International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Sound Noise Reduction Based on Deep Neural Network

Amtul Shanaz[1],C Sakshi[2],P Sharon Rose[3], E Sravanthi[4],T Sreeja[5]

[1]Assistant Professor, Department of Computer Science and Engineering, Bhoj Reddy Engineering College for Women, Telangana, India.

[2–4]Students, Department of Computer Science and Engineering, Bhoj Reddy Engineering College for Women, Telangana, India.

sakshiravte2005@gmail.com, psharonrose1307@gmail.com

Abstract

We investigate the viability of a variational U-Net architecture for denoising of single-channel audio data. Deep network speech enhancement systems commonly aim to estimate filter masks, or opt to work on the waveform signal, potentially neglecting relationships across higher dimensional Spectro temporal features. We study the adoption of a probabilistic bottleneck into the classic U-Net architecture for direct spectral reconstruction. Evaluation of several ablation network variants is carried out using signal-to-distortion ratio and perceptual measures, on audio data that includes known and unknown noise types as well as reverberation. Our experiments show that the residual (skip) connections in the proposed system are a prerequisite for successful spectral reconstruction, i.e., without filter mask estimation. Results show, on average, an advantage of the proposed variational U-Net architecture over its classic, nonvariational version in signal enhancement performance under reverberant conditions. To improved suppression of impulsive noise sources with the variational U-Net.

Keywords: *Variational U-Net, Speech Enhancement, Audio Denoising, Spectro-temporal Features, Signal-to-Distortion Ratio (SDR), Probabilistic Modeling, Spectral Reconstruction, Impulsive Noise Suppression.*

Introduction

Autoregressive neural synthesis systems are based on the idea that the speech signal's probability distribution can be formulated as a scalar autoregressive structure, where the probability of each speech sample st is conditioned on previous samples and a set of conditioning features (spectral information, pitch, etc. The paradigm was first introduced for text-to-speech using the Wave Net architecture. Soon thereafter Wave Net was shown beneficial also for low bit rate speech coding which was the first coder using neural generative synthesis. Another example of a code using Wave Net as synthesis generation. Since then other autoregressive generators with lower complexity have been introduced, and codes based on such are for example based on Sample RNN based on Wave RNN. However, the reproduction of real-world speech signals by generative models is still a challenge. Coding real-world speech signals with a neural coder requires solving a number of problems simultaneously. The difficulty of coding noisy signals can perhaps be explained by the signal structure, where the signal to be coded is the sum of a clean speech signal and an interfering signal. The autoregressive architecture is a good match for the structure of speech, but the addition of a second signal removes this match. This suggests that to reproduce both the speech and the additive signal with high quality a significantly larger model may be needed. Resisting the urge to increase the model size we instead performed experiments to find out whether there is a better training and inference strategy to improve robustness to background noise, without changing the network configuration.

Literature Survey

Speech processing and noise reduction have gained significant attention in recent years due to their importance in improving communication systems. Awotunde et al. [1] presented a deep learning-based approach for speech segregation in noisy environments. Speech serves as a natural and efficient mode of communication, conveying not only information but also emotions and attitudes. However, background noise often degrades speech quality, making it difficult to extract meaningful information. The study focused on separating desired speech signals from background noise using a supervised learning approach. A convolutional neural network (CNN) was employed to extract auditory features and temporal characteristics of speech signals. Initially, a speaker-independent model was used to separate vocalization patterns, followed by the application of signal-to-noise ratio (SNR) estimation to refine the model. The process involved iterative re-estimation of speech signals until convergence was achieved. The proposed method was evaluated using the TIMIT dataset, and the results demonstrated improved performance in speech isolation and faster convergence compared to traditional speech processing techniques, especially under varying SNR conditions.

Similarly, Bhat et al. [2] proposed a real-time CNN-based speech enhancement system aimed at assisting hearing-impaired individuals. The study emphasized the importance of speech segregation in improving clarity and intelligibility in noisy environments. The authors utilized a convolutional neural network to

model speech features and suppress background noise effectively. The system was designed to operate in real time using a smartphone platform, making it practical and accessible for everyday use. Like the previous study, the model incorporated SNR-based adjustments and iterative refinement to enhance speech quality. Experimental results indicated that the proposed approach significantly improved speech intelligibility and noise reduction performance. The system was found to be efficient, lightweight, and suitable for real-time applications, demonstrating its potential for assistive technologies.

Overall, these studies highlight the effectiveness of deep learning techniques, particularly convolutional neural networks, in addressing speech enhancement and noise reduction challenges. They demonstrate that AI-based approaches can outperform traditional methods in terms of accuracy, speed, and adaptability, making them highly suitable for real-world applications.

Methodology

The code was developed in Python and libraries which are necessary are used. The dataset is downloaded from kaggle. The data is then divided into training dataset and testing dataset. ML algorithms which are apt for this problem are used. SYSTEM MODULE: 3.1 Dataset collection Data Gathering is the first step of the deep learning life cycle. The goal of this step is to identify and obtain all data-related problems. In this step, we need to identify the different data sources, as data can be collected from kaggle such as audio files. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction. This step includes the below tasks: ¬ Identify various data sources ¬ Collect data ¬ Integrate the data obtained from different sources By performing the above task, we get a coherent set of data, also called as a dataset. It will be used in further steps.

Data Analysis Now the cleaned and prepared data is passed on to the analysis step. This step involves: ¬ Selection of analytical techniques ¬ Building models ¬ Review the result

Computational Resources

Hardware Requirements • Processor: Intel i3 • RAM: 4 GB • Hard Disk: 500 GB Software Requirements • Operating System: Windows 8 • Platform: Python 3.7 • IDE: Visual Studio Code • Framework: Streamlit • Audio Processing Library: Librosa • Deep Learning Library: TensorFlow / PyTorch

Implementation

Technologies

Python

Python is the primary programming language used in the development of the bone fracture detection system. It is widely preferred for artificial intelligence and machine learning applications due to its simplicity, readability, and extensive library support. In this project, Python is used for loading and preprocessing X-ray images, building and training deep learning models, and integrating the graphical user interface. Libraries such as TensorFlow and Keras are utilized for model development and inference, while image processing tasks are handled using OpenCV and NumPy. The flexibility of Python allows seamless integration of all system components into a unified application.

HTML and CSS

HTML and CSS are used to design and structure the user interface of the system. HTML (HyperText Markup Language) provides the basic framework of the web interface by organizing elements such as headings, images, buttons, and text content. CSS (Cascading Style Sheets) is used to enhance the visual appearance of the interface by applying styles such as colors, layouts, fonts, and spacing. Together, HTML and CSS ensure that the application interface is user-friendly, visually appealing, and responsive across different devices.

Streamlit

Streamlit is an open-source Python framework used to develop interactive web applications for data science and machine learning projects. It enables developers to convert Python scripts into fully functional web applications without requiring extensive knowledge of front-end technologies. Streamlit provides built-in components such as buttons, sliders, and input fields, allowing users to interact with the system easily. It also supports data visualization through libraries like Matplotlib and Pandas. Due to its simplicity and efficiency, Streamlit is used in this project to create an intuitive and interactive interface with minimal coding effort.

Python Libraries

NumPy

NumPy (Numerical Python) is a fundamental open-source library used for numerical and scientific computing. It provides support for large multi-dimensional arrays and matrices, along with a wide range of mathematical functions for efficient data processing. Compared to standard Python data structures, NumPy offers significantly faster computation due to its optimized internal implementation. It supports operations such as linear algebra, statistical analysis, and random number generation. The core data structure in NumPy is the ndarray, which enables efficient storage and

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

manipulation of data. NumPy also integrates seamlessly with other libraries, making it an essential tool for machine learning and data analysis tasks.

Pandas

Pandas is a powerful open-source Python library used for data manipulation, analysis, and preprocessing of structured data. It provides two primary data structures: Series (one-dimensional) and DataFrame (two-dimensional), which allow efficient handling of tabular data similar to spreadsheets or databases. Pandas supports various operations such as filtering, grouping, merging, and transforming data, making it highly useful in data science and machine learning workflows. It also allows easy reading and writing of data in multiple formats, including CSV, Excel, and SQL databases. Built on top of NumPy, Pandas integrates well with other libraries and plays a crucial role in preparing data for model training and analysis.

Testing

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

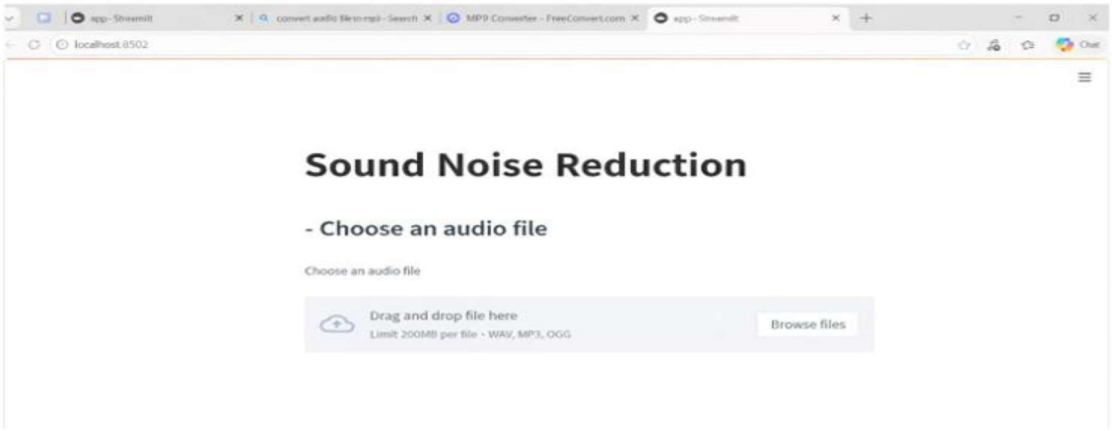
As per the current trend, due to constant change and development in digitization, our lives are improving in all areas. The way we work is also changed. We access our bank online, we do shop online; we order food online and many more. We rely on software's and systems. What if these systems turnout to be defective? We all know that one small bug shows huge impact on business in terms of financial loss and goodwill. To deliver a quality product, we need to have Software Testing in the Software Development Process.There are different type of testings.

Test Cases

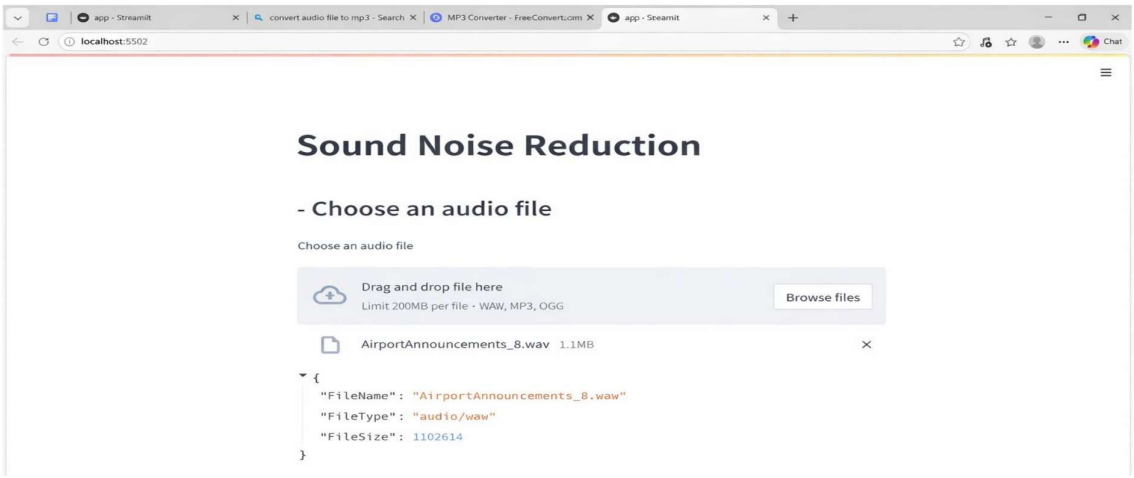
Test Id	Test name	Inputs	Expected Output	Actual Output	Satus
1	Load Dataset	CSV File	Read Dataset	Load Dataset	success
2	Split dataset	Train 80% and test 20%	Divide the training set and testing set	Split train and test	success
3	Train Dataset	Train dataset, random value, predicted class	Train with best accuracy	Train with best accuracy	success
4	Validate Model	No.of Epochs	Validate the model with best fit	Model Generated	success
5	Predict Accuracy and Error Rate	Accuracy	Plot expected accuracy and predicted accuracy	Plot expected predicted accuracy	success
6	Test Data	Test Column	Predicted accuracy	Predicted accuracy	success

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

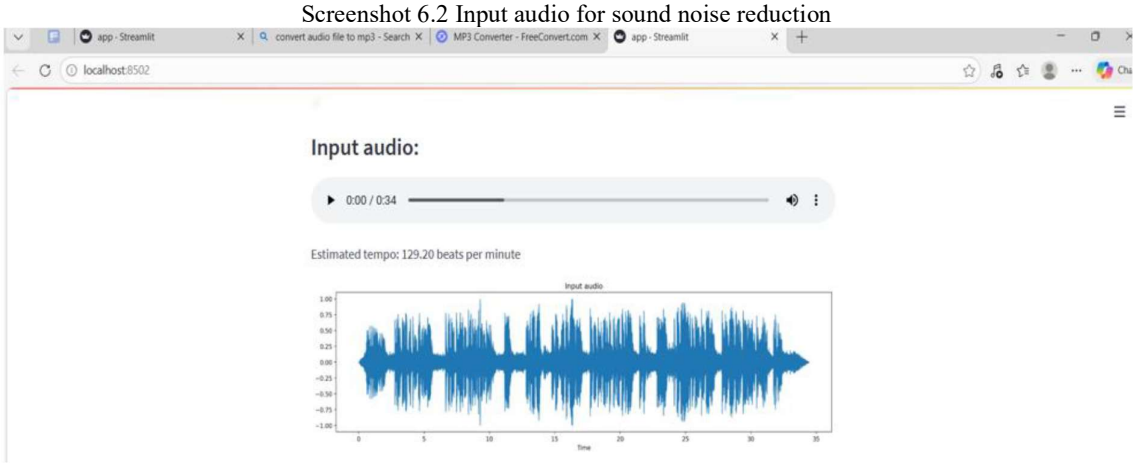
Screenshots



Screenshot 6.1 Sound noise reduction by uploading audio

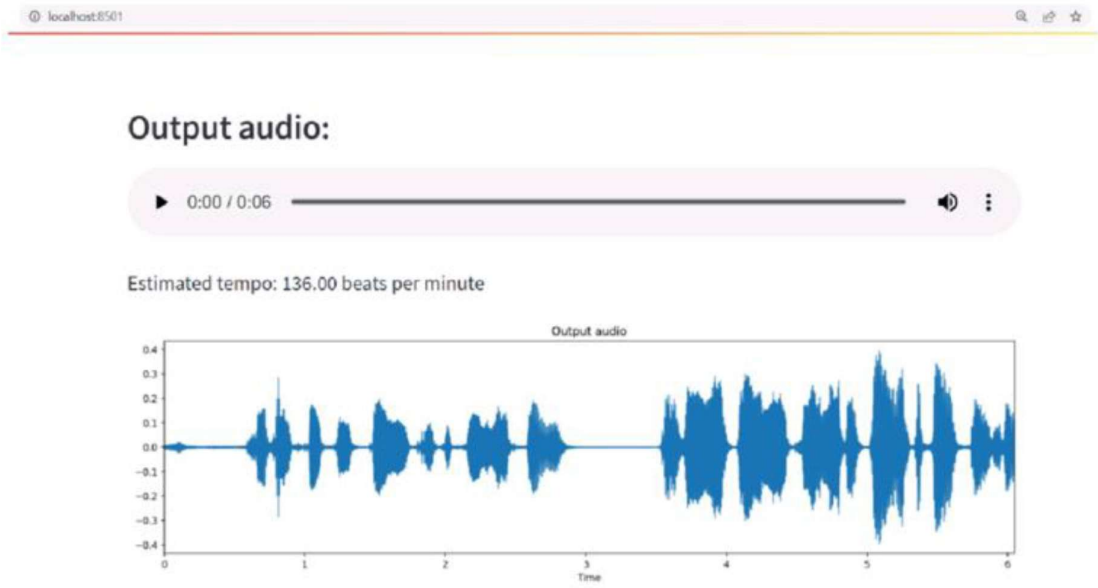


Screenshot 6.2 Input audio for sound noise reduction

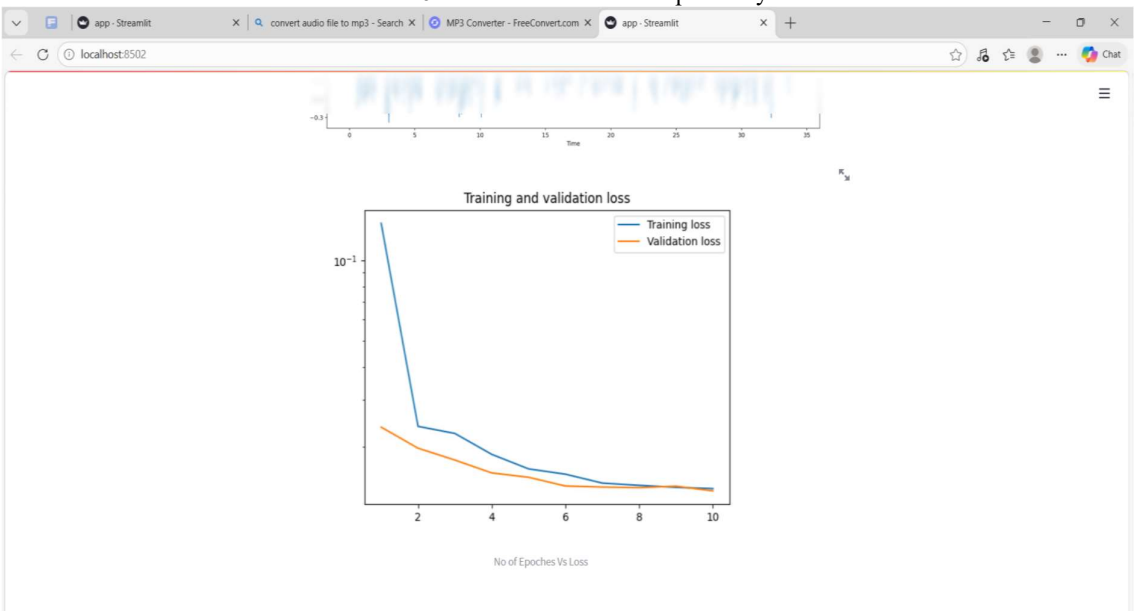


Screenshot 6.3 Input audio

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026



Screenshot 6.4 Denoised audio for input noisy audio



Screenshot 6.5 Graph between training and validation loss

Conclusion

We provide this platform for audio denoising to present a high fidelity historical recordings, we provide the better denoised recordings which are significantly better than the historical recordings. Therefore in this audio we do varying levels of background noise and remove noisy data's. Our audio denoising platform enhances the quality of historical recordings by effectively removing background noise while maintaining high fidelity. It produces clearer, distortion-free sound compared to original audio. The system demonstrates the power of machine learning in restoring and improving audio clarity, making old recordings more valuable for preservation, research, and real-world applications.

References

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2024.

[2] W. B. Kleijn, F. S. C. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, "WaveNet-based low-rate speech coding," in *Proc.*

ISSN 2347–3657
Volume 14, Issue 1, 2026

IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 676–680.

[3] G. S. Bhat, N. Shankar, C. K. A. Reddy, and I. M. S. Panahi, "A real-time convolutional neural network-based speech enhancement for hearing-impaired listeners using smartphone," *IEEE Trans. Audio, Speech, and Language Processing*, 2023.

[4] S. Moon and J.-N. Hwang, "Coordinated training of noise removing networks," *IEEE Signal Processing Letters*, 2024.