Speech Enhancement Using Deep Neural Networks

V. Sudha Rani[1]; Dr. A. N. Satyanrayana[2]; Aroju Santhosh[3]; Maliha[4]; Erravelly Sricharan[5]
[1,2,3,4,5]Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana 501301, India

Abstract:- A comprehensive study is conducted to enhance audio quality in challenging noisy environments, departing from conventional approaches that target specific sound components. This paper focuses on a modified U-Net architecture integrating broader audio features and implementing a probabilistic framework for direct spectral content reconstruction. Multiple variants of this system were rigorously tested across diverse noise levels and reverberation conditions, with performance evaluation conducted using objective metrics such as SDR, signal-to-noise ratio, evaluation of voice, and intelligibility scores.

The paper demonstrates that proposed enhanced U-Net architecture, characterized by strategically designed connections within its structure, consistently outperforms traditional audio enhancement methods across a range of noise scenarios. Notably,the improvements in audio quality were most pronounced in highly reverberant environments, where conventional techniques often struggle to deliver satisfactory results. These results high- light the effectiveness of our novel approach in significantly enhancing audio fidelity and intelligibility, particularly in real- world noisy conditions.

Keywords:- *Audio Enhancement, Noisy Environments, U-Net Architecture, Spectral Content Reconstruction, SDR, SNR.*

I. INTRODUCTION

Real-world speech communication often encounters chal- lenges like background noise, reverberation, and overlapping speakers, which can degrade speech quality and hinder ap- plications such as automatic speech recognition and speaker identification. Speech enhancement techniques are designed toaddress these issues by improving the signal-to-noise ratio andenhancing overall speech clarity.

Recent advancements in speech enhancement leverage the powerful capabilities of deep learning networks, particularlyin handling non-stationary noise and diverse acoustic envi-ronments. These approaches broadly fall into two categories: those that generate filter masks and those that directly mapnoisy input to cleaner speech, known as end-to-end systems. Deep learning models, including large architectures, often use log-power spectra with extended temporal contexts to learn features that effectively represent clean speech. Conversely, smaller auto-encoder models prefer more concise features likeMel-frequency power spectra and short-term Fourier transformspectra computed over short segments or smaller temporal contexts.

State-of-the-art speech enhancement networks directly uti- lize deep architectures to process time-domain signals and generate ideal filter masks based on learned speech represen- tations. Each mask corresponds to a specific target speaker, demonstrating the effectiveness of these approaches.

This paper introduces and explores a novel approach in acoustic signal processing: the variational U-Net architecture for speech enhancement. It proposes integrating a probabilistic bottleneck into the architecture to enhance robustness against out-of-distribution effects such as unknown noise types. The paper includes a performance analysis of the proposed model and various modified versions to validate its efficacy in im- proving speech quality under challenging real-world condi- tions.

II. LITERATURE SURVEY

Existing approach for speech enhancement is Spectral Sub- traction, a classical method operating in the frequency domain. It involves transforming audio signals using techniques like Fast Fourier Transform or Short-Time Fourier Transform to convert them into frequency representations. The core princi- ple of Spectral Subtraction is to estimate the noise spectrum from the noisy audio signal and subtract it from the original $^{\text{Draft}}$spectrum to enhance the speech signal.

Traditionally, Spectral Subtraction required manual estima- tion or selection of the noise profile, which was a cumbersomeand subjective process. This manual intervention limited its adaptability to different noise types and levels, especially when handling rare or non-stationary noises effectively. SpectralSubtraction's performance could degrade significantly when faced with unexpected or infrequent noise patterns.

Wiener Filtering is another method that extends Spectral Subtraction by incorporating statistical signal processing tech- niques to enhance noise reduction while preserving speech intelligibility. Unlike traditional Spectral Subtraction, Wiener Filtering dynamically adjusts the noise estimation based on the SNR of each frequency. It uses an adaptive approach where noise variance is estimated based on the local SNR at each frequency bin. This adaptive nature allows Wiener Filtering to handle varying noise levels across different frequency components better.

Wiener Filtering preserves speech characteristics by using statistical models to estimate optimal filtering parameters. It strikes a balance between noise reduction and the preserva- tion of important speech features like spectral envelope and transient characteristics.

Despite improvements over traditional Spectral Subtraction, Wiener Filtering relies on accurate noise estimation, which can be challenging in real-world noisy environments with complex noise profiles. Moreover, both Spectral Subtraction and Wiener Filtering are considered linear methods and may struggle withnon-linear noise distortions or reverberation artifacts.

III. HARDWARE AND SOFTWARE REQUIREMENTS

- *RAM - 4 GB:* For smooth operation and multitasking, it's recommended to have a minimum of 4 GB of RAM, especially when using resource-intensive applications or handling large files. With 4 GB or more of Random Access Memory, users can enjoy improved system responsiveness, reduced lag during software execution, and enhanced overall computing performance. This capacity is ideal for everyday computing tasks such as document editing, browsing, and storing data.

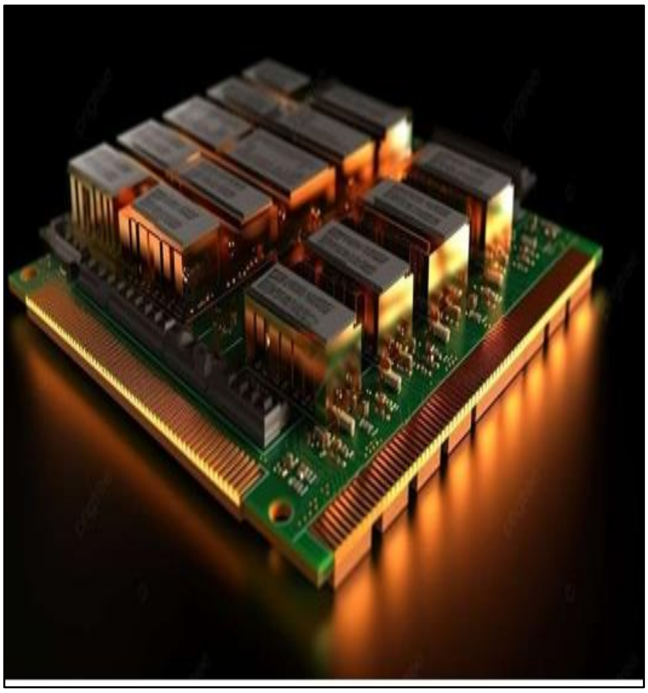


Fig 1: RAM

- Processor Intel Core i5: The Intel i5, or its equivalent from other manufacturers, strikes a good balance between performance and efficiency for various computing needs. Equipped with multiple cores and threads, an Intel Core i5 processor ensures smooth multitasking, faster processing speeds for applications, and overall better responsiveness. This level of processing power is suitable for demanding tasks like photo and video editing, programming, gaming, and running virtual machines without significant performance issues.

Fig 2: Processor

- Storage - 256 GB SSD: To benefit from faster data access, quicker boot times, and improved system performance, it's recommended to have a minimum of 256 GB of Solid State Drive Storage or higher. SSDs offer significant advantages over traditional Hard Disk Drives in terms of speed, reliability, and energy efficiency. With a 256 GB SSD or larger, users can store their operating system, frequently used applications, and data files, ensuring swift access to information and seamless workflow management.

Fig 3: Solid State Drive

- *Stable Network Connection:* A stable network connection is crucial for uninterrupted internet access, smooth online collaboration, reliable streaming, and efficient data transfers. Whether using a wired Ethernet connection or a wireless Wi-Fi connection, stability ensures consistent bandwidth, low latency, and minimal disruptions during online activities. A stable network connection is particularly essential for tasks like video conferencing, online gaming, cloud storage access, and downloading/uploading large files. Employing quality networking hardware and optimizing network settings can help maintain a stable and reliable connection for productive computing experiences.

- *Operating System: Windows, MacOS:* The system requirements include compatibility with popular operating sys- tems like Windows and macOS. Windows is widely utilized across various devices, offering a familiar user interface, extensive software compatibility, and robust system securityfeatures.

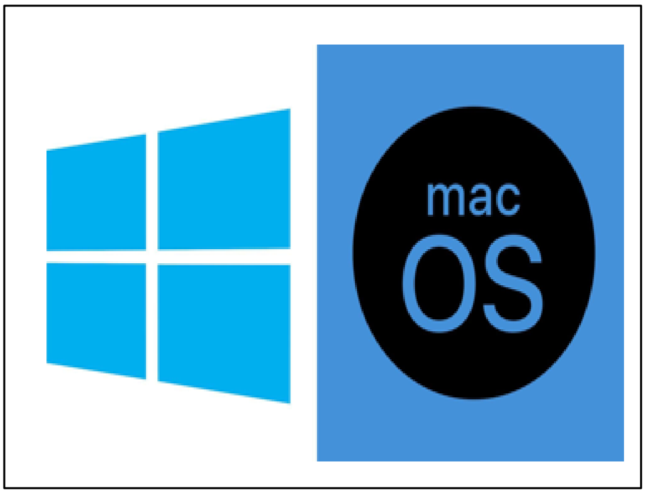


Fig. 4: Operating System

macOS, known for its intuitive design and seamless inte- gration with Apple hardware, excels in creative tasks such as graphics design, video editing, and music production. Supporting both Windows and macOS ensures flexibility and accessibility for users across different platforms, enabling them to leverage the software's capabilities regardless of their preferred operating system.

- Programming Language: Python 3.7 is specified as the required programming language, providing a versatile and powerful platform for software development, data analysis, machine learning, and scientific computing. Python's syntax simplicity, readability, and extensive library ecosystem make it a preferred choice for both beginners and experienced programmers. Version 3.7 offers language enhancements, per- formance optimizations, and compatibility with a wide range of third-party libraries and frameworks, ensuring users can leverage the latest features and tools necessary for efficient software development and data analysis tasks.
- Execution Environment: Python IDLE, Google Colab, Jupyter*: The execution environment requirements encompass Python Integrated Development Environment, Google Colab, and Jupyter Notebook. These environments provide functional platforms for writing, testing, and debugging Python code, making them suitable for beginners and quick script development tasks. Google Colab offers a cloud-based plat- form with integrated Python support, providing access to GPU/TPU resources for machine learning experiments, collaborative coding, and seamless integration with Google Drive for data storage and sharing. Jupyter Notebook, with its interactive and notebook-style interface, enables users to create programs containing livecode, symbolization, and explanatory text, fostering collaborative and reproducible computing environment. Supporting these execution environ- ments ensures versatility, collaborative capabilities, and access to cloud resources for efficient Python programming and data analysis workflows.

IV. BLOCK DIAGRAM AND FLOWCHART

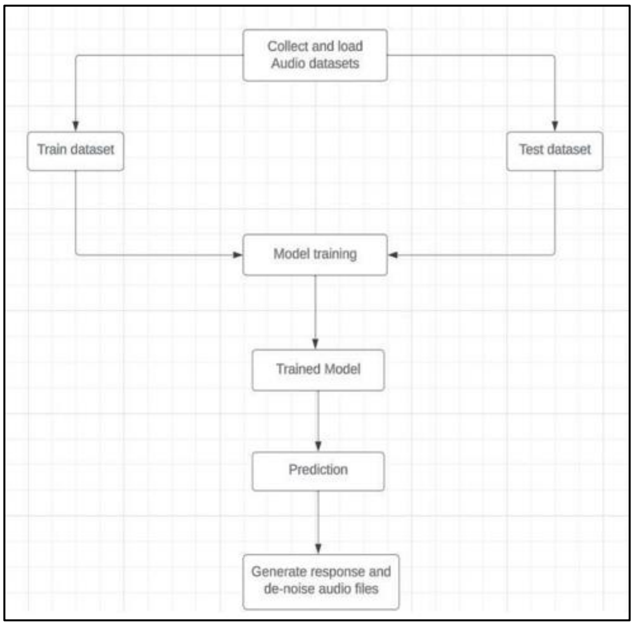

Collect and load Audio datasets
Train dataset
Test dataset
Model training
Trained Model
Prediction
Generate response and de-noise audio files

Fig 5: Block Diagram

The system architecture for the audio denoising project begins with a user-friendly input interface created using Streamlit, allowing users to upload noisy audio files easily. Upon submission, the backend processes the uploaded files by handling them and generating spectrograms to visualize the audio's frequency content over time. The UNET algorithm, a deep learning model, analyzes these spectrograms to predict the noise present in the audio. Using advanced deep learning techniques, the predicted noise is then effectively removed from the original audio signal.After the denoising process, users can access the cleaned audio through the Streamlit interface, where they can conveniently listen to and download the enhanced version. This seamless workflow provides users with an intuitive platform to enhance the quality of their audio recordings.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24APR2694

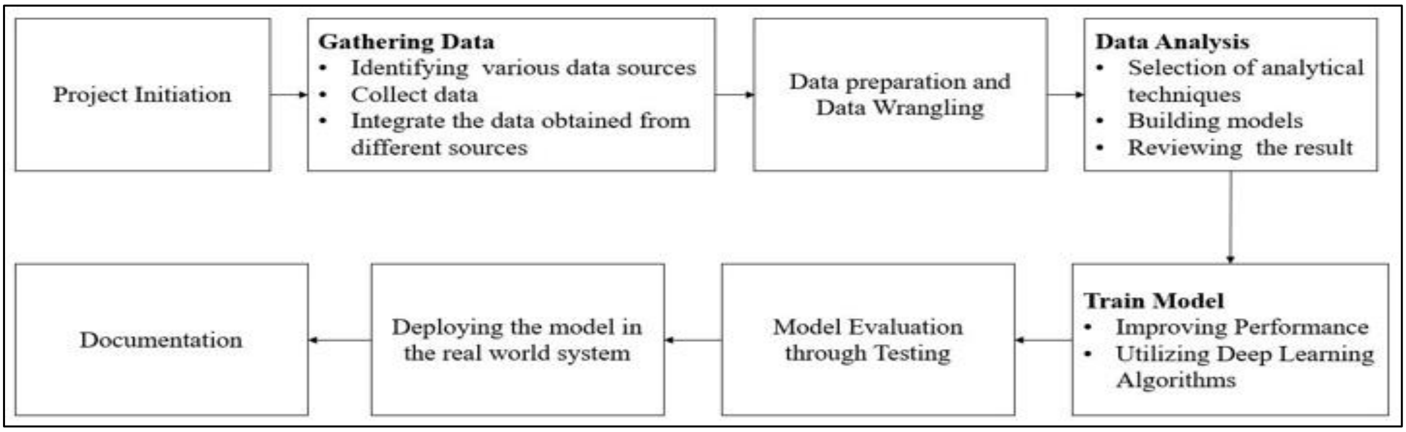

Project Initiation
Gathering Data
Identifying various data sources
Collect data
Integrate the data obtained from different sources
Data preparation and Data Wrangling
Data Analysis
Selection of analytical techniques
Building models
Reviewing the result
Train Model
Improving Performance
Utilizing Deep Learning Algorithms
Model Evaluation through Testing
Deploying the model in the real world system
Documentation

Fig 6: Flow Chart

V. U-NET ARCHITECTURE AND ITS APPLICATION

The U-Net architecture, initially developed for image segmentation, has gained significant popularity and applicability across diverse domains due to its effectiveness in handling complex data structures and tasks. Its distinctive ”U” shape comprises an encoding path that progressively reduces spatial dimensions using convolutional and pooling layers, followed by a symmetric decoding path that upsamples and reconstructs the segmented output. In this architecture,the encoding layers serve as feature extractors, capturing hierarchical representations of input data. The bottleneck layer acts as a bridge between the encoding and decoding sections, refining features and facilitating information flow. The decoding layers then reconstruct the segmented output by upsampling and merging features from earlier stages, enablingprecise segmentation and localization of objects within the input data. Key to the U-Net’s success are the skip connections,which directly link corresponding encoding and decoding layers. These connections preserve spatial information and facilitate gradient flow during training, leading to enhanced performance in tasks such as image segmentation, denoising, and reconstruction. The versatility, robustness, and ability to perform well with limited data make the U-Net architecturea valuable asset in various machine learning and computer vision applications.

The data preparation phase involves extracting audio signalsfrom specified directories and converting them into NumPy arrays. These arrays are then transformed into spectrograms, capturing both amplitude and phase information essential for subsequent processing. During model development and training, (Fig. 5.1) is constructed using the pre- processed data. This model comprises encoding, bottleneck, and decoding layers optimized for efficient segmentation and denoising tasks. The encoding layers extract and downsample hierarchical features, while bottleneck layers refine representa- tions through convolutions and regularization techniques like dropout. Decoder layers then reconstruct segmented output by up sampling and combining feature maps.

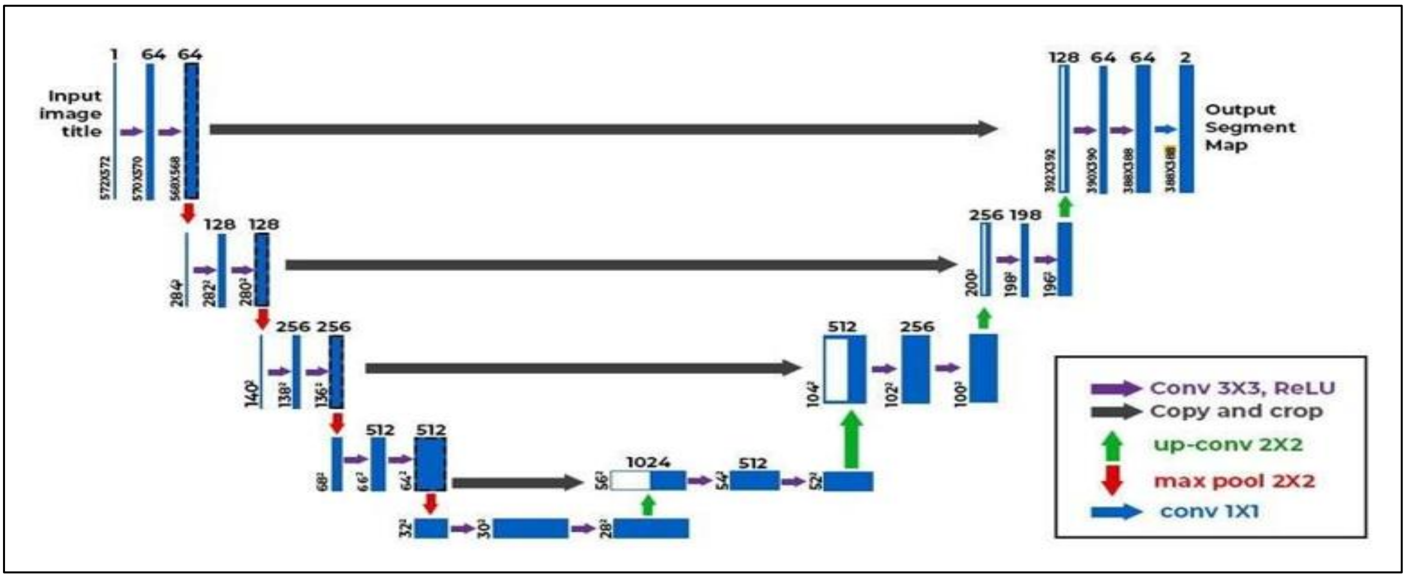


Fig 7: U-NET Architecture

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24APR2694

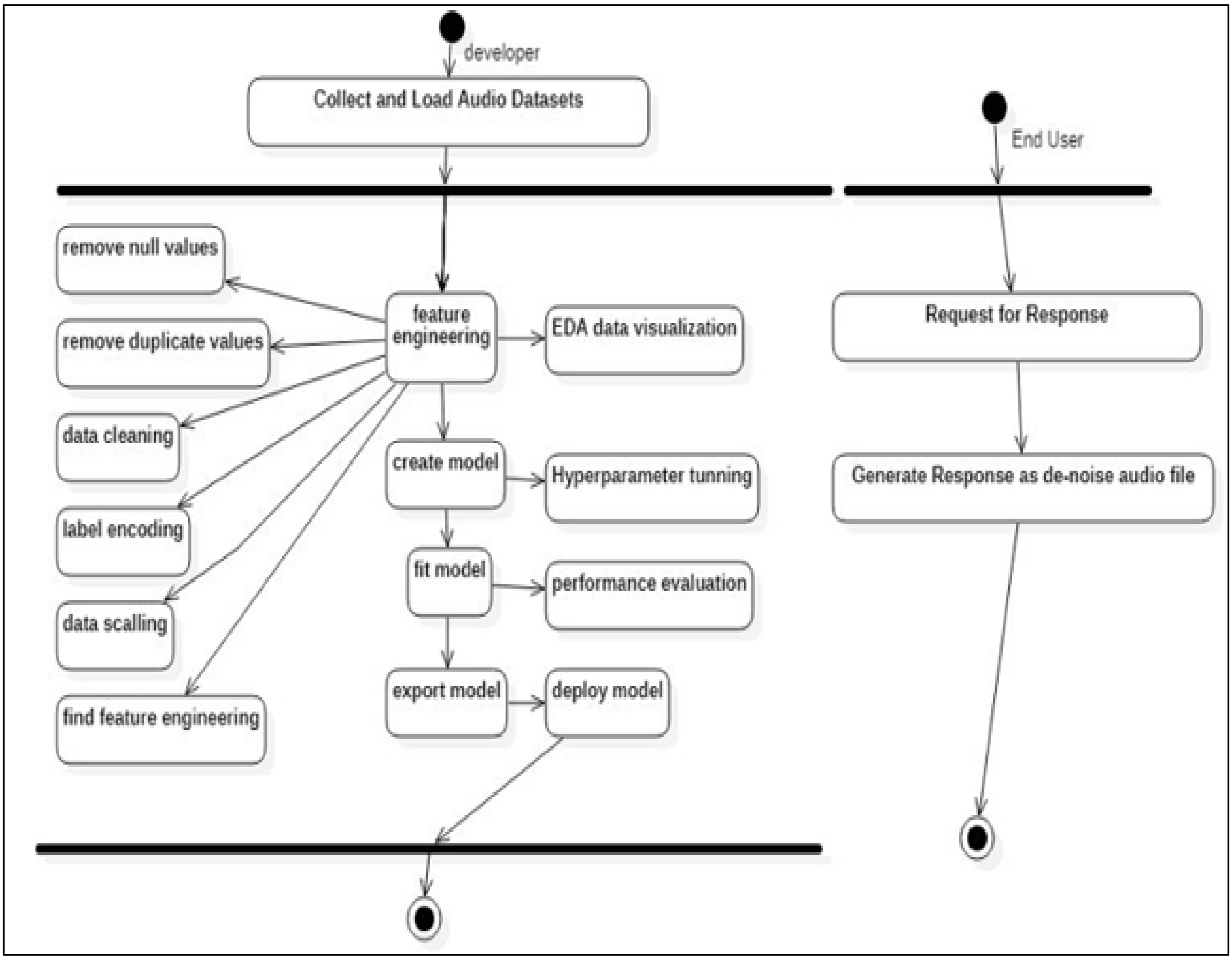


Fig 8: Working

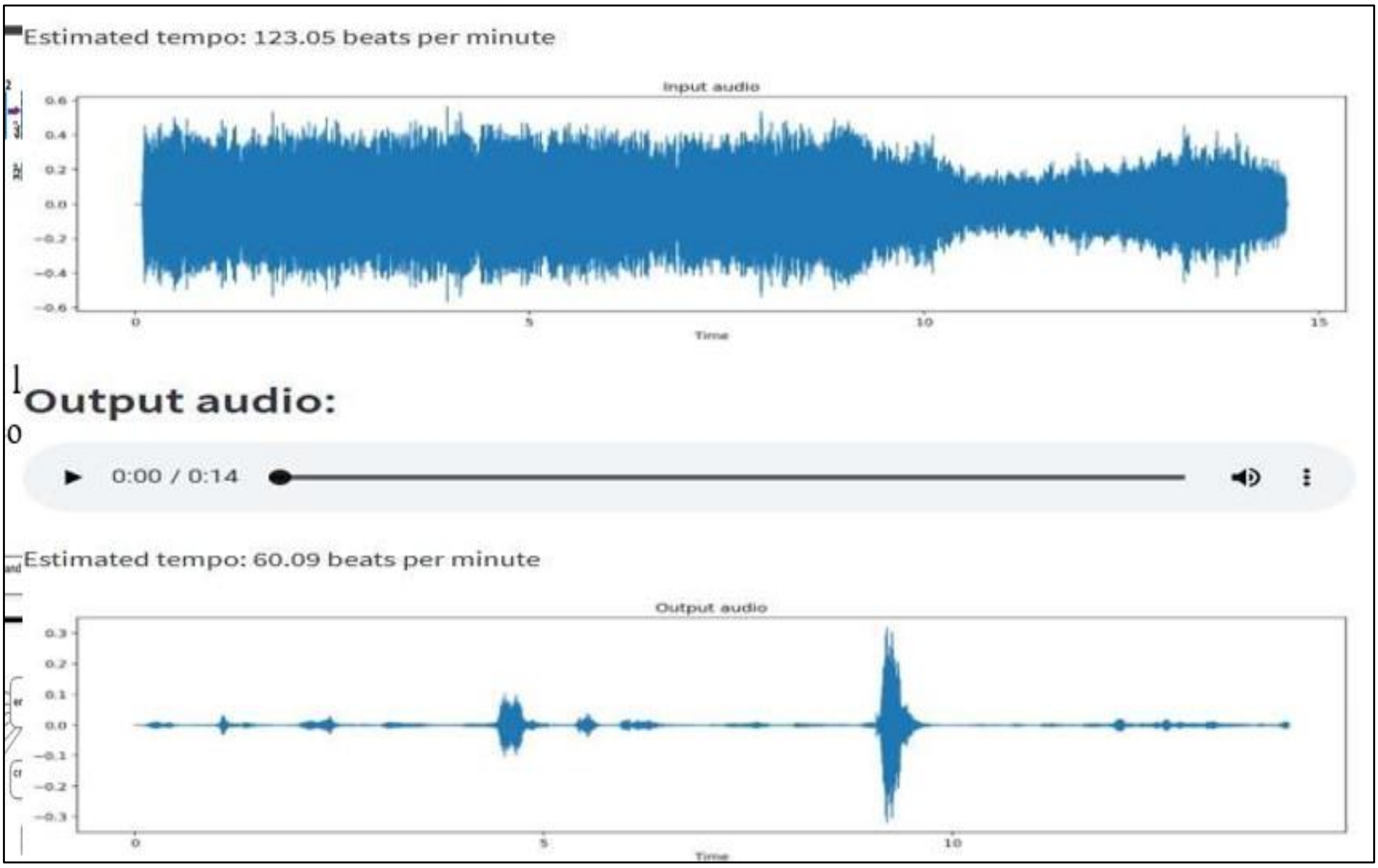
Training utilizes the prepared data to optimize the model for accurate denoising predictions. In the prediction phase, spectrograms from audio files are fed into the pretrained U-Net, which outputs denoised spectrograms, effectively re- moving noise from the audio signals. Additionally, a user- friendly interface is developed to allow users to upload audio files for denoising and listen to the cleaned audio output.This streamlined process provides an intuitive platform for enhancing audio quality.

VI. RESULTS

In the first scenario (Fig. 6.1.), we observe a waveform representing a politician's speech, likely contaminated with background noise such as crowd chatter or microphone interference. The result here is to isolate the politician's voice while reducing or removing the unwanted background noise, resulting in a clearer and more intelligible speech output.

In the second scenario (Fig. 6.2.), we see a waveform depicting vehicle sounds such as engine rumbling and tire noise. Vehicle audio recordings often contain significant noise,especially in busy traffic or industrial environments. The goal of speech enhancement in this case is to extract the desired vehicle sounds for analysis or monitoring purposes, filtering out undesirable noise to improve overall audio quality.

The third scenario (Fig. 6.3.) shows a waveform of random audio, which may include a mixture of music, ambient noise, or various environmental sounds. Speech enhancement tech- niques can be applied to such diverse audio sources to enhance specific elements or separate different audio components, ultimately improving audio clarity and quality. This process involves isolating desired sounds while minimizing unwanted noise, resulting in enhanced audio suitable for various appli- cations.



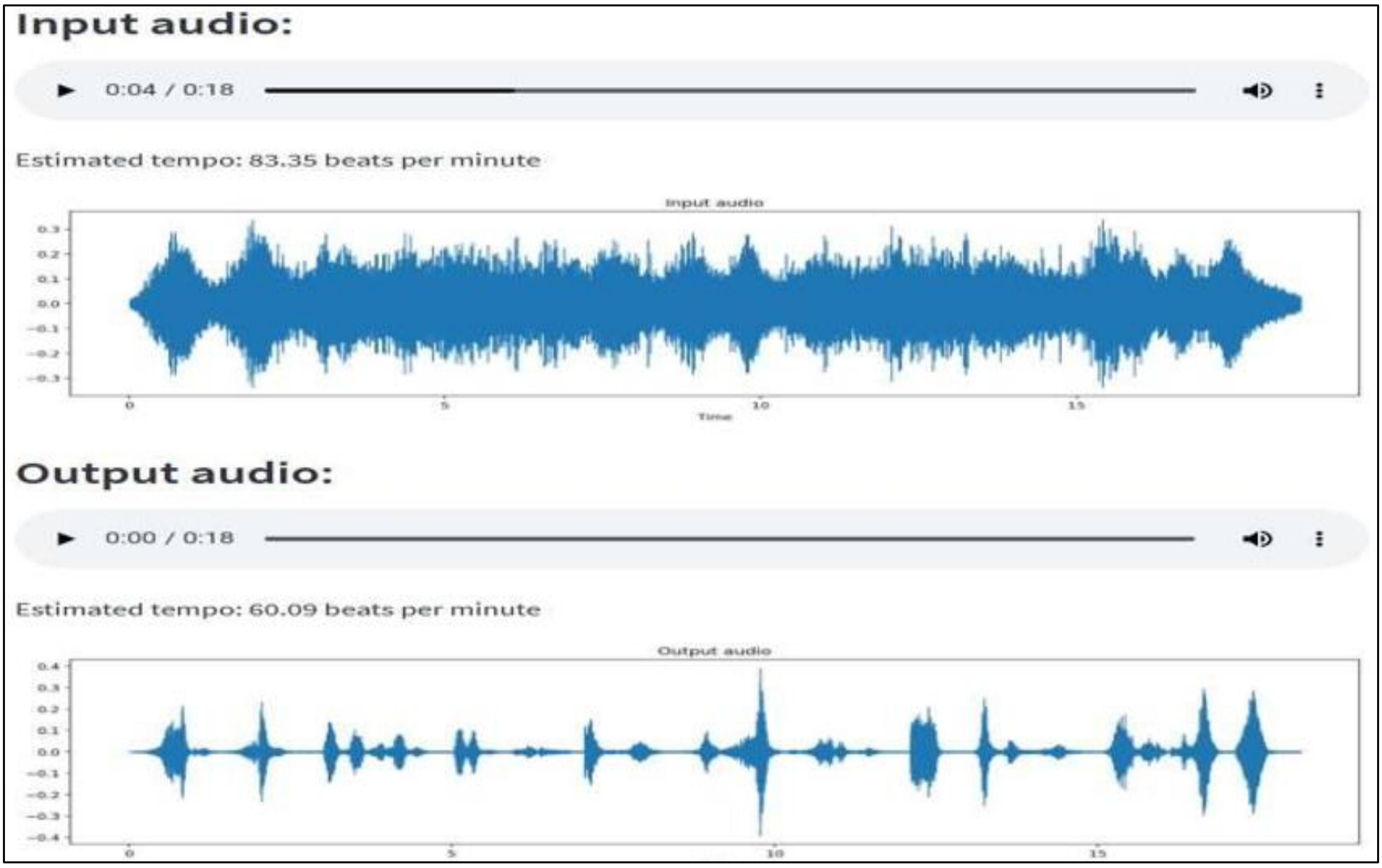
Estimated tempo: 123.05 beats per minute

Output audio:

▶ 0:00 / 0:14

Estimated tempo: 60.09 beats per minute

Fig 9: Enhanced Audio of a Politician’s Speech



Input audio:

▶ 0:04 / 0:18

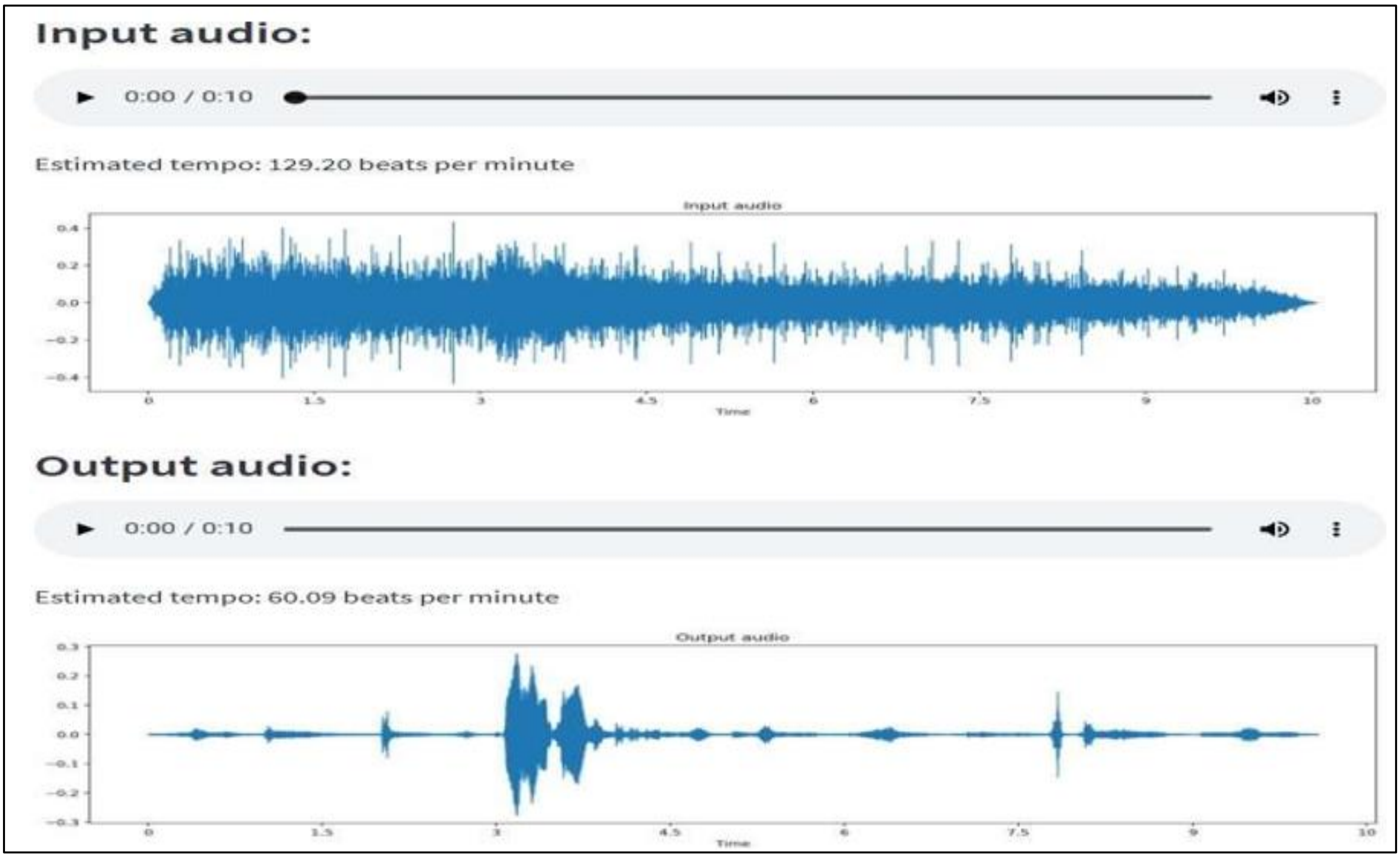
Estimated tempo: 83.35 beats per minute

Output audio:

▶ 0:00 / 0:18

Estimated tempo: 60.09 beats per minute

Fig 10: Enhanced Audio of a Vehicle Sounds



Input audio:

Estimated tempo: 129.20 beats per minute

Output audio:

Estimated tempo: 60.09 beats per minute

Fig 11: Enhanced Audio

VII. CONCLUSION

This study introduces a novel U-Net architecture featuring a variational bottleneck for speech enhancement, a concept not previously explored in the context of acoustic signals but with established variants in image segmentation research. A comparison between the variational U-Net and its modified versions, using data and benchmarks from the deep noise separation challenge, highlighted the importance of the vari- ational bottleneck in adapting from non-reverberant training to reverberant test scenarios. The inclusion of lateral U-Net connections and the use of dilated convolutions were also validated through experimental results.

The superior performance of the variational model confirms our hypothesis, attributing its success to the generative Gaussian model within the variational bottleneck. This model assists in guiding reconstruction towards output data that closely aligns with the training distribution. Conversely, purely deterministic models like the standard non-variational U-Net architecture exhibited inferior performance in scenarios of train/test mismatch.

Future research efforts will focus on exploring applications involving distribution shifts between the model and observa- tion. For example, dereverberation processes could potentially benefit from a variational approach to source reconstruction, highlighting the broader utility and potential advancements in utilizing variational architectures in audio signal processing.

REFERENCES

[1]. F. Rund, V. Vencovsky, and M. Semansk ´ y, “An evalu-ation of click detection algorithms against the results of listening tests,” J. Audio Eng. Soc., vol. 69, no. 7/8, pp. 586–593, July/Aug. 2021.

[2]. H. T. de Carvalho, F-R. Avila, and L. W. P. Biscainho, “Bayesian restoration of audio degraded by low frequency pulses modeled via Gaussian process,” IEEE J. Selected Topics Signal Process., vol. 15, no. 1, pp. 90–103, Oct. 2021.

[3]. J. Berger, R. R. Coifman, and M. J. Goldberg, “Removing noise from music using local trigonometric bases and wavelet packets,” J. Audio Eng. Soc., vol. 42, no. 10, pp. 808–818, Oct. 1994.

[4]. P. A. A. Esquef, “Audio restoration,” in Handbook of Signal Processing in Acoustics, pp. 773–784. Springer, New York, NY, USA, 2008.

[5]. S. Boll, “Suppression of acoustic noise in speech using spectral subtrac- tion,” IEEE Trans. Acoust. Speech Signal Process., vol. 27, no. 2, pp. 113–120, Apr. 1979.

[6]. S. J. Godsill and P. J. W. Rayner, Digital Audio Restoration - A Statistical Model Based Approach, Springer, 1998.

[7]. Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log- spectral amplitude estimator,” IEEE Trans. Acoust. Speech Signal Process., vol. 33, no. 2, pp. 443–445, Apr. 1985