RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Project Report on

ChordCut

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Maria Diya Fiju (U2103130)

Mathew Jagan Thomas (U2103131)

Heinz Abraham Koshy (U2103102)

Juniot Mariyam Thomas (U2103119)

Under the guidance of

Ms. Meenu Mathew

Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2025

CERTIFICATE

This is to certify that the project report entitled ***"ChordCut"*** *is a bonafide record of the work done by* ***Maria Diya Fiju (U2103130), Mathew Jagan Thomas (U2103131), Heinz Abraham Koshy (U2103102), Juniot Mariyam Thomas (U2103119)****, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2021-2025.*

Ms. Meenu Mathew
Project Guide
Assistant Professor
Dept. of CSE
RSET

Ms. Anu Maria Joykutty
Project Co-ordinator
Assistant Professor
Dept. of CSE
RSET

Dr. Preetha K G
Professor & HOD
Dept. of CSE
RSET

ACKNOWLEDGEMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and Dr Preetha K G, Head of the Department of Computer Science and Engineering for providing us the opportunity to undertake our project, "ChordCut".

We are highly indebted to our project coordinators, **Ms. Anu Maria Joykutty**, Assistant Professor, Department of Computer Science and Engineering, **Dr. Sminu Izudheen**, Professor, Department of Computer Science and Engineering, for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide, **Ms. Meenu Mathew** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Maria Diya Fiju
Mathew Jagan Thomas
Heinz Abraham Koshy
Juniot Mariyam Thomas

Abstract

In the dynamic world of music production and audio engineering, the ability to distinguish and manipulate individual elements within a song—such as different musical instruments and vocals—is crucial for remixing, mastering, and various creative endeavors. The "ChordCut" project addresses this essential need by employing advanced machine learning techniques to accurately differentiate and isolate distinct audio components within a track. ChordCut is a powerful tool designed to process audio inputs and separate them into individual instrumental and vocal tracks. This allows users to remove, manipulate, or enhance specific elements with ease. Leveraging deep learning and spectral analysis, ChordCut identifies and extracts each component with remarkable precision, ensuring that the quality and integrity of the original sounds are preserved. ChordCut streamlines tasks such as vocal elimination, instrument isolation, and the creation of instrumental versions, making these audio manipulations efficient and accessible. As a versatile solution for music producers, audio engineers, and sound designers, ChordCut automates the complex process of audio element separation, significantly boosting workflow efficiency and expanding creative possibilities. This project represents a leap forward in audio processing technology, combining cutting-edge machine learning with practical audio engineering insights to deliver precise and high-quality sound separation. By addressing the challenges of audio component isolation, ChordCut is poised to become an indispensable tool in the music and audio production industry, driving innovation and excellence in sound manipulation.

Contents

List of Abbreviations

SDR - Signal-to-Distortion Ratio

MSS - Music Source Separation

MFCC - Mel-frequency Cepstral Coefficients

STFT - Short-Time Fourier Transform

GRU - Gated Recurrent Unit

CNN - Convolutional Neural Network

STDCT - Short-Time Discrete Cosine Transform

AUC ROC - Area Under the Receiver Operating Characteristic Curve

MSE - Mean Squared Error

List of Figures

List of Tables

Chapter 1

Introduction

The chapter presents some information of the "ChordCut" work, an innovative system which separates or distinguishes music sounds from sound recordings. Background, problem definition, scope, objectives, challenges, industrial and societal implications in terms of the project are all addressed

1.1 Background

Digital audio processing changed music, with artists, producers, and even listeners able to interact with it in new ways. But being able to disentangle particular musical elements from a mixed down track is still a major issue. The need for high-quality separation tools increases, especially in music production, karaoke, remixing, and podcasting. Current techniques suffer from interfering frequencies and loss of phase information, which compromise separation quality. ChordCut aims to solve these problems, offering a sophisticated, high-quality solution for separating musical elements.

1.2 Problem Definition

The goal of the project is to create a cutting-edge computer tool referred to as ChordCut capable of extracting distinct musical elements from a song. This would make it possible for users to re-use or remix audio recordings without degrading the quality and clarity of every one of the elements.

1.3 Scope and Motivation

ChordCut is applicable both to professionals in the music and audio sectors and also as a software for a music lover or an audio expert. ChordCut enables individuals to split audio files into its vocals, instruments, and percussion, making it easy to remix or alter

these parts. With the use of deep learning techniques, ChordCut provides zero error and slight distortion, assisting DJs and producers to create high-quality outputs. However, these kinds of processes previously were managed with hugely expensive equipment accompanied by sophisticated technical knowledge and are today facilitated for users and enthusiasts as well because of ChordCut.

1.4 Objectives

- Create a model that can effectively isolate vocals and instruments from mixed audio tracks.
- Reduce distortion and maintain sound quality while separating.
- Provide remixing features, such as pitch-shifting and time-stretching, for creative editing.
- Should have a simple interface to interact effortlessly with the functionalities of the tool.
- Experiment and try the functionality of the tool using a variety of music genres and levels of complexity levels of complexity and genres

1.5 Challenges

The primary obstacles of this project involve dealing with frequency overlap, preserving phase alignment, and ensuring minimal degradation of audio quality during separation and remixing.

1.6 Assumptions

1. Supported audio file types will be received by the modules.
2. Models previously utilized must be given required computational demands for training and deployment.
3. End-users possess basic knowledge of audio editing software.

1.7 Societal / Industrial Relevance

ChordCut has social and industrial uses. Socially, it allows music producers and musicians to reuse and remix music in a creative, lyric way, thus democratizing access to audio engineering. Industrially, ChordCut can be used as a commercial tool by audio engineers, broadcasters, and music producers who look for efficient dividing, editing, or remixing of audio components without degrading fidelity.

1.8 Organization of the Report

- **Chapter 1**

  The chapter gives an overall summary of the project's framework, presenting instrument and voice extraction problem definition. This chapter also explains the scope, motivation, goals, and difficulties involved in development of the project. It also specifies the assumptions made and underscores the social and business importance of successful voice isolation in audio technology.

- **Chapter 2**

  The chapter is an in-depth summary of the essential concepts, novel methods, and research achievements in voice and audio source separation. It covers state-of-the-art methods including deep learning, time-frequency processing, and audio processing. Major developments in Music Source Separation (MSS) and how they connect with methods applied to this project are highlighted. Significant models, such as convolutional and recurrent neural networks, spectrogram-based approaches, and hybrid approaches, are analyzed for their contribution to source separation. This chapter is the theoretical foundation for the methods employed in the ChordCut system.

- **Chapter 3**

  The chapter gives a broad description of the major units of the project, i.e., Audio feed and preprocessing, Attribute extraction, Sound isolation, and Output processing. Each unit's role is explained, and the methodologies used to deliver quality separation.

- **Chapter 4**

This chapter describes the experimental results and analysis of the vocal separation model. It provides performance evaluation on the basis of qualitative and quantitative parameters, and comparison evaluation in terms of existing methodologies. Problems that were faced during implementation are also described, along with suggestions for possible improvement and future work.

- **Chapter 5**

  This chapter gives the conclusion of the project, including the main findings and the success of the used vocal separation system. It presents the contributions, mentions limitations, and proposes potential future enhancements in music source separation.

1.9 Conclusion

This chapter has presented the aim and importance of ChordCut in the field of audio processing. Following chapters will increasingly discuss the technicalities, methodologies, and implication of the system.

Chapter 2

Literature Survey

Current developments in music source separation using deep learning technology approaches are reviewed in the introduction. Recent research work on MSS and audio classification is conducted in this study. Steps involved in these tasks, ie, instrument classification, vocal and accompaniment isolation are reinforced by implementing machine learning algorithms, and learning approach with respect to them are explained.

2.1 CatNet: Music Source Separation with Mix-Audio Augmentation(2021)

2.1.1 Introduction

CatNet is a framework that will attempt to resolve issues of musical elements separation from multitrack music files. CatNet employs both spectrogram-based and time-domain methods, based on U-Net and WavUNet structures, the information gained from both methods are combined to increase the accuracy of the corresponding MSS task.[1]

2.1.2 Methodology

CatNet uses both time-domain and spectrogram-based methods, benefiting from both UNet and WavUNet architectures to improve separation quality. One of the attributes of CatNet is that it utilizes mix-audio augmentation, wherein training samples are synthesized from individual sources in the MUSDB18 dataset. By so doing, this attribute enhances the capacity of the model to process a wide range of complex audio situations.

$$\hat{s} = \hat{s}_{\mathrm{U}} + \hat{s}_{\mathrm{WU}}$$

Figure 2.1: Output of CatNet

2.1.3 Results

CatNet resulted in a 7.54 dB vocal separation SDR that outperformed other methods such as MMDenseNet and Demucs. The performance improved greatly using the mix-audio augmentation technique, particularly on overlapping content cases.

2.1.4 Advantages

Complementary Data Integration: Integrates spectrogram and waveform data for improved separation outcomes.
Enhanced Robustness: Mix-audio augmentation improves performance on intricate inputs.
Superior Performance: Performs better with higher SDR values than MSS approaches previously.

2.1.5 Disadvantages

High Computational Load: Dual processing of spectrogram and waveform data increases computational demand.
Dataset Dependency: Certain datasets are required for effective augmentation, which may hamper generalizability.

2.1.6 Conclusion

CatNet achieves significant advancements in MSS by combining spectrogram and waveform data and capitalizing on mix-audio augmentation. Future studies might investigate usage beyond MSS and improve computing capability.

2.2 Efficient Short-Time Discrete Cosine Transform and MultiResUNet Framework for Music Source Separation(2022)

2.2.1 Introduction

The research presents a new methodology for MSS using an Attentive MultiResUNet network in combination with the Short-Time Discrete Cosine Transform (STDCT). In

contrast to the Fourier Transform, STDCT avoids phase recovery issues, thereby improving separation quality and computational costs. The focus of the paper is primarily vocal, bass, and drum separation in dense audio mixes.[2]

2.2.2 Methodology

The model incorporates MultiResUNet with attention mechanisms to achieve improved feature separation and extraction capability. The model training is dependent on real-valued STDCT spectrograms drawn from the MUSDB18 database, avoiding computational complexity through handling complex numbers.

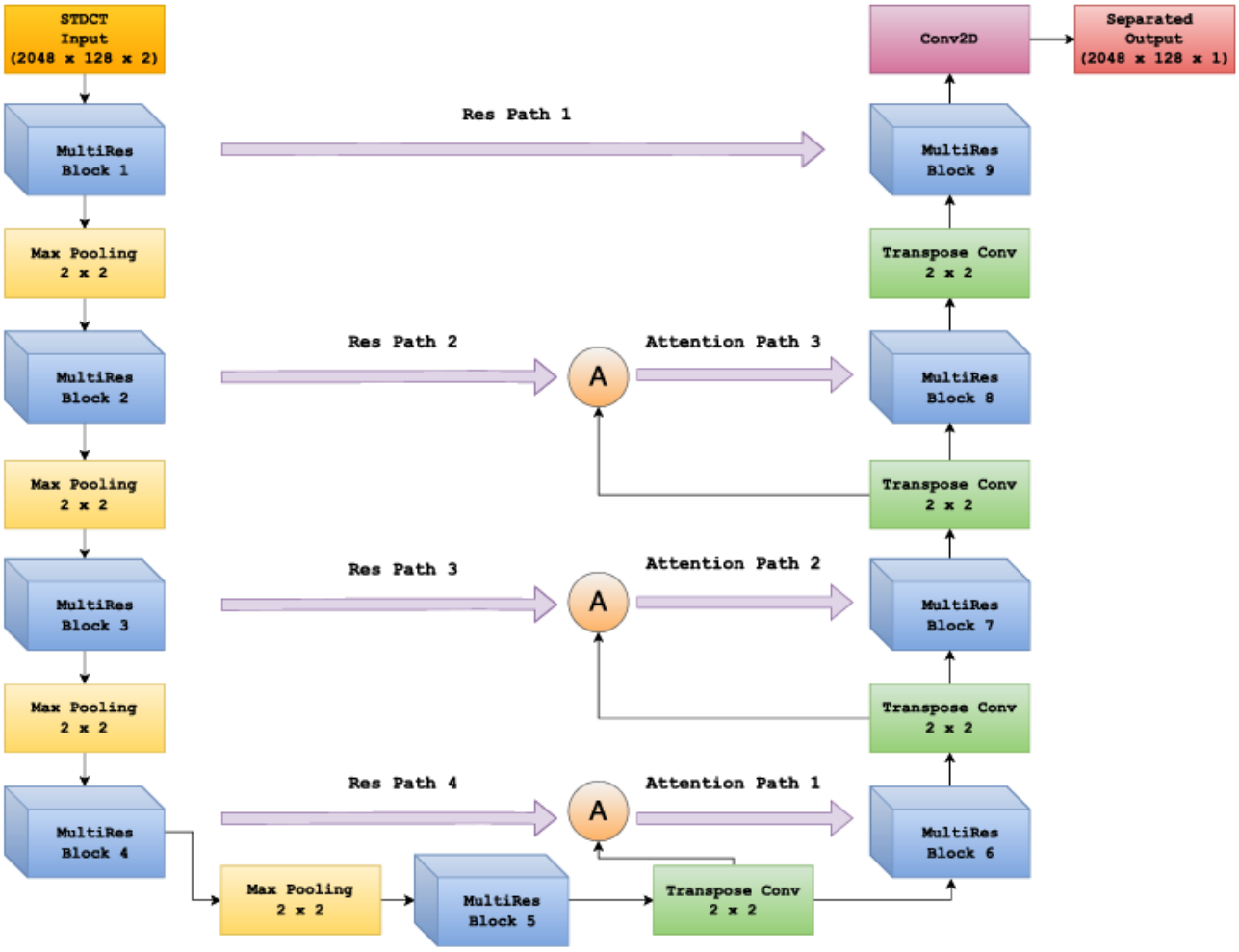


Figure 2.2: Attentive MultiResUNet Architecture

2.2.3 Results

The procedure attained comparative SDR levels and efficiency gain over earlier applied models. STDCT and attention mechanisms helped to efficiently manage phase information, making the system prepared for real-time use with reduced computational complexity.

2.2.4 Advantages

Efficiency: Minimizes computational burden via real-valued transformations.
Higher Accuracy: Mechanisms of attention improve source separation accuracy.
Scalability: Extension to various tasks of separation via the MultiResUNet design.

2.2.5 Disadvantages

Limited Temporal Context: Capable of becoming stuck in attempts to capture dependencies over long term.
Stereo-Specific Design: Mainly optimized for the stereo audio and thus may prove less adaptable when dealing with multichannel schemes.

2.2.6 Conclusion

By integrating STDCT and MultiResUNet, this paper introduces an effective MSS framework with robust performance. Its future work should be to enhance temporal modeling and extend its application to various audio formats.

2.3 Data Augmentation for Audio Classification(2020)

2.3.1 Introduction

This study investigates different data augmentation strategies to improve deep learning-based audio classification models. These approaches generate heterogeneous training samples, eliminating overfitting and model improvement in sparse data scenarios.[3]

2.3.2 Methodology

The comparison is made for a number of augmentation methods such as Gaussian noise addition, pitch shifting, and SpecAugment. These are methods that distort raw audio or spectrograms in order to make training data more diverse, thereby allowing models to generalize more.

	Input	Sample	Label	Mode
Add Noise	raw audio	warp	preserving	–
Time Stretch	raw audio	warp	preserving	–
Pitch Shift	raw audio	warp	preserving	–
Cutout[5]	spectrogram	mask	preserving	nonlinear
Mixup[8]	spectrogram	mix	combination	linear
SamplePairing[7]	spectrogram	mix	combination	linear
SpecAugment[11]	spectrogram	mask	combination	nonlinear
SpecMix[17]	spectrogram	mix	combination	nonlinear
VH-Mixup[10]	spectrogram	mix	combination	nonlinear
Mixed Frequency Masking	spectrogram	mix	combination	nonlinear/linear

Figure 2.3: Comparison of various data augmentation methods

2.3.3 Results

Mixed Frequency Masking and SpecAugment achieved maximum classification accuracies, and they boosted mAP@3 by over 1% compared to the baseline. Mixed-spectrogram methods worked best to achieve model robustness.

2.3.4 Advantages

Better Generalization: Avoids overfitting by providing varied training samples.
Efficient in Low-Data Cases: In situations where large datasets are not present.

2.3.5 Disadvantages

High Computational Expense: Some augmentation techniques require high computation costs.
Narrow Applicability: All techniques have dedicated applications.

2.3.6 Conclusion

This research stresses the significance of data augmentation towards better audio classification models. Research could be pursued in other data augmentation techniques that could further strengthen generalization in the future.

2.4 Neural Network-Based Techniques for Vocals-Accompaniment Separation(2023)

2.4.1 Introduction

This study compares GRU and CNN models for vocals-accompaniment separation based on their performance in real-time audio processing..[4]

2.4.2 Methodology

Comparisons were made between a semantic segmentation-based CNN-based model and a GRU-based model that leverages sequential dependencies. The GRU model was superior in exploiting temporal knowledge of audio separation.

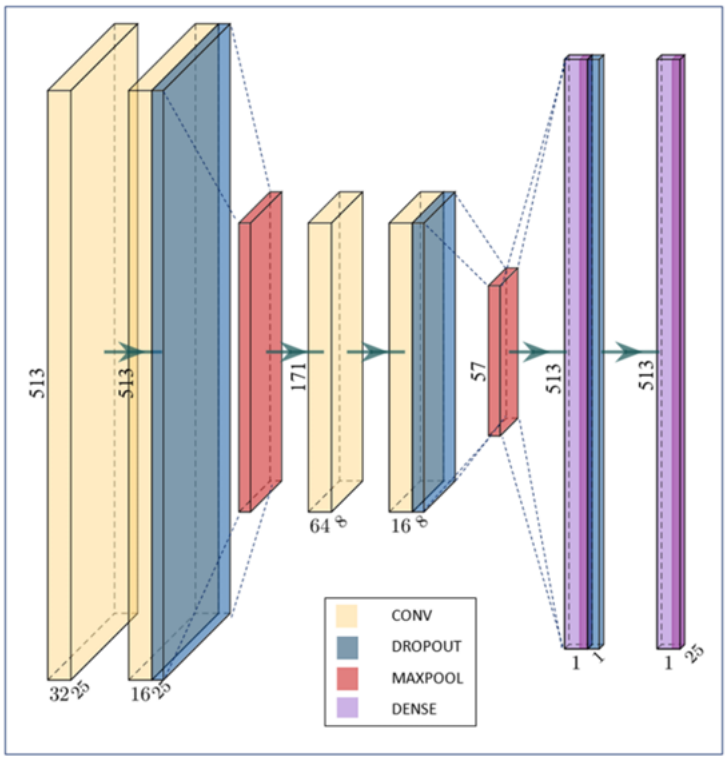


Figure 2.4: Hard Mask CNN Architecture

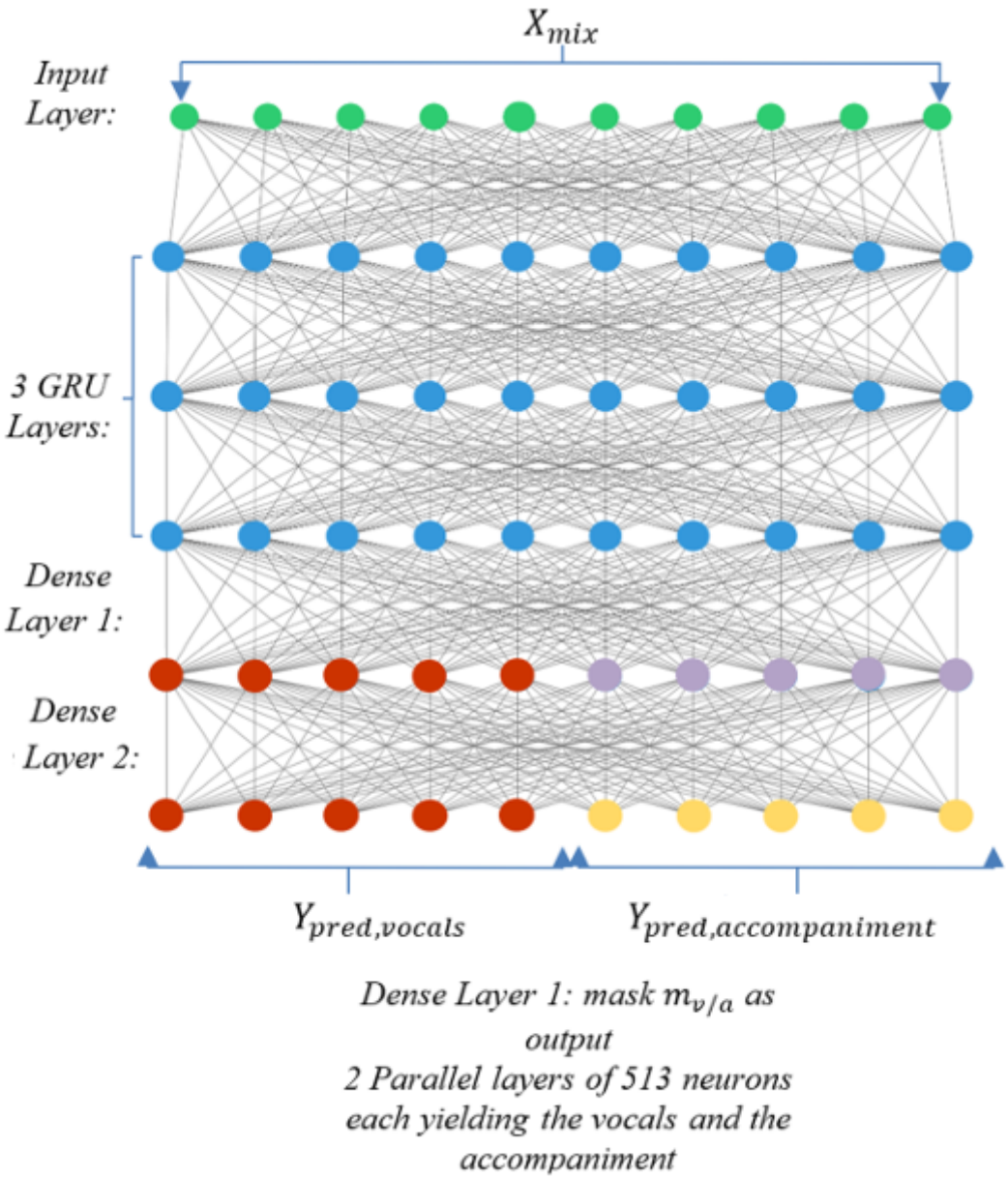


Figure 2.5: GRU Architecture

2.4.3 Results

GRU model fared better, especially in difficult situations with overlapped audio components.

2.4.4 Advantages

Increased Time Based Sensitivity: GRU achieves the representation of sequential audio patterns.

Increased Accuracy: Better performance on complex mixes.
Potential for Real-Time Use: With high computational demands, GRU offers great potential for real time processing of audio .

2.4.5 Disadvantages

Increased Computational Demand: GRU models are computationally expensive.
CNN drawbacks: CNN performs poorly with sequential information, limiting accuracy in complicated situations.

2.4.6 Conclusion

The GRU model demonstrates high potential for real-time MSS. Optimization would improve its computational efficiency.

2.5 Deep Learning Approaches for Musical Instrument Identification(2022)

2.5.1 Introduction

The research employs CNNs in musical instrument identification in polyphonic audio to facilitate automatic music analysis and retrieval.[5]

2.5.2 Methodology

The model employs specialized CNN pathways for distinct instrument features. Training involves different samples of instruments, with the assessment being on recall, precision and AUC ROC.

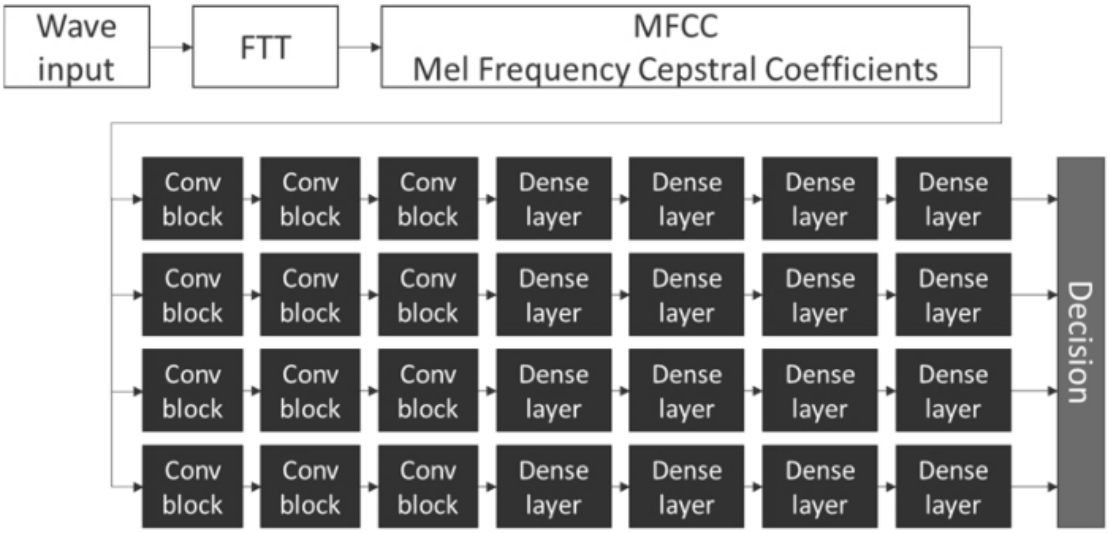


Figure 2.6: Deep CNN Architecture

2.5.3 Results

The method had high accuracy, with precision being 0.99 for some instruments such as drums.

2.5.4 Advantages

Scalability: Scalable with ease to identify more instruments.
High Accuracy: Works fine with well-defined instrument sounds.
Efficiency in Polyphonic Mixes: Works fine with complex audio scenarios.

2.5.5 Disadvantages

Computational Complexity: Requires significant processing resources.
Limited Temporal Analysis: CNNs struggle to model time-based dependencies.

2.5.6 Conclusion

The work is able to illustrate the capability of CNNs for instrument identification while pointing towards enhanced temporal modeling as well as greater efficiency in follow-up research.

2.6 Summary and Gaps Identified

The table below summarizes the strengths and weaknesses of each music source separation method described above.

Table 2.1: Summary of Music Source Separation Techniques

Method	**Advantages**	**Disadvantages**
CatNet:Music source separation with audio mix augmentation	Using spectral representation and sound wave data, good performance for audio mixture	Significant demand of computational resources, only for expert datasets
Short-Time Discrete Cosine Transform Optimized and MultiResUNet Architecture	High efficiency, low complexity, good usage of attention mechanism	Narrow temporal context, especially appropriate for stereo mixtures
Data Augmentation for Audio Classification	Improved generalization, good when used with small data	Highly computationally expensive, usable only for certain kinds of augmentation
Neural Network-Based Methods for Vocals-Accompaniment Separation	High temporal sensitivity, good for real-time usage	High computational expense, constrained by CNN ability to process temporally
Deep Learning Methods for Musical Instrument Recognition	High precision in polyphonic sound, elastic CNN pathways, flexible towards various instruments	Computationally expensive, constraining in temporal analysis

Although deep learning methods have been improving MSS, there are still some gaps:

1. Limited Real-Time Processing Capabilities: Models such as CatNet, operating on waveforms as well as spectrograms, are highly computationally demanding, and their application in real time is difficult. The issue lies in making them quicker without lowering accuracy for real-world applications.

2. Dataset Dependency and Generalizability: CatNet models depend on labeled datasets for their best performance, but it may restrict their generalizability.

3. Lack of Temporal Dynamics: Current models that use CNNs, have trouble capturing the change sounds have over time. This is a problem because music and speech are dynamic—they evolve with time. Adding models like GRUs, which are better at recognizing patterns over a period of time could help the system separate different sounds more accurately.

4. Complexity in Handling Phase Information: Methods using Fourier-based transformations struggle to recover phase information, which affects how well they can separate audio sources. Phase-free methods, eg: STDCT, can be a feasible solution.

5. High Computational Cost of Data Augmentation Techniques: Data augmentation will enhance the model on new sounds. But techniques, such as SpecAugment, are slow and need lots of computing power. So researchers are searching for quicker, more efficient methods of achieving the same advantages without consuming too many resources.

Literature review indicates satisfactory progress in deep learning-based MSS, with the models utilizing heterogeneous data inputs and intricate network architectures to achieve improved audio separation and classification. Key issues persist—real-time processing, temporal dependency modeling, and generalizability across datasets. These can be addressed by scaling computing power, enhancing temporal modeling, and designing more flexible data augmentation methods.

Chapter 3

System Design

The "ChordCut" project consists of dominant modules, each of which is destined to design and manage some piece of the system functionality to achieve efficient and high-grade separation of vocals.

3.1 System Architecture

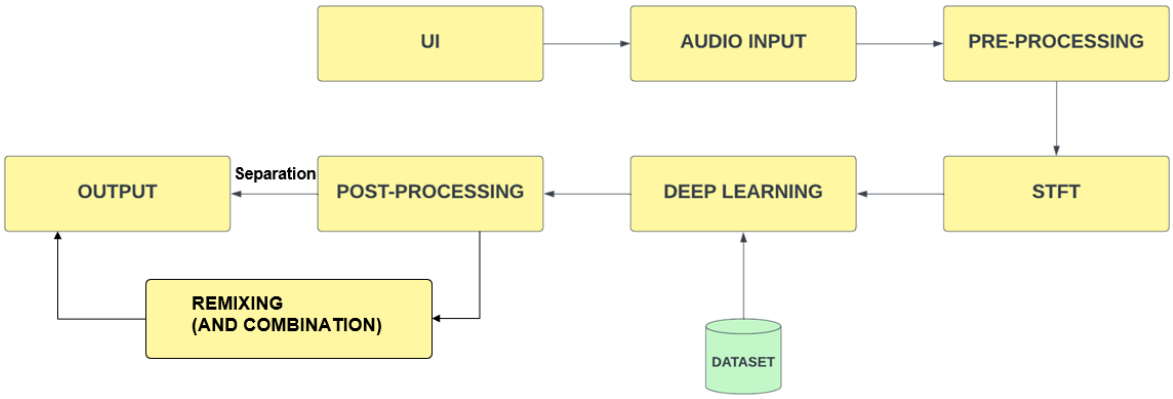


Figure 3.1: Architecture Diagram

3.2 Designing Components

1. User Interface:

   It is a connection between end user and system

2. Audio Input:

   Is responsible for ingesting audio data into the system.

3. Pre-Processing:

   Processes raw audio data in order to transform and analyze it.

4. Short-Time Fourier Transform(STFT):

   Transforms audio signals from the time domain to the frequency domain for improved processing.

5. Deep Learning:

   Uses neural networks to conduct audio separation as the fundamental operation.

6. Dataset:

   Provides labelled dataset that train and test the given model.

7. Post-Processing:

   Smooths the deep learning algorithm output to achieve high quality.

8. Output:

   Provides the final processed audio to the user.

3.3 Data Flow Diagram (DFD)

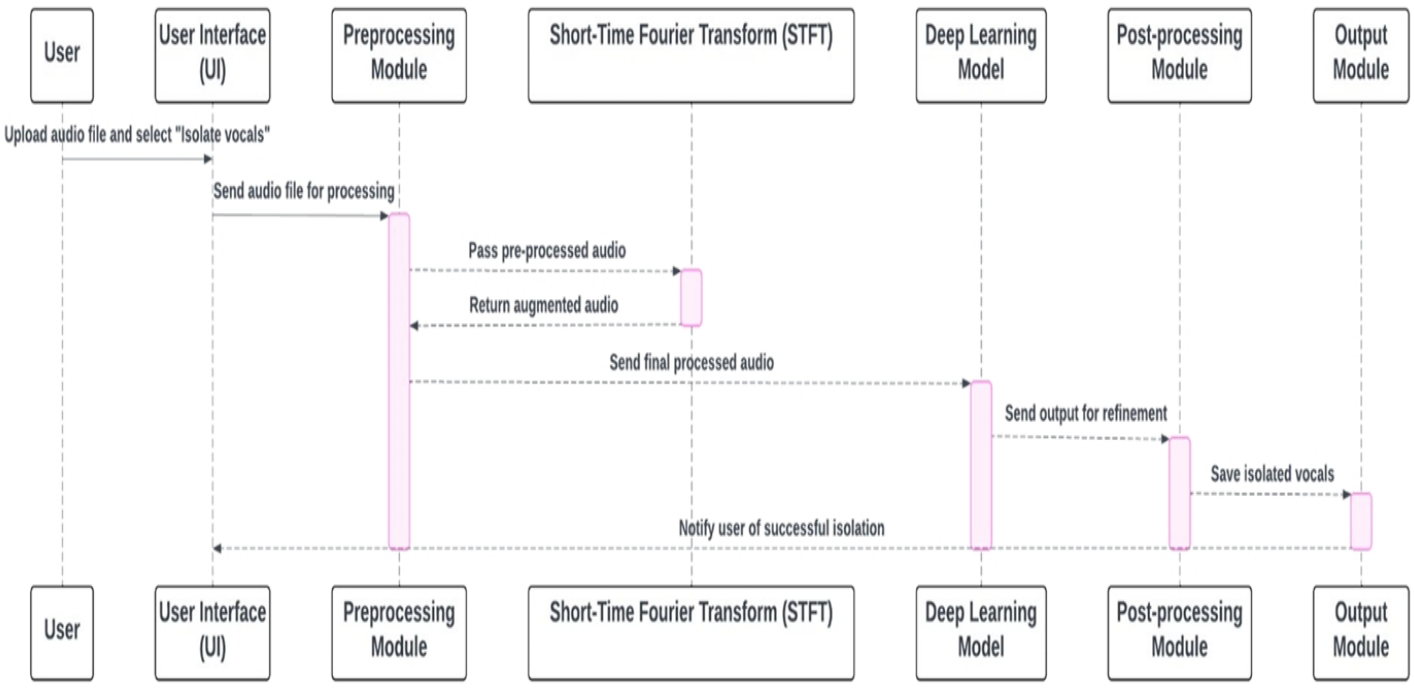


Figure 3.2: Sequence Diagram

3.4 Tools and Technologies: S/w and H/w Requirements

ChordCut utilizes modern tools and technologies in audio separation but in a much simpler manner. The project is implemented on Python, which has a vast library environment. TensorFlow is utilized to execute deep learning models, and librosa and other audio processing libraries are utilized for feature extraction and preprocessing . The UI is based on Django as the core GUI toolkit with a light weight and human-oriented web interface. Django enables users to load audio files, define parameters,and obtain processed results. The hardware demand is there to support high-end GPUs for the training of deep learning models, sufficient storage size for large dataset and intermediates, and fidelity-rich audio in/output devices to handle quality input/output processing and playback. It will provide a viable computationally empowered platform with easy UI.

3.5 Dataset Identified

Training is done on MUSDB18 dataset, a top freely available benchmark for music source separation. The dataset has stereo audio recordings with different instruments and vocals, offering suitable conditions for learning the neural networks. Genre diversity and blending style ensure that the model will act correctly under different audio conditions.

3.6 Module Division

3.6.1 Audio Input and Pre-processing

The first steps involved in processing raw audio input to the system. Sound data is especially significant in the pre-processing activities such as noise reduction, normalization, resampling. All of these activities establish consistency across all inputs. Optimization of audio data provides the system with more efficient feature separation and extraction, which leads to more accuracy.

3.6.2 Feature Extraction

Feature extraction of preprocessed audio, i.e., instrumentation dynamics and vocal features, is performed here. Basic techniques used here are MFCCs, STFT, and Spectral analysis. These are used for useful frequency and time information extraction.

3.6.3 Source Separation

ChordCut project had a module used for source-separating instrumentals and vocals from audio files. It made use of the Unet model, a specialized neural model that excels at audio separation. The system used frequency and time properties to achieve quality separation under interference. This effectively separated accompaniments from vocals among the users.

3.6.4 Output Processing

Improving the sound quality of output through application of normalization to equalization to post-filtering which results in isolated instrumental vocal tracks. The sequence above enhance overall sound quality.

3.7 Project Timeline

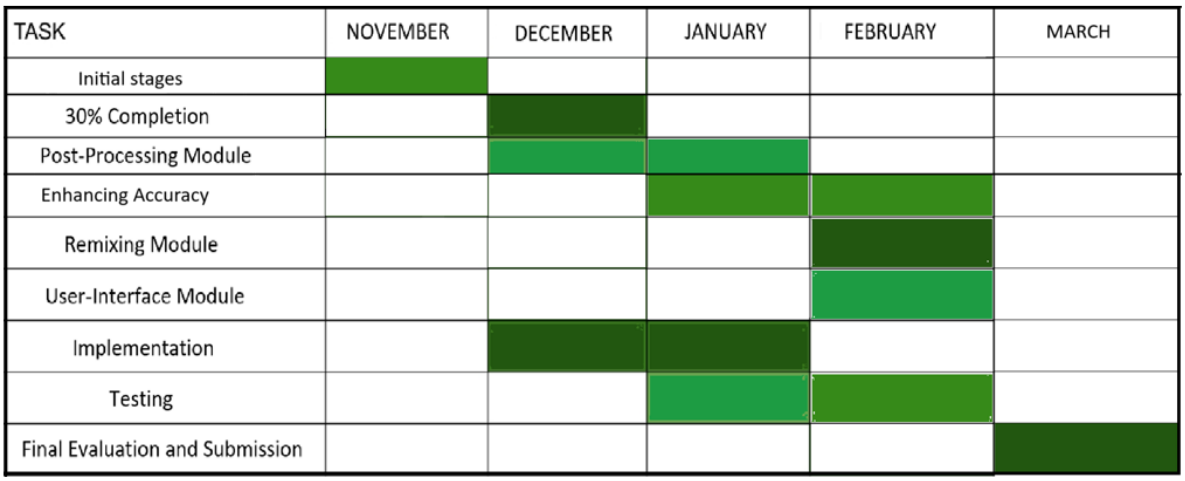


Figure 3.3: Gantt Chart

Chapter 4

Results and Discussions

This chapter gives a detailed analysis of the outcome of the results from the ChordCut project and the performance, challenges, and future scope of the system developed.

4.1 Introduction

This chapter reports the results, discussion, and outputs derived from the ChordCut project. The main goal of this project was to come up with a high-performance music source separation system based on deep learning methods. The outcome is discussed according to spectrogram visualizations, waveform differences, and audio quality measures.

4.2 Results

4.2.1 Spectrogram Analysis

The below spectrograms show the success of the separation process with the suggested deep learning model.

- Input Spectrogram: Displays the mixed audio input with multiple instruments and vocals.
- Separated Vocals Spectrogram: Emphasizes the isolated vocal component after separation.
- Separated Instrumental Spectrogram: Displays the instrumental components separated from the mix.

4.2.2 Waveform Comparisons

- Waveform analysis assures the integrity of phase and time coherence between separated sources.

- The reconstructed waveforms showed minimal distortion.
- The harmonic structure of the separated parts remained preserved.

4.3 Discussions

4.3.1 Comparison with Existing Methods

- In comparison to other traditional separation techniques, ChordCut performed better at maintaining the harmonic integrity of the sources.
- The use of U-Net and data augmentation methods greatly enhanced the accuracy of separation.
- The phase recovery operation was also optimized through the use of spectral modeling methods.

4.3.2 Challenges Encountered

- Frequency components that overlapped made it difficult to separate accurately.
- High computation time due to the complexity of the neural network.
- Limited generalization ability due to small datasets.

4.3.3 Future Enhancements

- Optimization of the deep learning model to minimize computational overhead.
- Increasing the dataset size to enhance generalization over various music genres.
- Real-time processing ability for live audio applications.
- More remixing options.

4.4 Outputs

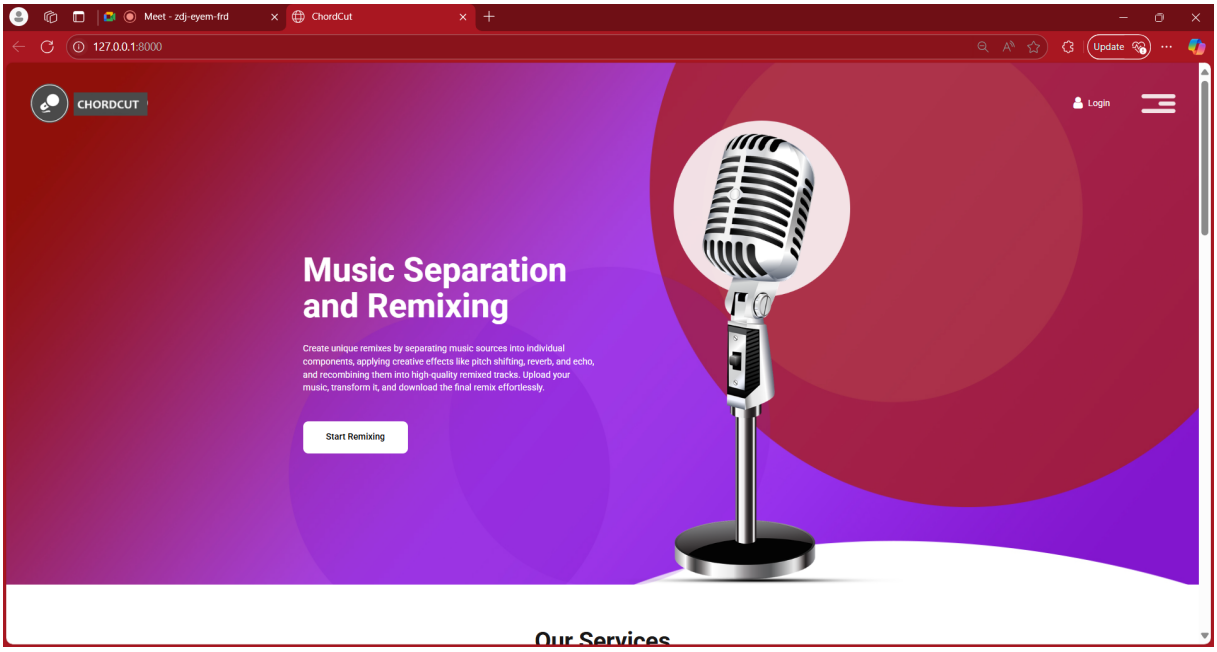


Figure 4.1: Website

4.5 Outputs

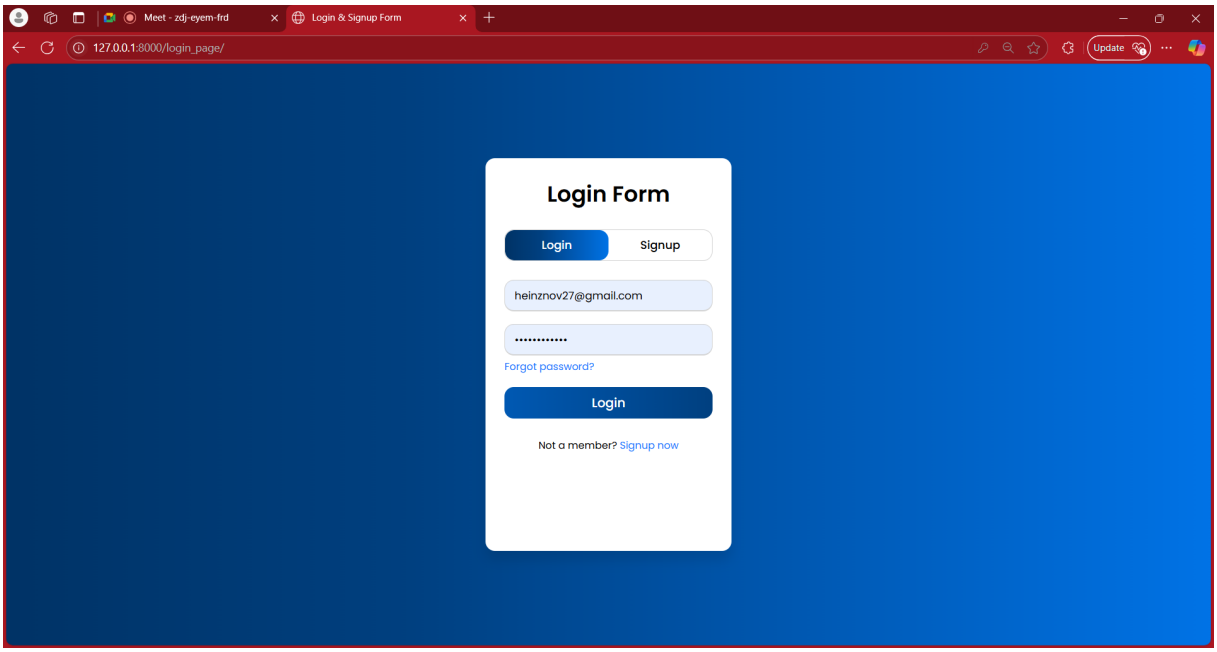


Figure 4.2: User Login

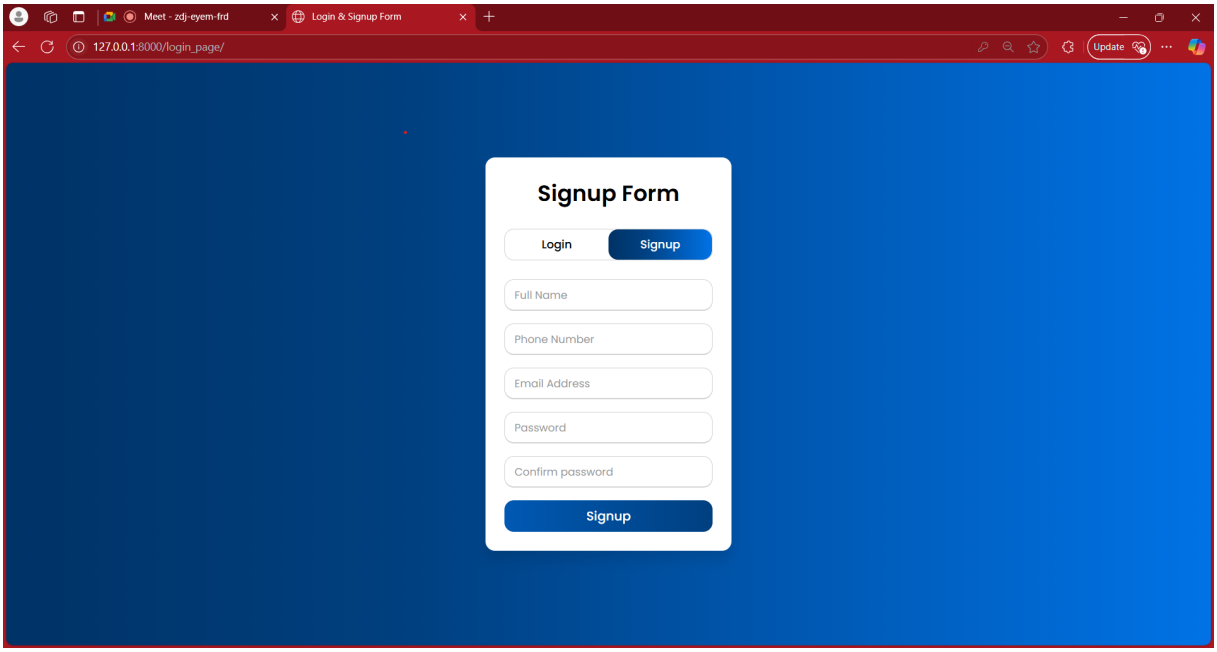


Figure 4.3: User Signup

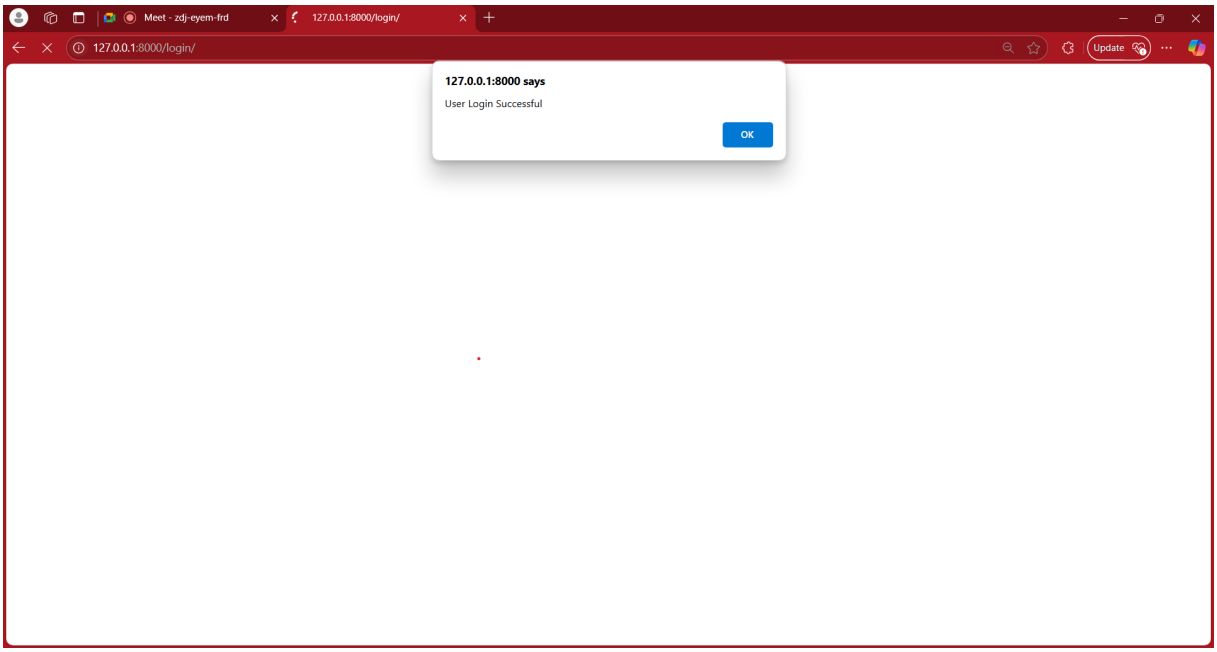


Figure 4.4: User Login Comfirmation

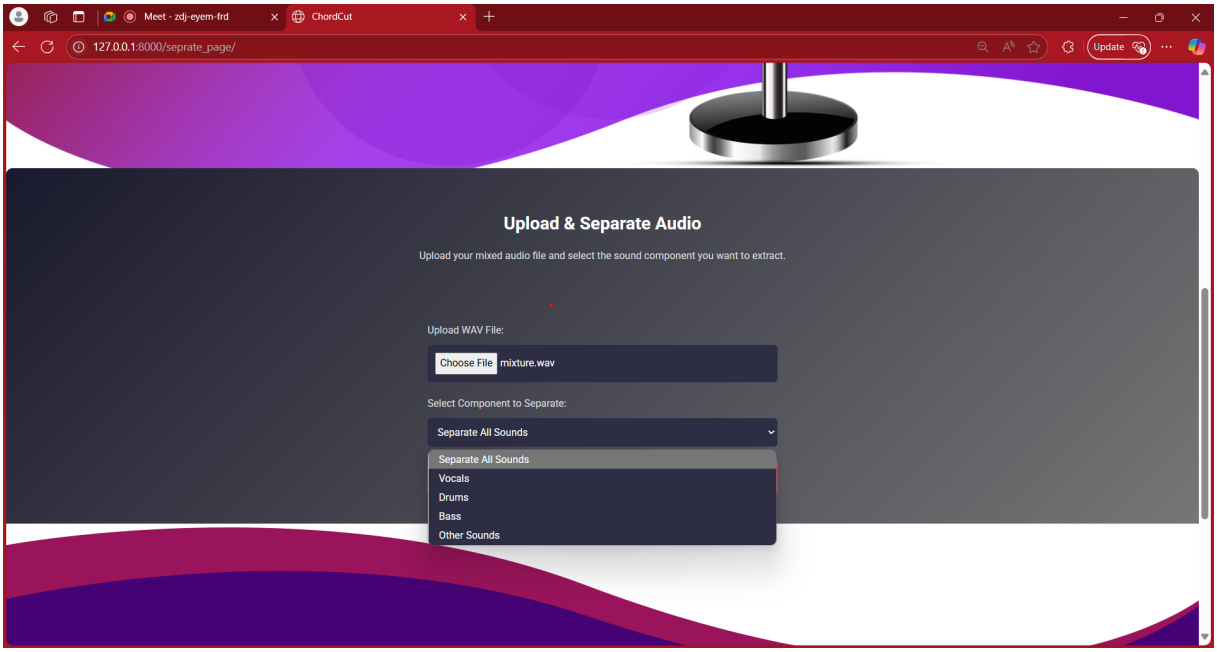


Figure 4.5: Audio Separation Interface

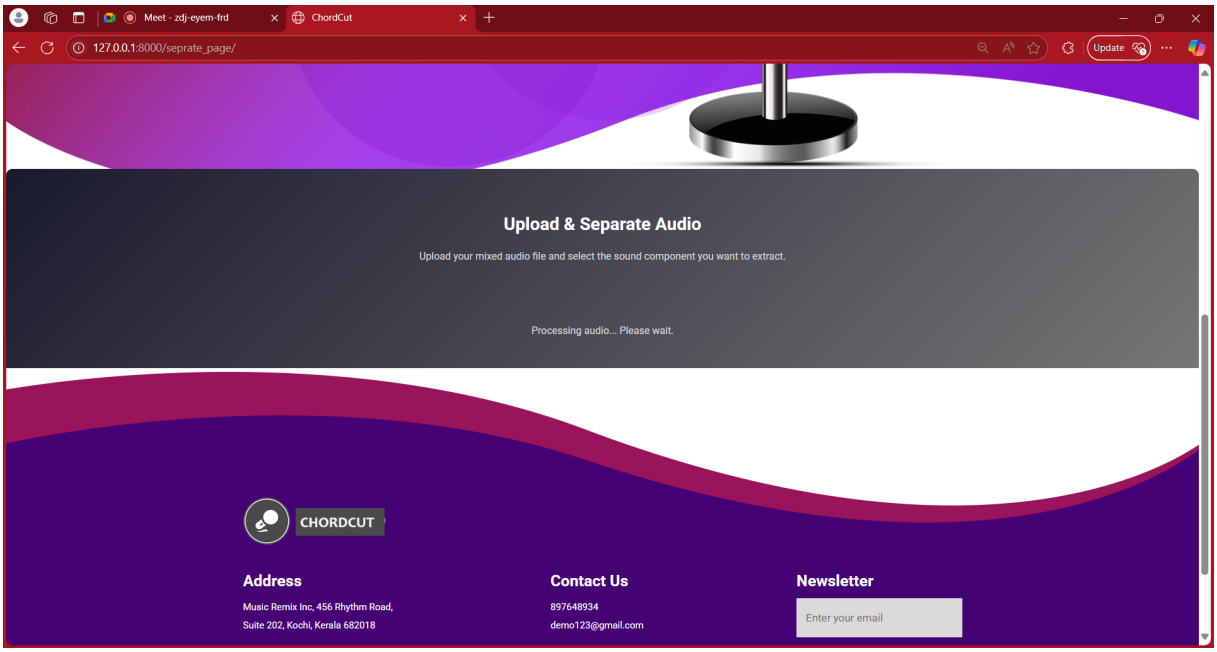


Figure 4.6: Audio Separation Processing

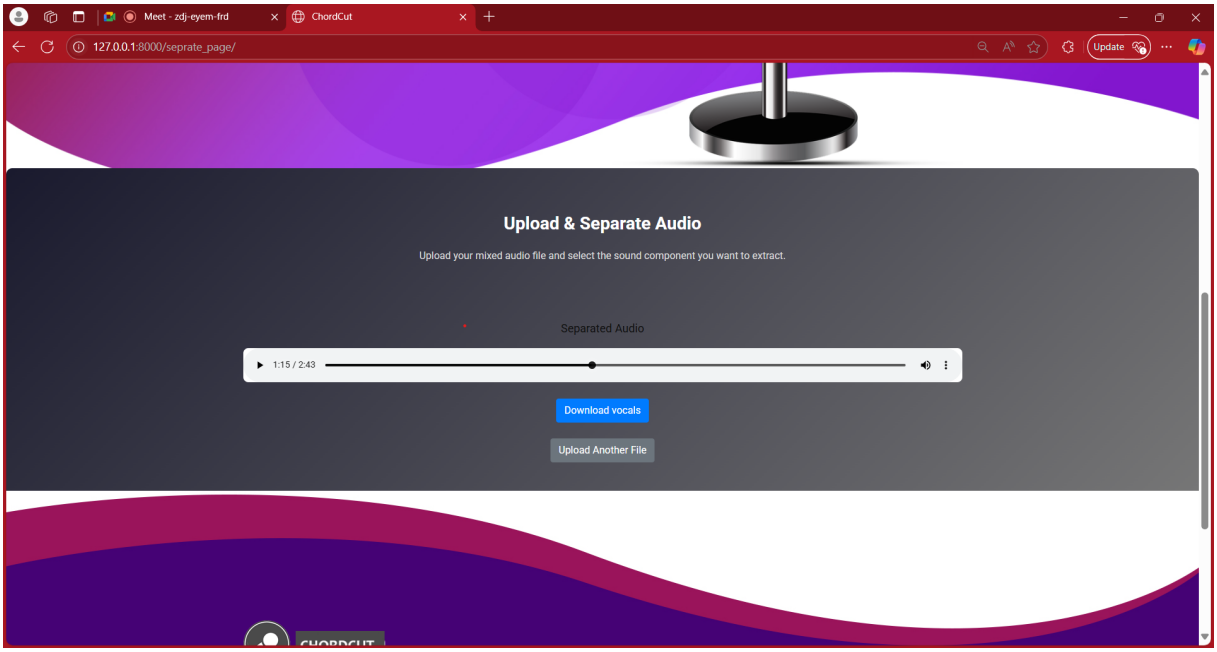


Figure 4.7: Separated Audio

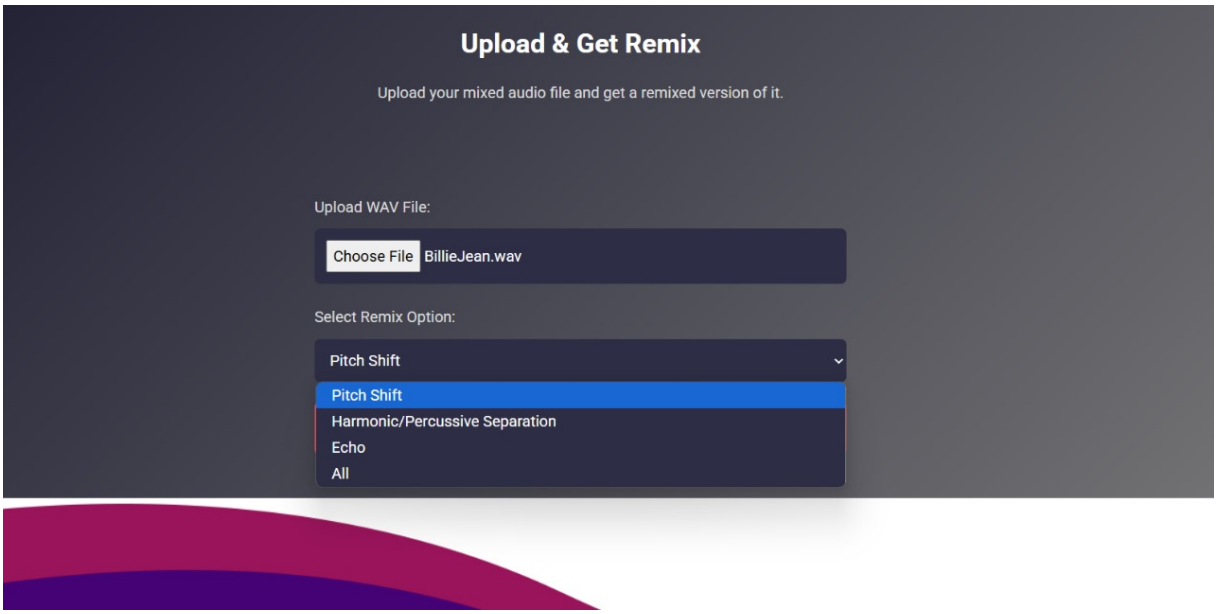


Figure 4.8: Audio Remixing Interface

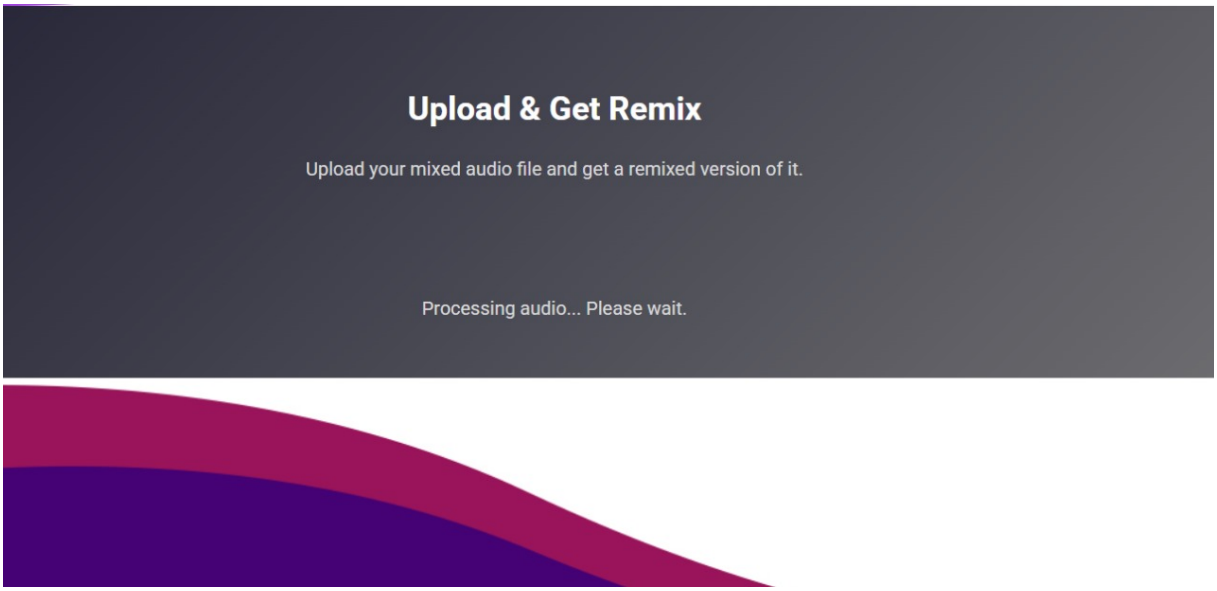


Figure 4.9: Audio Remix Processing

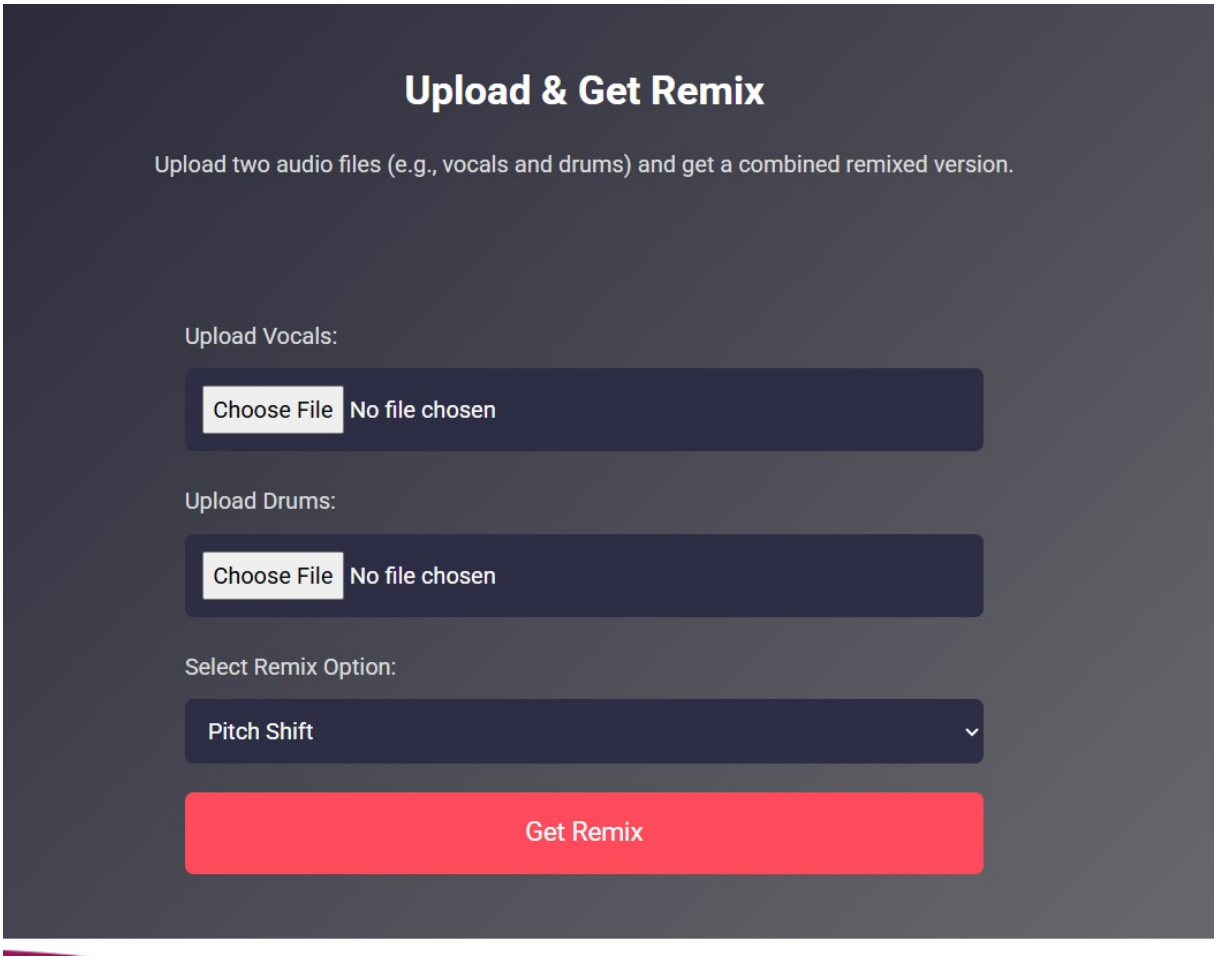


Figure 4.10: Audio Combining Interface

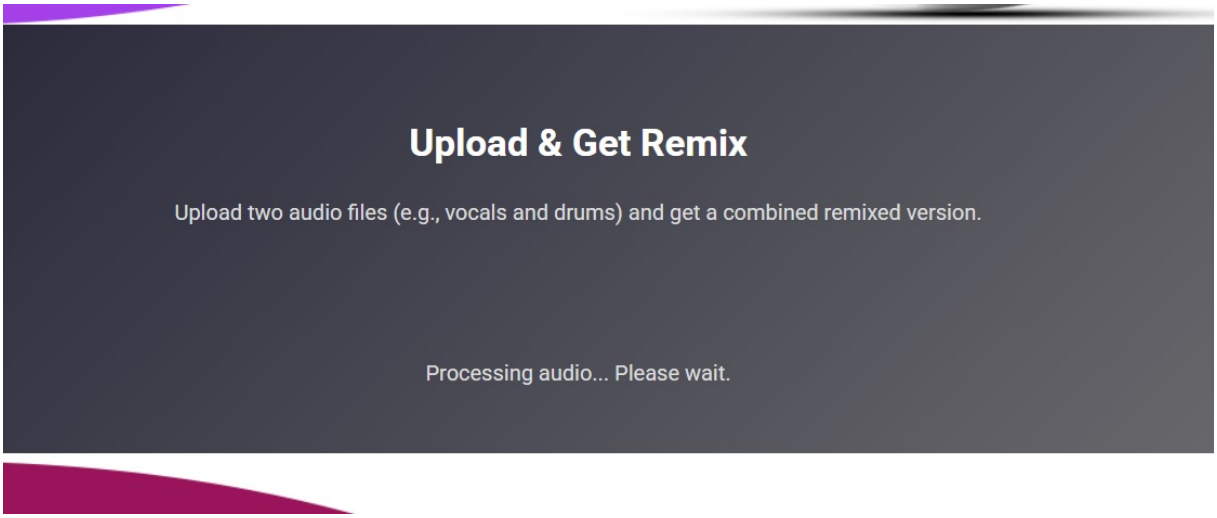


Figure 4.11: Audio Combine Processing

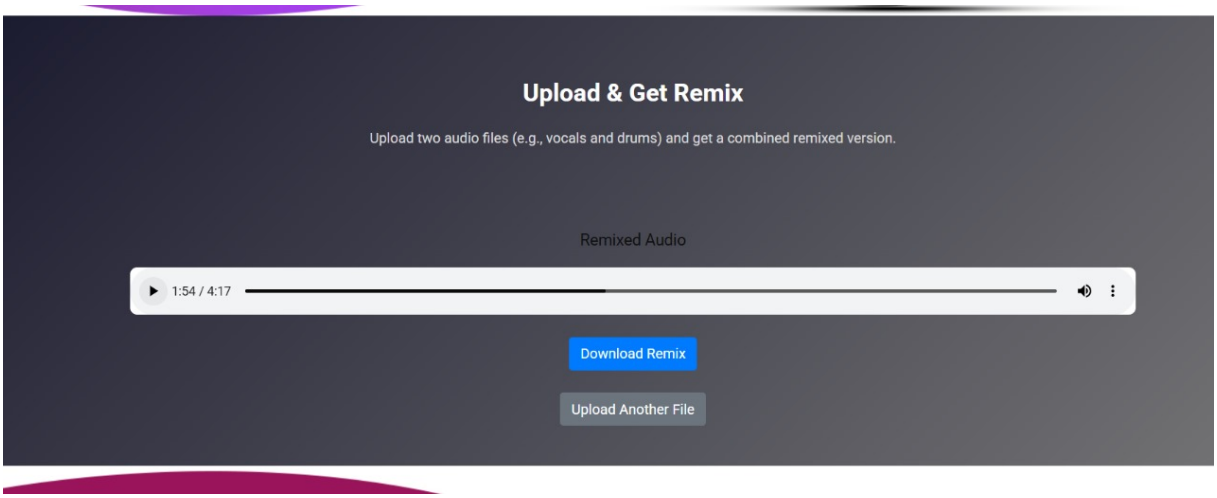


Figure 4.12: Audio Combined Output File

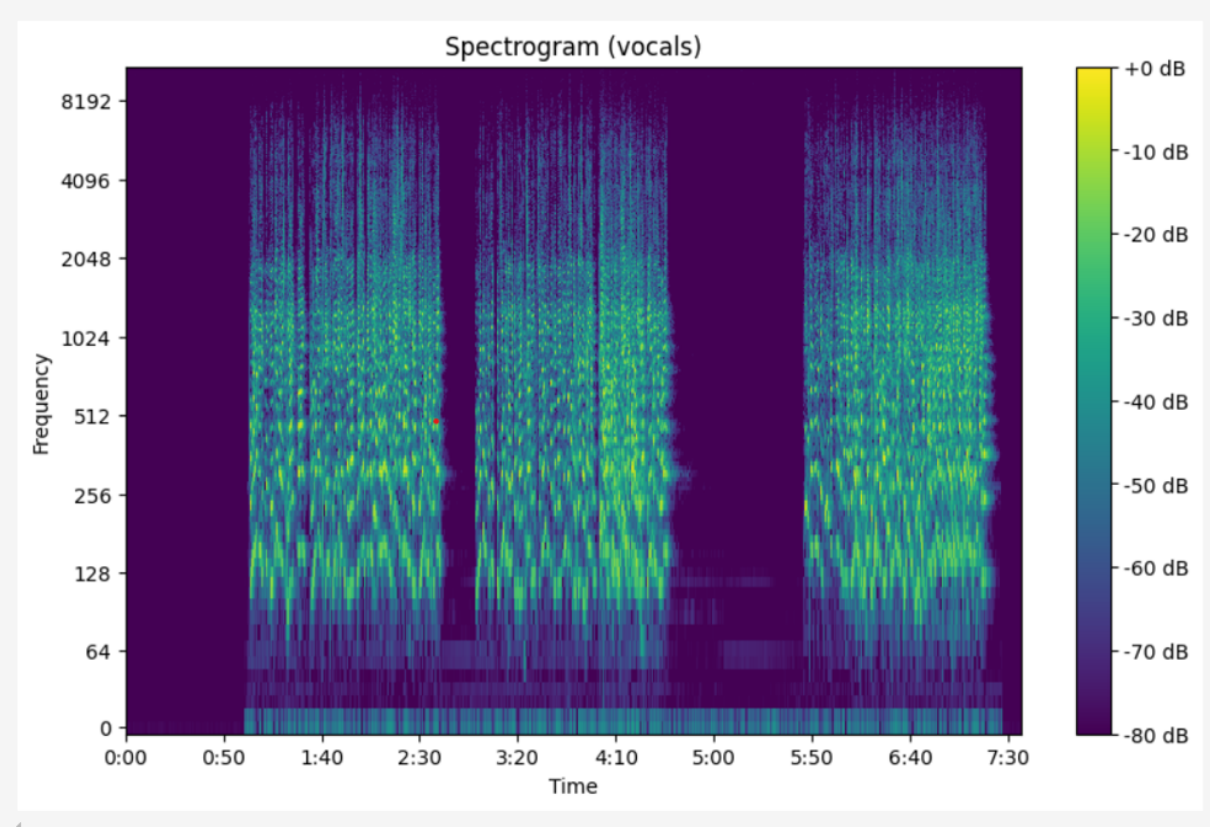


Figure 4.13: Spectrogram-Vocals

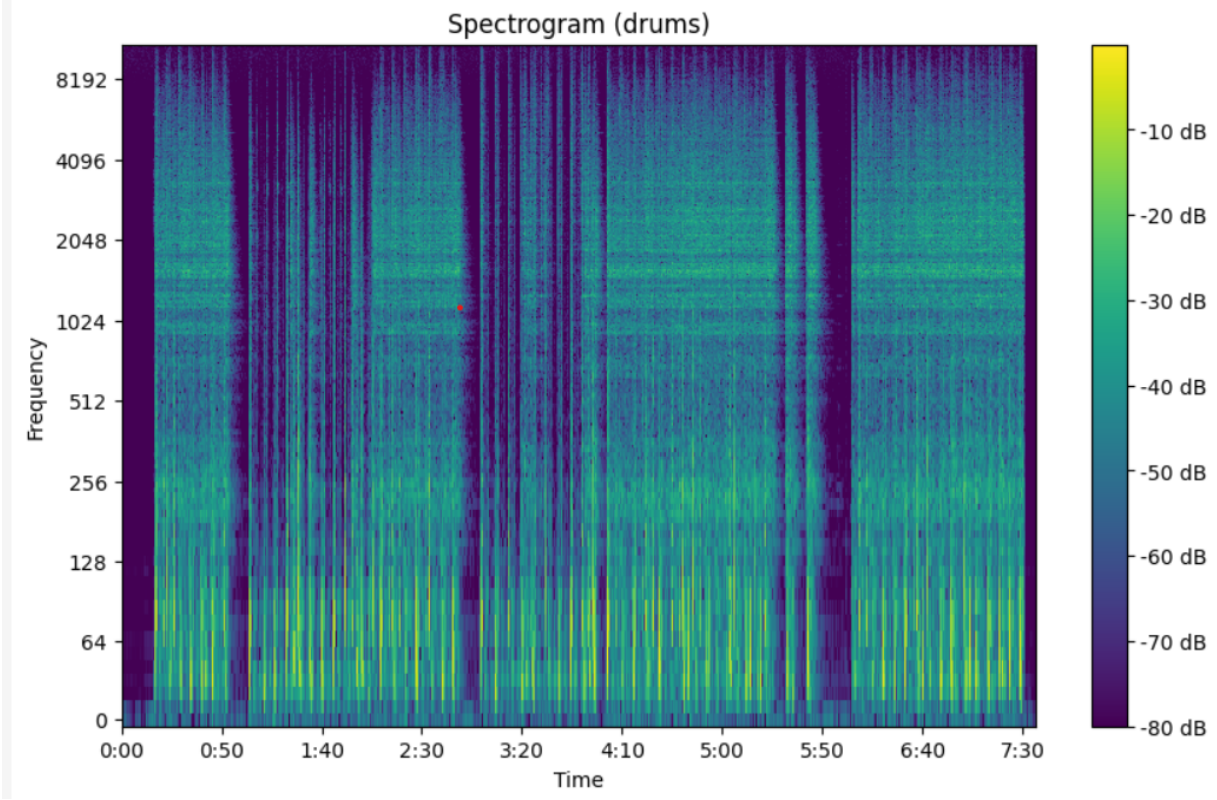


Figure 4.14: Spectrogram-Drums

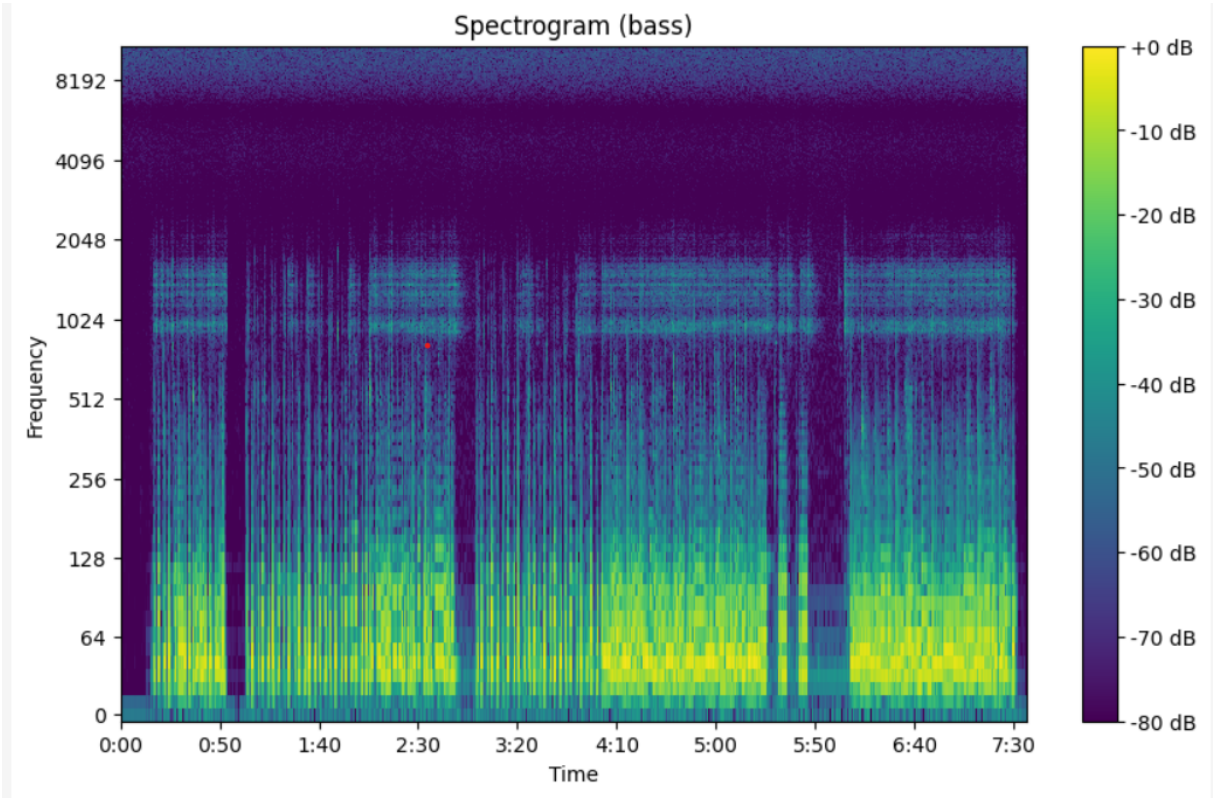


Figure 4.15: Spectrogram-Bass

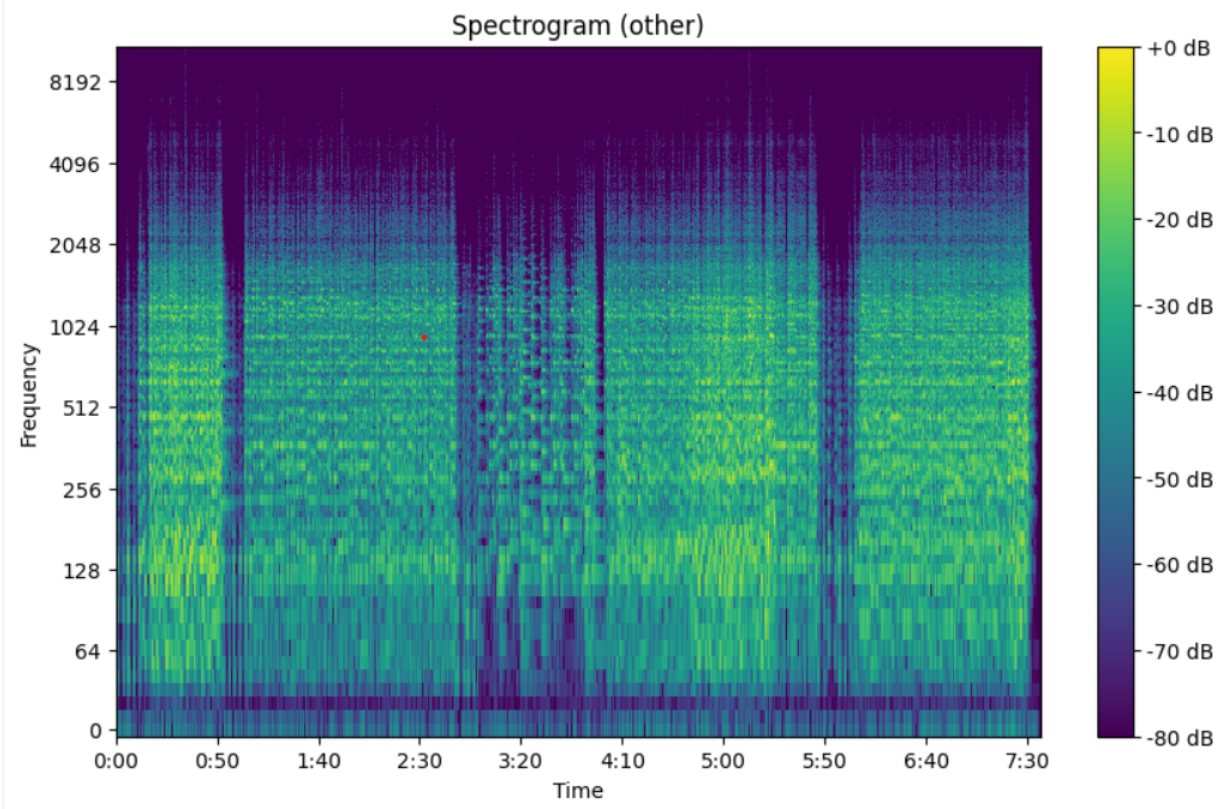


Figure 4.16: Spectrogram-Others

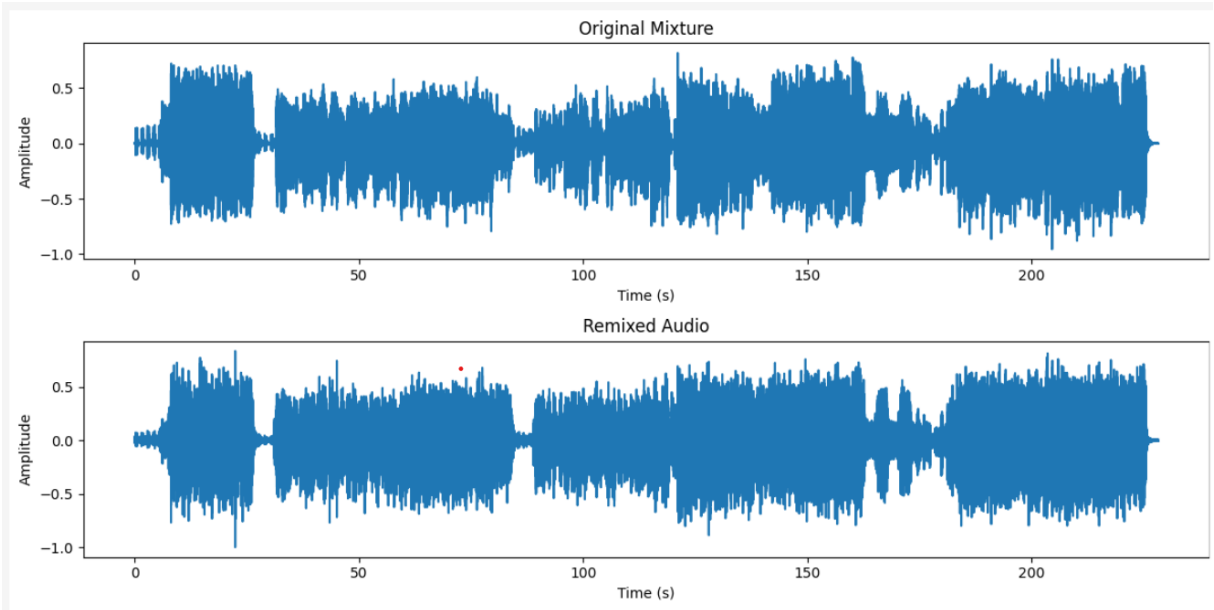


Figure 4.17: Remix Waveform

4.6 Conclusion

The chapter described the results achieved by means of thorough testing and evaluation. The ChordCut project effectively applied deep learning methods for efficient music source separation. Despite issues, the outcomes suggest promising applications in music production, remixing, and audio post-processing.

Chapter 5

Conclusion

The ChordCut project thus managed to adequately address the music source separation issue through its solid system implementation grounded on the deployment of deep learning and spectral analysis methods. Utilized techniques involve STFT as well as neural network structures such as UNet. There was considerable enhancement in the separation and reconstruction of audio components such as vocals and instruments. Modular architecture, comprised of preprocessions, feature extraction, source separation, and steps in output processing, guarantees clean and highly controllable outcomes. The system will be used for music production, remixing, and sound engineering and is intended for commercial audio processing and domestic hobbyist audio track manipulation.

The future work for the project will be extending this dataset so that it encompasses other genres of music and forms of formats. Support for real-time processing will provide them with entry points in real-time audio. In addition to that, the emotion detection feature or genre tagging is going to open up wider opportunities for application. Hardware partners' optimization to prepare the system edge-device ready will have the tendency to increase availability, and more research to reduce computational complexity at the expense of not compromising high accuracy will progressively increase its usability.

References

[1] X. Song, Q. Kong, X. Du, and Y. Wang, "Catnet: Music source separation system with mix-audio augmentation," in *arXiv preprint arXiv:2102.09966*. arXiv, 2021.

[2] T. Sgouros, A. Bousis, and N. Mitianoudis, "An efficient short-time discrete cosine transform and attentive multiresunet framework for music source separation," *IEEE Access*, vol. 10, pp. 119 448–119 459, 2022.

[3] S. Wei, S. Zou, F. Liao, and W. Lang, "A comparison on data augmentation methods based on deep learning for audio classification," in *Journal of Physics: Conference Series*, vol. 1453. IOP Publishing, 2020, p. 012085.

[4] V. Agrawal and S. Karamchandani, "Audio source separation as applied to vocals-accompaniment extraction," in *2022 IEEE 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2022, pp. 150–156.

[5] M. Blaszke and B. Kostek, "Musical instrument identification using deep learning approach," *Sensors*, vol. 22, p. 3033, 2022.

Appendix A: Presentation

CHORDCUT

Guided By:
Ms. Meenu Mathew
Asst.Professor
Dept of Computer Science, RSET

Group Members:
Maria Diya Fiju
Heinz Abraham Koshy
Mathew Jagan Thomas
Juniot Mariyam Thomas

PROBLEM DEFINITION

- The project focused on separating audio components (such as vocals, drums, bass,others) from a mixture and then manipulating these separated elements.

- Music source separation has long been a challenge due to overlapping frequencies in time-frequency points and issues like Fourier phase information loss during reconstruction.

- Current methods such as deep learning-based models (e.g., U-Net, WavUNet) are leveraged for better source separation and remixing.

PURPOSE AND NEED

- The need arises from various applications such as music production, karaoke, remixing,audio-visual post-production, and podcasts.

- Traditional techniques require access to isolated recordings, which limits creative manipulation.

- Our project aimed to create a solution that separates and recombines music tracks, overcoming challenges like phase recovery and remixing artifacts using modern deep learning methods.

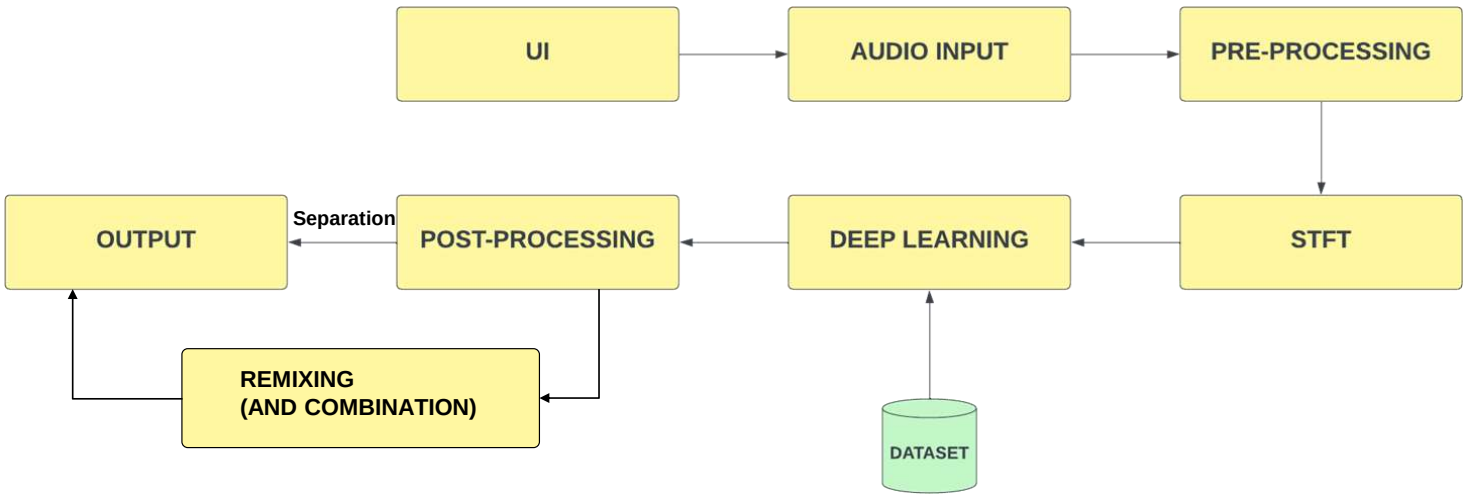
PROJECT OBJECTIVE

- Developed a system capable of separating music sources from an audio mixture.
- Enhanced the accuracy of the separated sources .
- Applied creative manipulation and remixing techniques like pitch-shifting, echo.
- Recombined them into a new , remixed audio track with minimal quality loss.

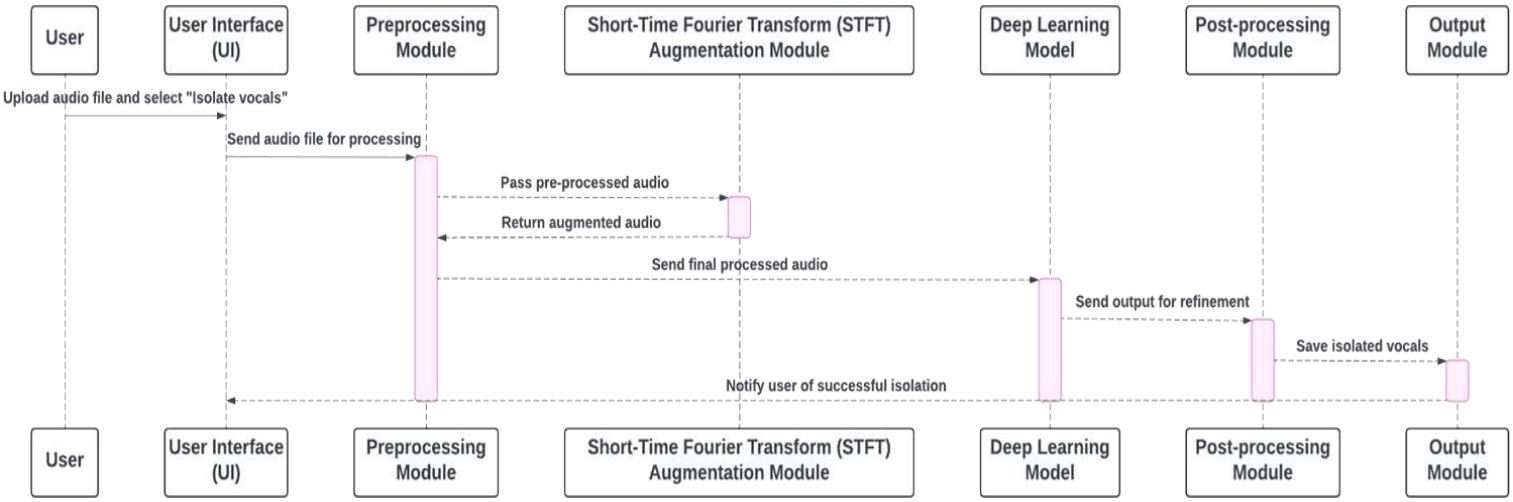
PROPOSED METHOD

- **Input**: A mixture of music sources is fed into the system.
- **Separation**: Deep learning models (U-net) are used to separate individual components, leveraging both spectrogram-based and time-domain methods for enhanced separation.
- **Remixing/Manipulation**: Techniques like pitch shifting, harmonic manipulation, and creative audio effects (pitch-shift, echo, etc.) are applied to the separated sources.
- **Output**: The system recombines the manipulated sources into a new, remixed audio track with minimal loss in quality.

ARCHITECTURE DIAGRAM



SEQUENCE DIAGRAM



MODULES

- **User Interface Module**
- **Deep Learning Module**
- **Post-Processing Module**
- **Remixing & Combination Module**
- **Output Module**

MODULE DETAILS

User Interface Module:

Functionality: Provided a user-friendly interface for uploading audio files, selecting remixing options, and downloading the final output.

- **Work:**
  - Designed and implement a web-based interface using HTML, CSS, and JavaScript.
  - Integrated the interface with the backend modules(Django).
  - Implemented features for uploading audio files, selecting remixing options, and downloading the final output.

Deep Learning Module:

Functionality: Separated the input audio into its constituent components using a deep neural network.

- **Work:**
  - Trained a deep neural network (U-net) on the MUSDB18 dataset.
  - Integrated the trained model into the system.
  - Applied the model to the preprocessed input audio to obtain the separated components.

Post-Processing Module:

- **Functionality:** Refined the separated components to improve quality and reduce artifacts.
- **Work:**
  - Implemented phase correction techniques to align the phases of the separated components.
  - Used spectral modeling or other methods to reduce artifacts.

Remixing Module:

- **Functionality:** Allowed users to manipulate and combine the separated components.
- **Work:**
  - Implemented features for modifying individual components(Echo,Pitch-shift).

Combination Module:

- **Functionality:** Merged the remixed components into a single audio track.
- **Work:**
  - Combined the modified components based on user-specified settings.
  - Ensured proper phase alignment and level balancing.

Output Module:

- **Functionality:** Saved the final remixed audio in .wav format.
- **Work:**
  - Retrieved the output audio from model and download it to downloads folder.

WORK BREAKDOWN

- **Data Preparation:** Collected and prepared the MUSDB18 dataset for training the deep processing module(Maria)
- **Model Training:** Trained the deep neural network on the prepared dataset.(Heinz)
- **Module Implementation:** Implemented the individual modules based on the specified functionalities.(Mathew)
- **Integration:** Integrated the modules into a cohesive system(Collective)
- **User Interface Development:** Created a user-friendly interface for interacting with the system(Juniot)

COMPARISON

30%:Vocals separated from the audio mixture.

50%:Separated of individual instruments.

75%:Integrated a web interface.

100%:Remixing and combination options added.

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements:

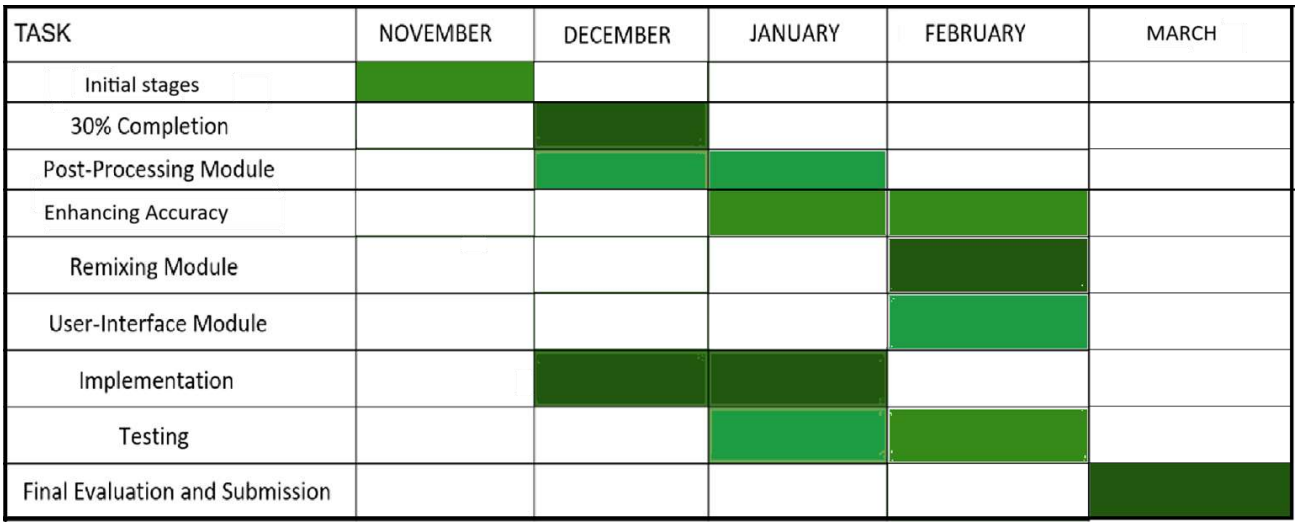
Processor (CPU): Quad-core CPU
Graphics Processing Unit (GPU):NVIDIA GTX1060
Memory (RAM): Minimum 16GB RAM
Storage:20GB of storage for the MusDB18 dataset.
Operating System: Windows 10.

Software Requirements:

Backend Framework: Django
Frontend Technologies: HTML5, CSS3, JavaScript
Audio Processing Libraries:
- librosa: For audio analysis and manipulation.
- pydub: For applying effects like reverb and echo.
- musdb: For accessing and handling the MusDB18 dataset.

Deep Learning Framework:TensorFlow or PyTorch for training and using source separation models (U-Net).
Database: SQLite

GANTT CHART



BUDGET

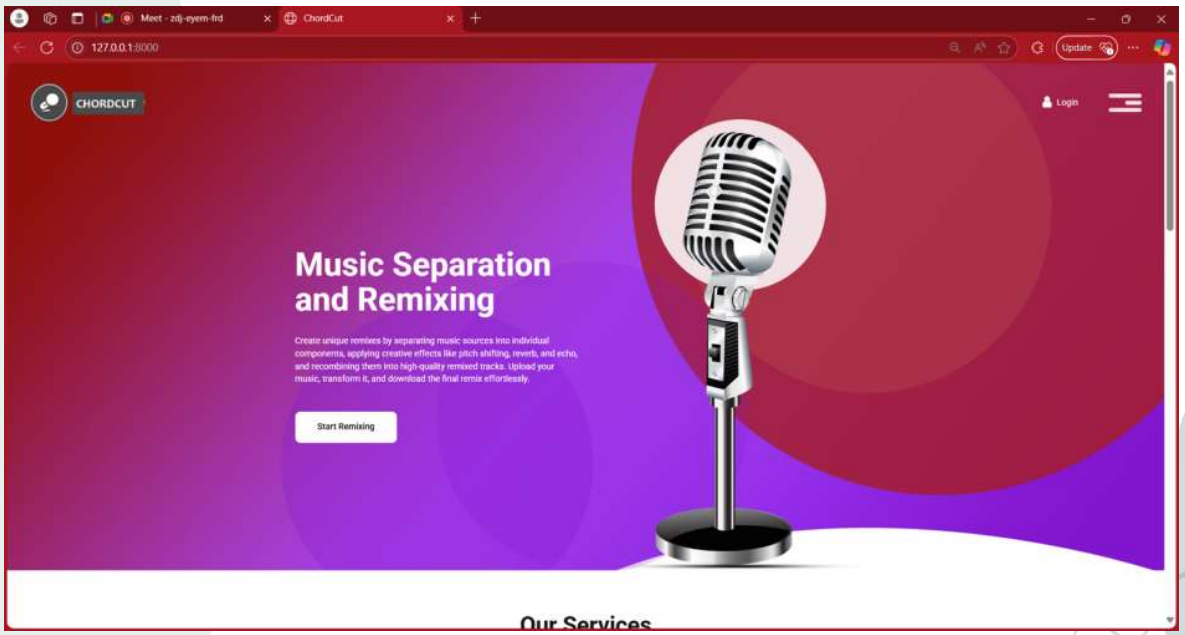
SL.No.	Items	Budget
1	Google Colab Pro	1025
2	Google Colab Pro +	3192
3	Other expenses	1500

EXPECTED OUTPUT

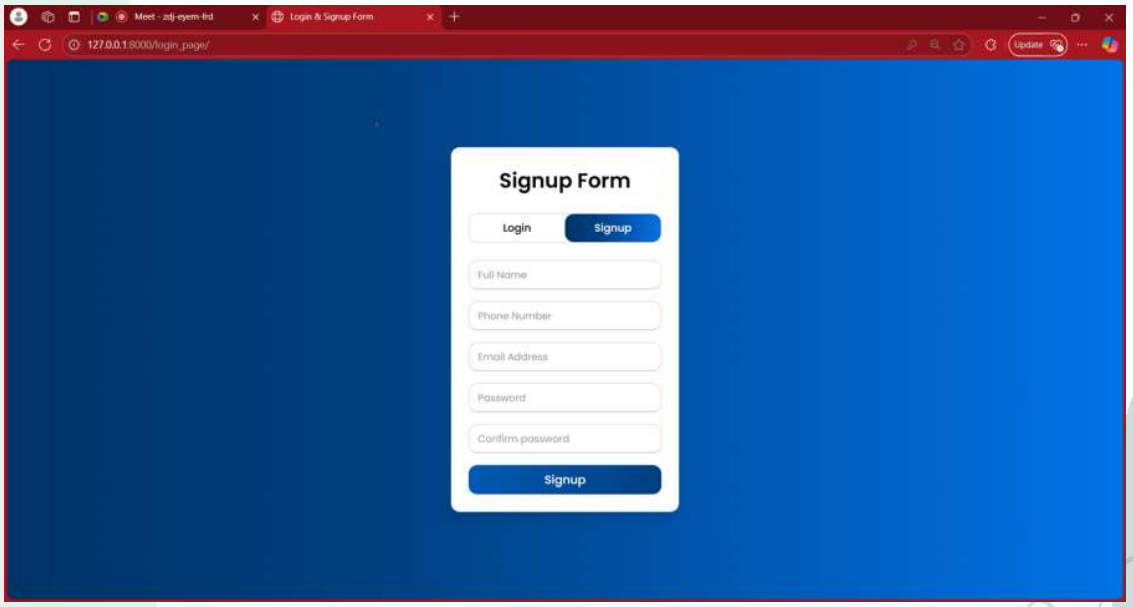
- **Source Separation**: The aim is to achieve state-of-the-art separation with minimal artifacts using advanced architectures like U-net.

- **Remix Capabilities**: Enable users to manipulate the levels and effects of individual sources (vocals, drums, bass) and remix them creatively.
- **Efficiency**: Solutions that balance quality and speed, aiming for models that are computationally efficient without sacrificing performance.
- **Custom Augmentation Techniques**: Leverage data augmentation methods (like Mixed- Audio Data Augmentation) to improve generalization, particularly in handling small datasets, improving source separation and remixing performance.

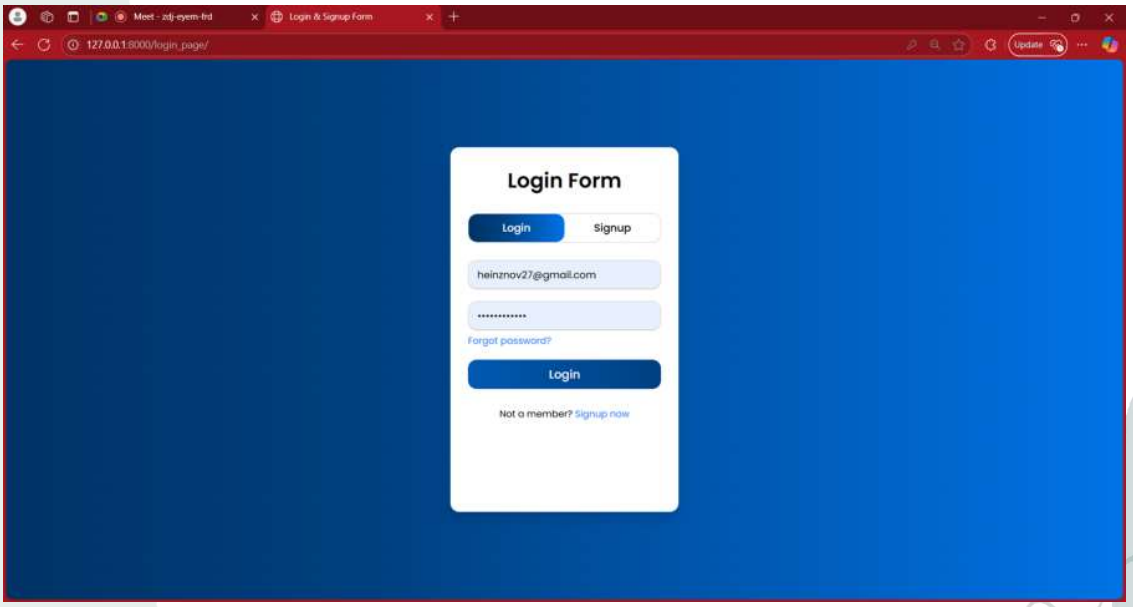
OUTPUTS



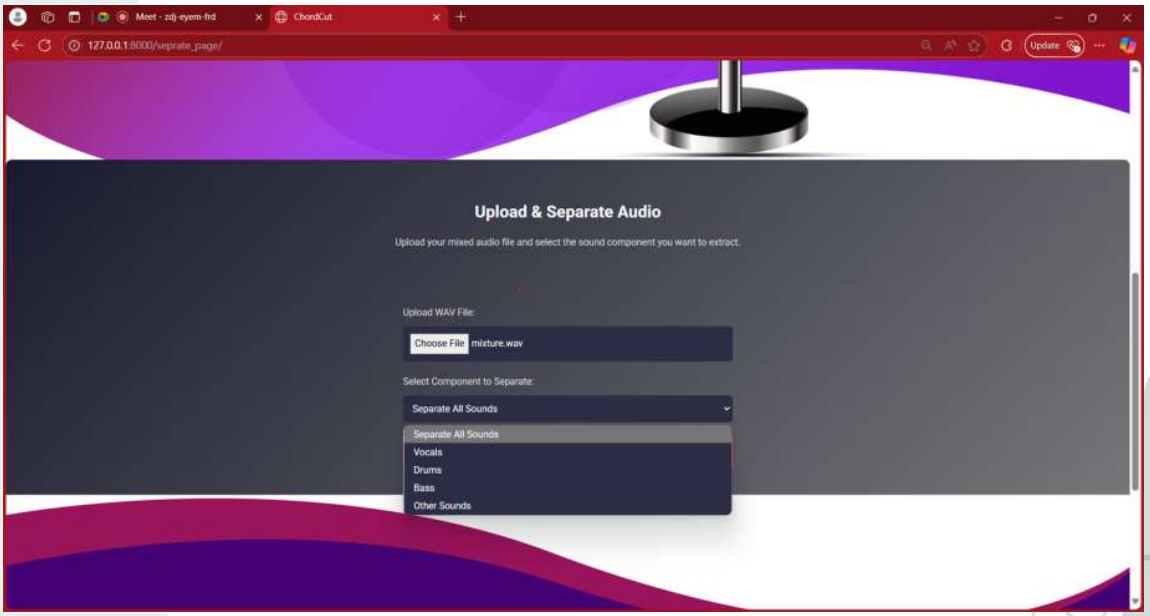
OUTPUTS



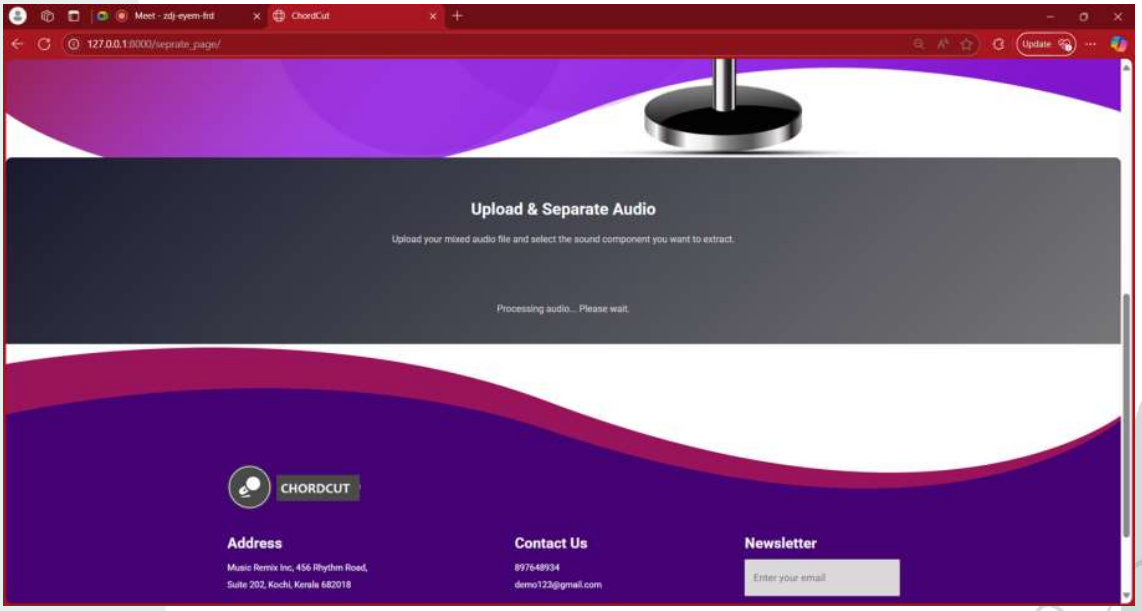
OUTPUTS



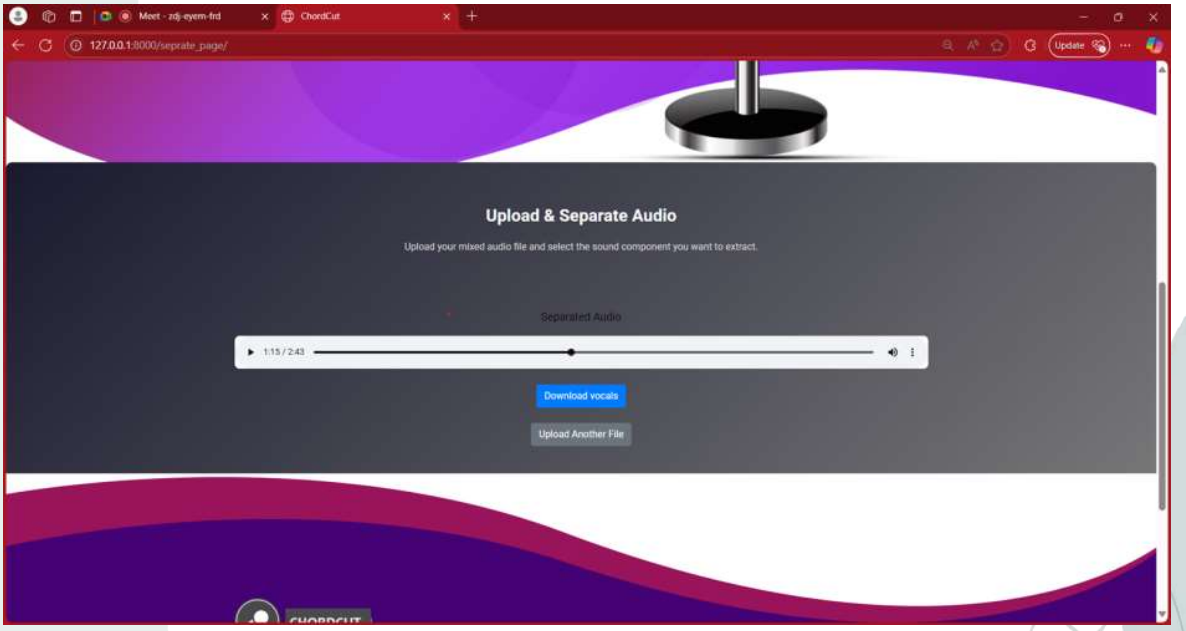
OUTPUTS



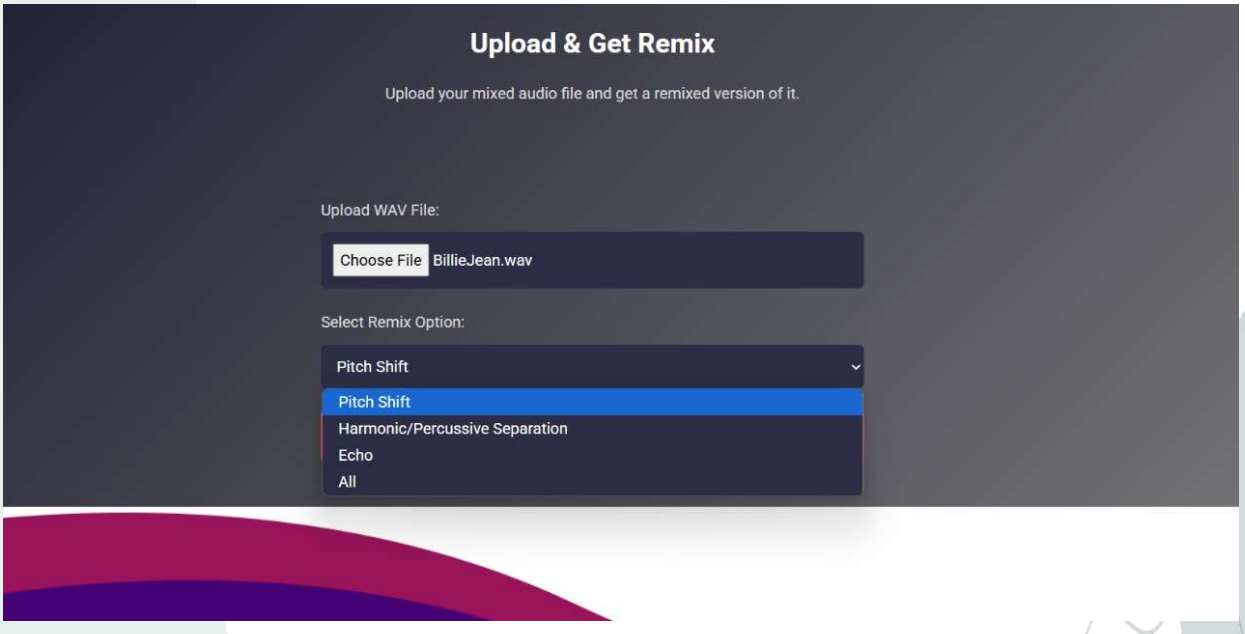
OUTPUTS



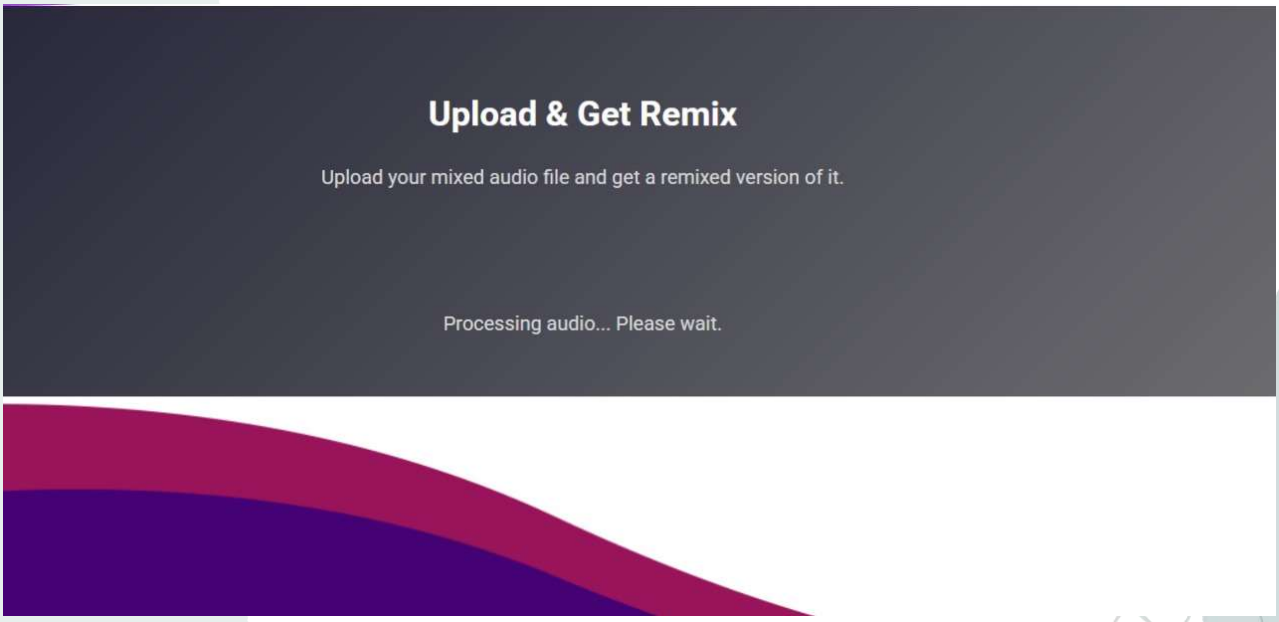
OUTPUTS



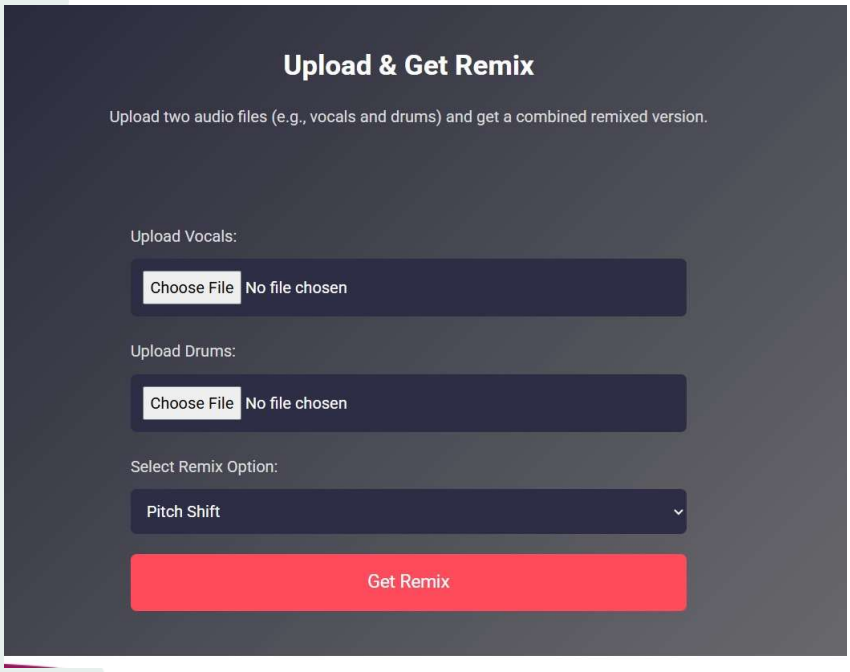
OUTPUTS



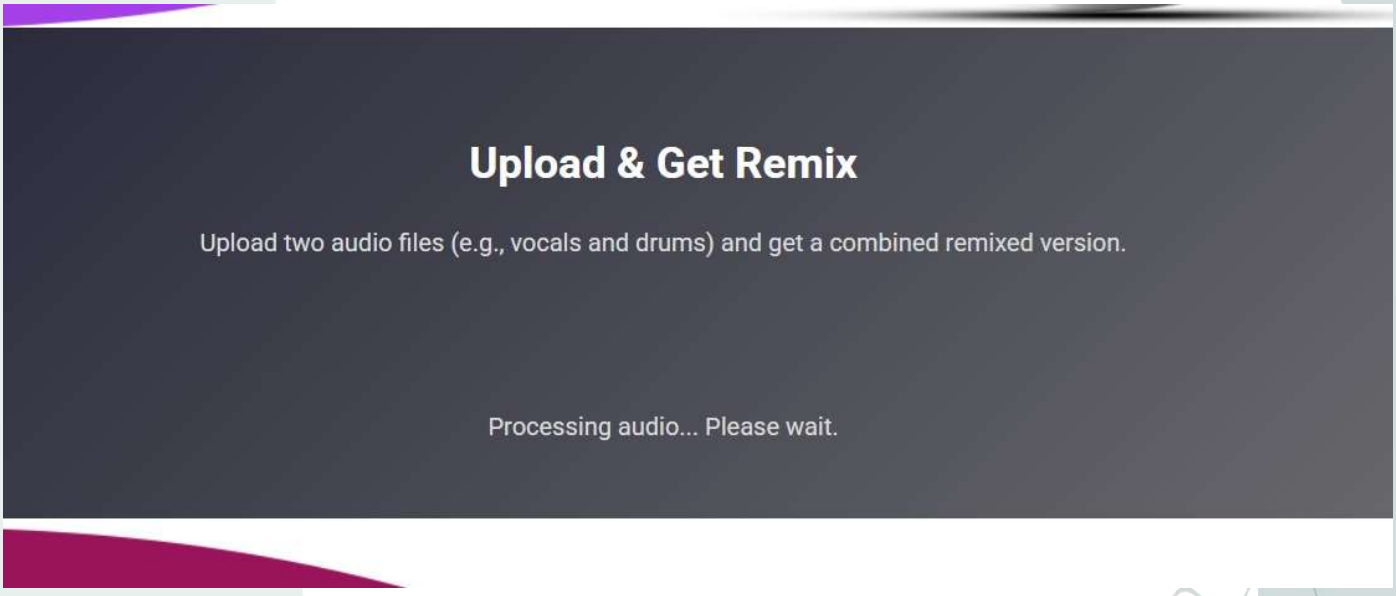
OUTPUTS



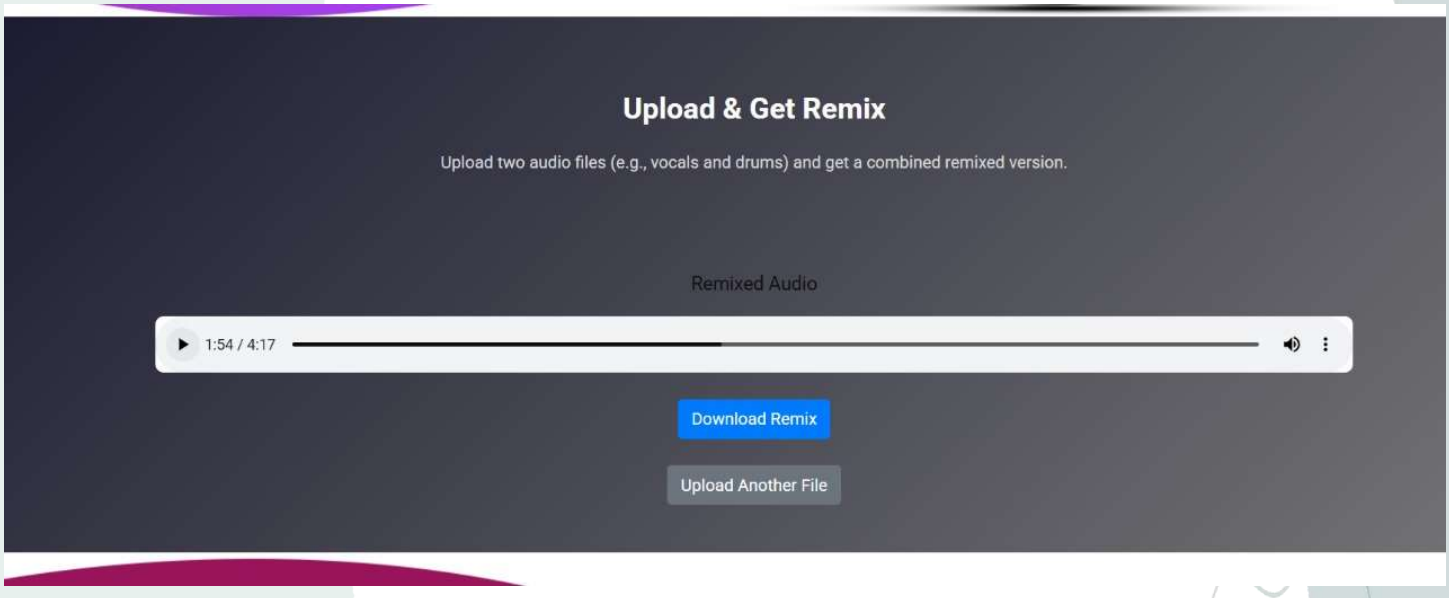
OUTPUTS



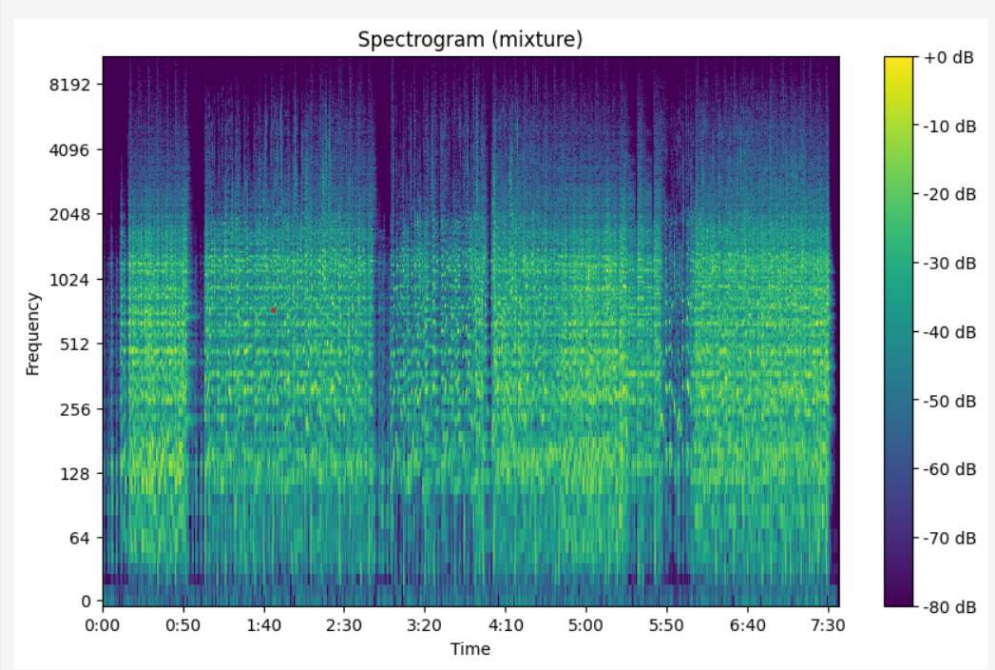
OUTPUTS



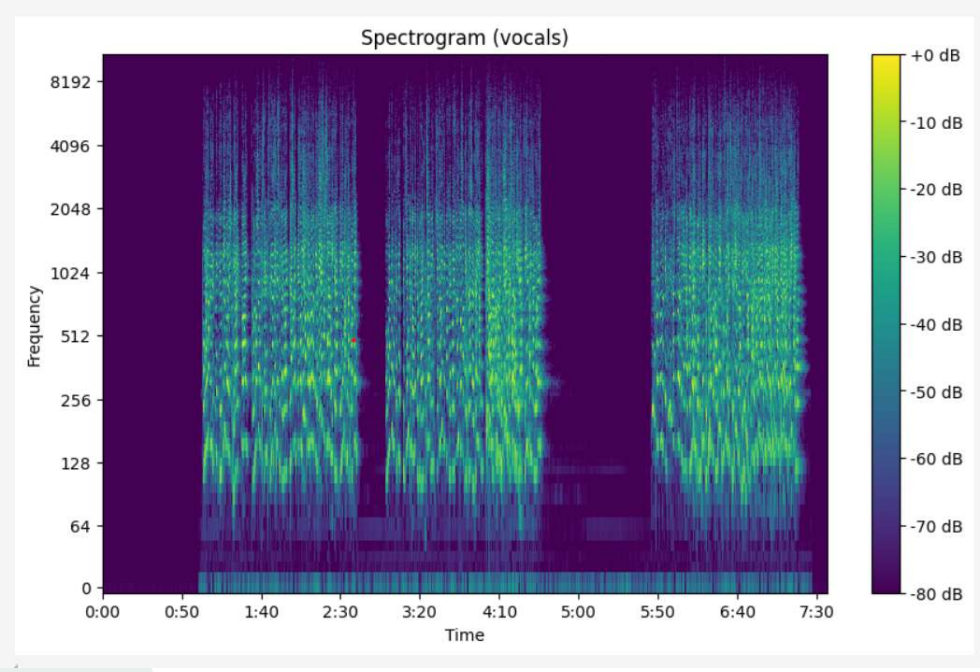
OUTPUTS



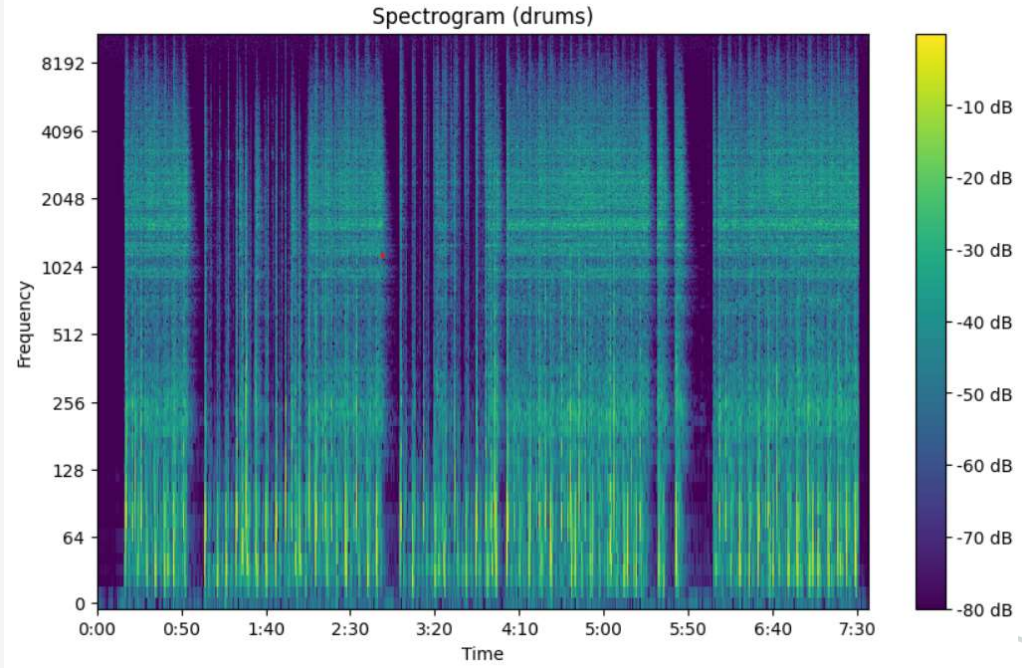
OUTPUTS-Spectrograms



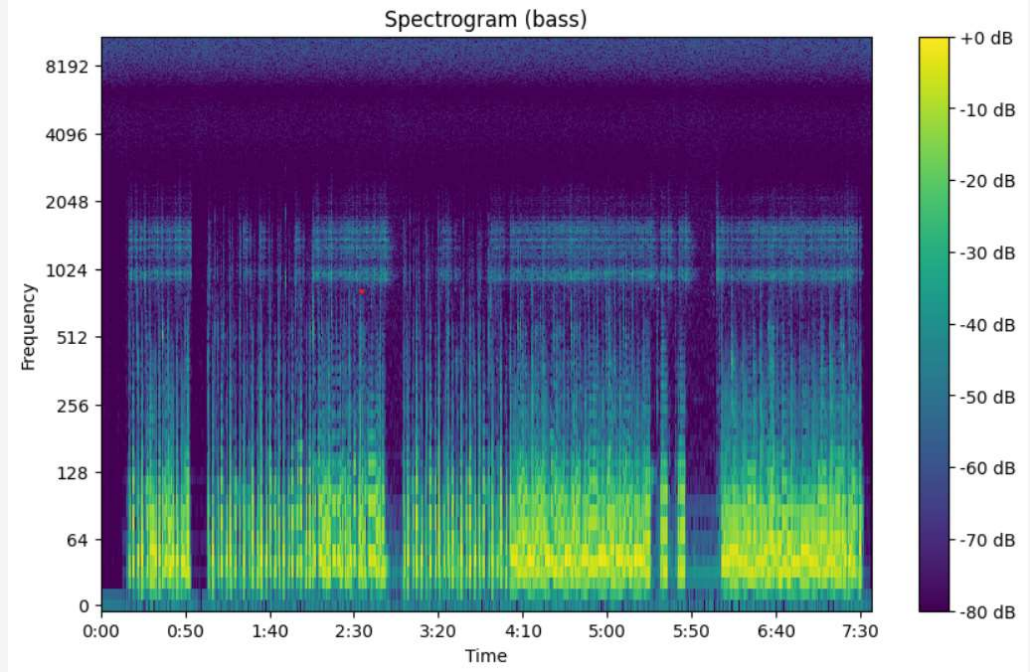
OUTPUTS-Spectrograms



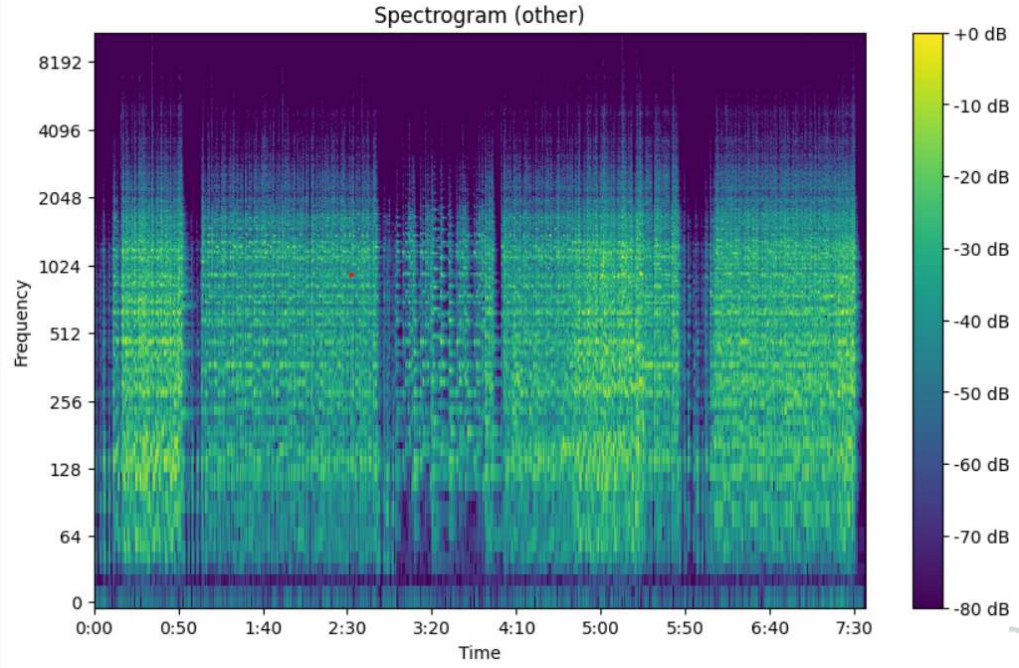
OUTPUTS-Spectrograms



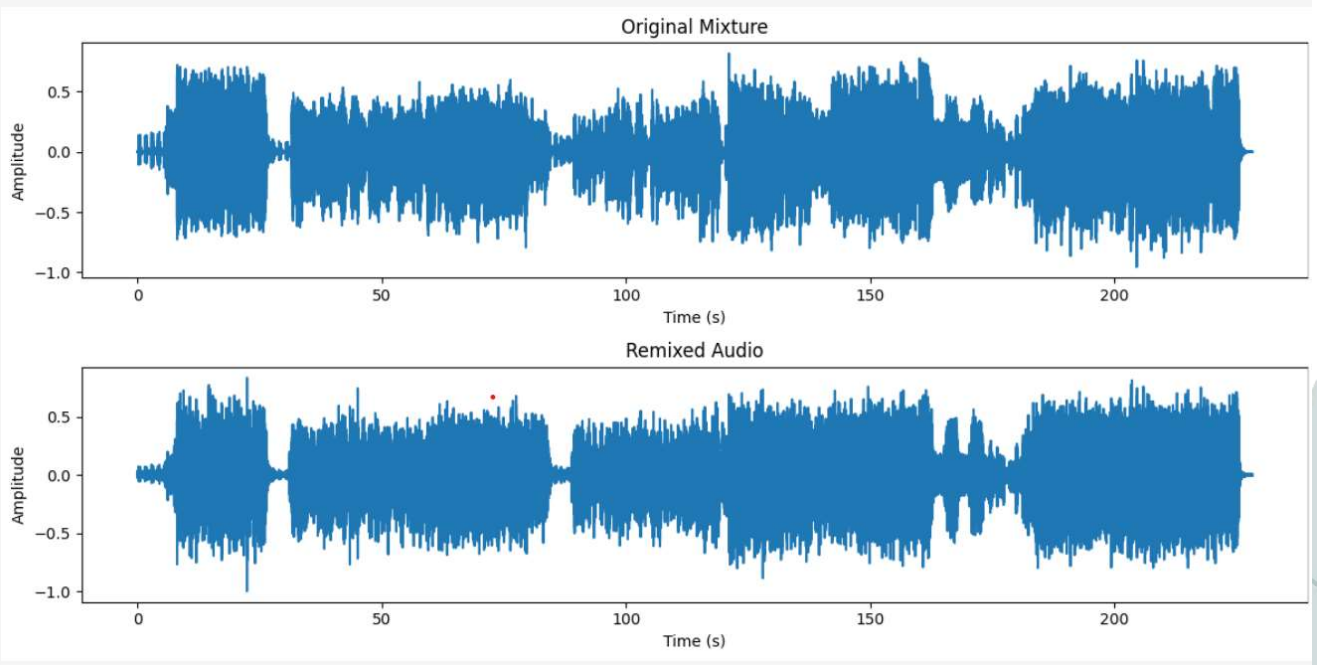
OUTPUTS-Spectrograms



OUTPUTS-Spectrograms



OUTPUTS-Remix waveform



CONCLUSION

This project focused on improving music source separation and manipulation using neural networks. By using techniques like U-Net , along with custom data augmentation methods, we aimed to achieve better sound quality and efficiency.

REFERENCES

M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," Proceedings of the IEEE, vol. 96, no. 4, pp. 668–696, 2008.

Z. Duan, Y. Zhang, C. Zhang, and Z. Shi, "Unsupervised single-channel music source separation by average harmonic structure modeling," IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 4, pp. 766–778, 2008.

Z. Rafii, A. Liutkus, F. St¨oter, 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.

A. Ozerov and C. F´evotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 3, pp. 550–563, 2009.

D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi scale neural network for end-to- end audio source separation," International Society for Music Information Retrieval (ISMIR), 2018.

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Appendix B

Vision: To become a Centre of Excellence in Computer Science & Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Mission: To inspire and nurture students, with up-to-date knowledge in Computer Science & Engineering, Ethics, Team Spirit, Leadership Abilities, Innovation and Creativity to come out with solutions meeting the societal needs.

Program Outcomes:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes:

PSO1: Computer Science Specific Skills: The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills: The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills: The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

CO1: Model and solve real world problems by applying knowledge across domains.

CO2: Develop products, processes, or technologies for sustainable and socially relevant applications.

CO3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks.

CO4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms.

CO5: Identify technology/research gaps and propose innovative/creative solutions.

CO6: Organize and communicate technical and scientific findings effectively in written and oral forms.

Appendix C: CO-PO-PSO Mapping

Appendix C

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO 2	2	2	2		1	3	3	1	1		1	1		2	
CO 3									3	2	2	1			3
CO 4					2			3	2	2	3	2			3
CO 5	2	3	3	1	2							1	3		
CO 6					2			2	2	3	1	1			3

3/2/1: high/medium/low

Figure 5.1: CO-PO and CO-PSO Mapping