crossref	14 words — < 1%
44	eprints.glos.ac.uk Internet	14 words — < 1%
45	vivelaruta.es Internet	14 words — < 1%
46	www.iau.edu.sa Internet	14 words — < 1%
47	www.section.io Internet	14 words — < 1%
48	Aggarwal, Vishwam. "Quadrotor Design, Modeling and Control.", Arizona State University, 2020 ProQuest	13 words — < 1%
49	Y. Zhou, A. Sheremet, Y. Qin, J.P. Kennedy, N.M. DiCola, A. P. Maurer. "High-order theta harmonics account for the detection of slow gamma", Cold Spring Harbor Laboratory, 2019 Crossref Posted Content	13 words — < 1%
50	palevel.unza.zm Internet	13 words — < 1%

51	Bai, . "Serial Port Programming in LabVIEW", The Windows Serial Port Programming Handbook, 2004. Crossref	12 words — < 1%
52	Jin-Cheon Na, Tun Thura Thet. "Effectiveness of web search results for genre and sentiment classification", Journal of Information Science, 2009 Crossref	12 words — < 1%
53	Uhlich, Stefan, Franck Giron, and Yuki Mitsufuji. "Deep neural network based instrument extraction from music", 2015 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2015. Crossref	12 words — < 1%
54	Tian Zhang, Tianqi Zhang, Congcong Fan. "Chapter 34 Accompaniment Music Separation Based on 2DFT and Image Processing (Workshop)", Springer Science and Business Media LLC, 2020 Crossref	11 words — < 1%
55	arrow.tudublin.ie Internet	11 words — < 1%
56	dokumen.pub Internet	11 words — < 1%
57	eprints.utar.edu.my Internet	11 words — < 1%
58	export.arxiv.org Internet	11 words — < 1%
59	mirlab.org Internet	11 words — < 1%
60	www.rdash.nhs.uk Internet	11 words — < 1%
61	arxiv-export-lb.library.cornell.edu Internet	10 words — < 1%
62	os.zhdk.cloud.switch.ch Internet	10 words — < 1%
63	patents.Google.com Internet	10 words — < 1%
64	repositorio-aberto.up.pt Internet	10 words — < 1%

65	repositorio.unifei.edu.br Internet	10 words — < 1%
66	scholar.ppu.edu Internet	10 words — < 1%
67	www.terasoft.com.tw Internet	10 words — < 1%
68	"Machine Learning and Autonomous Systems", Springer Science and Business Media LLC, 2022 Crossref	9 words — < 1%
69	3dfpo.swarthmore.edu Internet	9 words — < 1%
70	Bhuwan Bhattarai, Yagya Raj Pandeya, Joonwhoan Lee. "Parallel Stacked Hourglass Network for Music Source Separation", IEEE Access, 2020 Crossref	9 words — < 1%
71	Gao, Yi. "Adequacy assessment of composite generation and transmission systems incorporating wind energy conversion systems", Proquest, 20111109 ProQuest	9 words — < 1%
72	Jinjiang Liu, Xueliang Zhang. "DRC-NET: Densely Connected Recurrent Convolutional Neural Network for Speech Dereverberation", ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022 Crossref	9 words — < 1%
73	Reimer, L.. "Calculation of the angular and energy distribution of multiple scattered electrons using fourier transforms", Ultramicroscopy, 198910 Crossref	9 words — < 1%
74	e-sciencecentral.org Internet	9 words — < 1%
75	eprints.kingston.ac.uk Internet	9 words — < 1%
76	senior-secondary.scsa.wa.edu.au Internet	9 words — < 1%
77	www.epj-conferences.org Internet	9 words — < 1%

78	"4th Kuala Lumpur International Conference on Biomedical Engineering 2008", Springer Science and Business Media LLC, 2008 Crossref	8 words — < 1%
79	"Advances in Nonlinear Speech Processing", Springer Science and Business Media LLC, 2011 Crossref	8 words — < 1%
80	"Computational Collective Intelligence", Springer Science and Business Media LLC, 2020 Crossref	8 words — < 1%
81	"Global Trends in Information Systems and Software Applications", Springer Science and Business Media LLC, 2012 Crossref	8 words — < 1%
82	Chen, Wenqing . "Dynamic Vehicular Trajectory Optimization for Bottleneck Mitigation and Safety Improvement.", The University of Wisconsin - Milwaukee, 2020 ProQuest	8 words — < 1%
83	Christian Borss, Rainer Martin. "On the construction of window functions with constant-overlap-add constraint for arbitrary window shifts", 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012 Crossref	8 words — < 1%
84	Divyesh G. Rajpura, Jui Shah, Maitreya Patel, Harshit Malaviya, Kirtana Phatnani, Hemant A. Patil. "Effectiveness of Transfer Learning on Singing Voice Conversion in the Presence of Background Music", 2020 International Conference on Signal Processing and Communications (SPCOM), 2020 Crossref	8 words — < 1%
85	Earl J. Kirkland. "Chapter 1 Introduction", Springer Science and Business Media LLC, 2020 Crossref	8 words — < 1%
86	Hideyuki Tachibana, Takuma Ono, Nobutaka Ono, Shigeki Sagayama. "Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source", 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010 Crossref	8 words — < 1%
87	Nguyen, Minh Thac. "Public Policy Effects on Health and Education Outcomes", University of Illinois at Chicago, 2020 ProQuest	8 words — < 1%

88	Yordan Mirchev, Krasimir Staykov, Damyan Ganchev. "Application of Synthetic Aperture Focusing Technique for inspection of plate-like structures using EMAT generated Lamb waves", MATEC Web of Conferences, 2018 Crossref	8 words — < 1%
89	asmp-eurasipjournals.springeropen.com Internet	8 words — < 1%
90	dspace.mist.ac.bd:8080 Internet	8 words — < 1%
91	library.umac.mo Internet	8 words — < 1%
92	repository.dl.itc.u-tokyo.ac.jp Internet	8 words — < 1%
93	www.degruyter.com Internet	8 words — < 1%
94	www.sdbs.uop.gr Internet	8 words — < 1%
95	"Medical Image Computing and Computer Assisted Intervention – MICCAI 2019", Springer Science and Business Media LLC, 2019 Crossref	7 words — < 1%
96	"Review on Video Watermarking Techniques in Spatial and Transform Domain", Advances in Intelligent Systems and Computing, 2016. Crossref	7 words — < 1%
97	Oded Goldreich. "Locally testable codes and PCPs of almost-linear length", Journal of the ACM, 7/1/2006 Crossref	7 words — < 1%
98	"Audio Source Separation and Speech Enhancement", Wiley, 2018 Crossref	6 words — < 1%
99	"Deep Learning and Big Data for Intelligent Transportation", Springer Science and Business Media LLC, 2021 Crossref	6 words — < 1%

100 Wen-Hsing Lai, Siou-Lin Wang. "RPCA-DRNN technique for monaural singing voice separation", EURASIP Journal on Audio, Speech, and Music Processing, 2022 — 6 words — < 1%
Crossref

101 Xu, Zhanyou. "Machine Learning Analytics for Predictive Breeding.", Iowa State University, 2020 — 6 words — < 1%
ProQuest

References

[1] E. Cano, D. Fitzgerald, A. Liutkus, M. Plumbley and F.-R. Stöter, "Musical Source Separation: An Introduction," *IEEE Signal Processing Magazine,* vol. 36, no. 1, pp. 31-40, 2019.

[2] S. Sofianos, A. Ariyaeeinia and R. Polfreman, "Singing Voice Separation Based on Non-Vocal Independent Component Substraction and Amplitude Discrimination," in *13th Int. Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, 2010.

[3] H. Tachibana, T. Ono, N. Ono and S. Sagayama, "Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.

[4] Z. Rafii and B. Pardo, "A simple music/voice separation method based on the extraction of the repeating musical structure," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011.

[5] P. Seetharaman, F. Pishdadian and B. Pardo, "Music/Voice separation using the 2D fourier transform," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2017.

[6] C.-L. Hsu, D. Wang, J.-S. R. Jang and K. Hu, "A Tandem Algorithm for Singing Pitch Extraction and Voice Separation From Music Accompaniment," *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING,* vol. 20, no. 5, pp. 1482-1491, 2012.

[7] V. Nichal, M. V.A and G. D. , "Tandem Algorithm with Supervised Classifier for Pitch Estimation and Voice Separation from Music Accompaniments: Survey,"

International Journal of Science and Research (IJSR) , vol. 4, no. 2, pp. 212-215, 2015.

[8] Z. Rafii, A. Liutkus, F.-R. Stoter, S. I. Mimilakis, D. FitzGerald and B. Pardo, "An Overview of Lead and Accompaniment Separation in Music," *IEEE/ACM Transactions on Audio, Speech and Language Processing,* vol. 26, no. 8, pp. 1307-1335, 2018.

[9] A. Chanrungutai and C. A. Ratanamahatana, "Singing Voice Separation in Mono-Channel Music," in *2008 International Symposium on Communications and Information Technologies*, Vientiane, Laos, 2008.

[10] S. Mavaddati, "A Novel Singing Voice Separation Method Based on Sparse Non-Negative Matrix Factorization and Low-Rank Modeling," *Iranian Journal of Electrical and Electronic Engineering,* vol. 15, no. 2, pp. 161-171, 2019.

[11] D. FitzGerald, "Harmonic/Percussive Separation using Median Filtering," in *13th Int. Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, 2010.

[12] D. Fitzgerald and M. Gainza, "Single Channel Vocal Separation using Median Filtering and Factorisation Techniques," *ISAST Transactions on Electronic and Signal Processing,* vol. 1, no. 4, pp. 62-73, 2010.

[13] S. Uhlichi, F. Gironi and Y. Mitsufuji, "Deep Neural Network Based Instrument Extraction From Music," in *International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Queensland, Australia, 2015.

[14] P.-S. Huang, M. Kim, M. Hasegawa-Johnson and P. Smaragdis, "Singing-Voice Separation from Monaural Recordings using Deep Recurrent Neural Networks," in *International Society for Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.

[15] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahash and Y. Mitsufuj, "Improving Music Source Separation Based on Deep Neural Networks Through Data Augmentation and Network Blending," in *International Conference on Acoustics, Speech, & Signal Processing (ICASSP)*, New Orleans, , 2017.

[16] A. J. Simpson, G. Roma and M. D. Plumbley, "Deep Karaoke: Extracting Vocals from Musical Mixtures Using a Convolutional Deep Neural Network," in *12th International Conference on Latent Variable Analysis and Signal Separation*, Liberec, Czech Republic, 2015.

[17] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar and T. Weyde, "Singing Voice Separation with Deep U-Net Convolutional Netwrks," in *International Society for Music Information Retrieval (ISMIR)*, , Suzhou, China, 2017.

[18] A. Newell, K. Yang and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," European conference on computer vision, Springer, Cham, 2016.

[19] S. Park, T. Kim, K. Lee and N. Kwak, "Music Source Separation Using Stacked Hourglass Networks," in *International Society for Music Information Retrieval (ISMIR)*, Paris, France, 2018.

[20] A. Bugler, B. Pardo and P. Seetharaman, "A Study of Transfer Learning in Music Source Separation," 20 October 2020. [Online]. Available: arXiv preprint arXiv:2010.12650. [Accessed 29 December 2021].

[21] "The DSD100 Dataset," SiSEC MUS, [Online]. Available: https://www.sisec17.audiolabs-erlangen.de/#/dataset. [Accessed 06 06 2021].

[22] "The MUSDB18 Dataset," SiSEC MUS, [Online]. Available: https://sigsep.github.io/datasets/musdb.html.

[23] K. M. M. Prabhu, "Review of Window Functions," in *Window Functions and Their Applications in Signal Processing*, New York, USA, CRC Press, Taylor & Francis Group, 2014, pp. 87-126.

[24] E. Manilow, P. Seetharaman and J. Salamon, "Open Source Tools & Data for Music Source Separation," 15 October 2020. [Online]. Available: https://source-separation.github.io/tutorial. [Accessed 12 January 2022].

[25] A. Navlani, "Neural Network Models in R," 9 December 2019. [Online]. Available: https://www.datacamp.com/community/tutorials/neural-network-models-r. [Accessed 14 June 2021].

[26] V. S. Kadandale, J. F. Montesinos, G. Haro and E. G'omez, "Multi-channel U-Net for Music Source Separation," in *IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, Tempere , 2020.

[27] A. Défossez, N. Usunier, L. Bottou and F. Bach, "Music Source Separation in the Waveform Domain," 2021.