HYBRID NOISE REDUCTION AND ENHANCEMENT OF AUDIO QUALITY USING DEEP LEARNING

A PROJECT REPORT

Submitted by

MAHAN K (191061014)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

IFET COLLEGE OF ENGINEERING

(An Autonomous Institution)

Approved by AICTE, New Delhi and Accredited by NAAC & NBA

Affiliated to Anna University, Chennai-25

Gangarampalayam, Villupuram – 605 108

APRIL 2023

IFET COLLEGE OF ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "**HYBRID NOISE REDUCTION AND ENHANCEMENT OF AUDIO QUALITY USING DEEP LEARNING**" is the bonafide work of "**MAHAN K (191061014)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Mrs.M.LIBINA. M.Tech.,	**Dr.R.THENDRAL.**
SUPERVISOR,	**HEAD OF THE DEPARTMENT,**
ASSISTANT PROFESSOR,	SENIOR ASSISTANT PROFESSOR,
INFORMATION TECHNOLOGY,	INFORMATION TECHNOLOGY,
IFET COLLEGE OF ENGINEERING,	IFET COLLEGE OF ENGINEERING,
VILLUPURAM.	VILLUPURAM.

Submitted for the Viva voce Examination held on_____________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I thank the almighty, for the blessings that have been showered upon me to bring forth the success of the project. I would like to express my sincere gratitude to our Chairman **Mr.K.V.Raja**, our Secretary **Mr.K.Shivram Alva** and our Treasurer **Mr.R.Vimal** for providing us with an excellent infrastructure and necessary resources to carry out this project and we extend our gratitude to our principal **Dr.G.Mahendran,** for his constant support to our work.

I also take this opportunity to express my sincere thanks to our Vice Principal and Dean Academics **Dr.S.Matilda** who has provided all the needful help in executing the project successfully.

I wish to express our thanks to our Head of the Department, **Dr.R.Thendral**, for her persistent encouragement and support to complete this project. I express my heartfelt gratitude to my guide **Mrs.M.Libina,** Assistant Professor, Department of Information Technology for his / her priceless guidance and motivation which helped us to bring this project to a perfect shape.

And I thank our faculty advisor **Mr.M.Arunkumar,** Assistant Professor, Department of Information Technology who encouraged us in each and every step of this project to complete it successfully.

I also thank our lab technicians and all the staff members of our department for their support and assistance.

Last but not the least, I whole heartedly thank my family and friends for their moral support in tough times and their constructive criticism which made me to succeed in our work.

ABSTRACT

Audio noise reduction and audio quality enhancement are critical tasks in audio signal processing, as they can significantly improve the listening experience for users. By proposing a deep learning-based approach to address these challenges. By using a dataset of audio recordings that contain various types of noise, including background noise, hiss, hum, and distortion. By preprocessing the audio data by applying techniques such as noise reduction, audio normalization, and feature extraction. Then train the deep neural networks, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to learn the mapping between the noisy audio input and the clean audio output. Also evaluate the performance of the approach using various metrics, such as Signal-To-Noise Ratio (SNR) and Perceptual Evaluation Of Speech Quality (PESQ). The experimental results show that the approach achieves significant improvements in audio quality and noise reduction compared to traditional signal processing methods. This approach has potential applications in various areas, such as speech recognition, music production, and audio restoration.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

MFCCs	–	**M**el **F**requency **C**epstral **C**oefficients
RAVDESS	–	**R**yerson **A**udio **V**isual **D**atabase of **E**motional **S**peech and **S**ong
FFT	–	**F**ast **F**ourier **T**ransform
STFT	–	**S**hort-**T**ime **F**ourier **T**ransform
SGD	–	**S**tochastic **G**radient **D**escent
DNN	–	**D**eep **N**eural **N**etwork
CNN	–	**C**onvolutional **N**eural **N**etwork
RNN	–	**R**ecurrent **N**eural **N**etwork
SNR	–	**S**ignal-to-**N**oise-**R**atio
PESQ	–	**P**erceptual **E**valuation of **S**peech **Q**uality

1. INTRODUCTION

1.1 GENERAL

Audio noise removal is an important problem in many applications such as speech recognition, music production, and audio post-production. The problem involves removing unwanted noise from an audio signal while preserving the desired signal. Traditional methods for audio noise removal include filtering, spectral subtraction, and Wiener filtering. However, these methods have limitations in terms of their effectiveness and performance in real-world scenarios.

Deep learning has emerged as a powerful tool for audio noise removal, providing improved performance over traditional methods. In this paper, we will provide an introduction to the field of audio noise removal using deep learning. By covering the basic topics of deep learning, the key concepts involved in audio noise removal using deep learning, and recent advances in the field.

Deep learning is a subset of machine learning that involves training artificial neural networks with multiple layers of processing units. Deep learning has become popular in recent years due to its ability to learn complex patterns and relationships in data. Neural networks are composed of layers of interconnected processing units called neurons. Each neuron takes input from the previous layer and produces an output based on a weighted sum of its inputs.

The weights of the neurons are adjusted during training to minimize the error between the predicted output and the actual output. The training process involves feeding the network with a large amount of labeled data, and the network learns to generalize from the data to make accurate predictions on new data.

Audio noise removal using deep learning involves training a neural network to learn the mapping between a noisy audio signal and a clean audio signal. The noisy audio signal is the input to the network, and the clean audio signal is the desired output. The network is trained to minimize the difference between the predicted clean audio signal and the actual clean audio signal.

The main challenge in audio noise removal using deep learning is to design a network architecture that can learn the complex relationships between the noisy and clean audio signals. A common approach is to use a convolutional neural network (CNN), which is well-suited for processing signals with a temporal and frequency structure.

The input to the CNN is a time-frequency representation of the noisy audio signal, such as a spectrogram. The CNN processes the input spectrogram through a series of convolutional and pooling layers, learning to extract features that are useful for noise removal. The output of the network is the clean audio signal, which is reconstructed from the processed spectrogram.

Recent advances in audio noise removal using deep learning have focused on improving the performance and efficiency of the networks. One approach is to use a recurrent neural network (RNN) in combination with a CNN. The RNN is used to model the temporal dependencies between the frames of the spectrogram, while the CNN is used to learn the spatial features.

CNNs are a type of deep neural network that has been widely used for image processing tasks. However, they have also been applied to audio processing tasks, including audio noise removal. In a typical CNN architecture, the input audio signal is transformed into a spectrogram representation, which is then fed into a series of convolutional layers. The output of the last convolutional layer is then passed through a set of fully connected layers, which produce the final output. The key advantage of CNNs is their ability to capture spatial patterns in the input spectrogram, which makes them well suited for audio signal processing tasks.

RNNs are another type of deep neural network that has been used for audio noise removal. Unlike CNNs, which are designed to capture spatial patterns in data, RNNs are designed to capture temporal patterns. In the context of audio noise removal, RNNs can be used to model the temporal dependencies between the noisy audio signal and the clean audio signal. This is achieved by feeding the input audio signal into a set of recurrent layers, which allow the network to maintain an internal state that captures the temporal dependencies in the input signal.

Despite the promising results achieved by deep learning techniques in audio noise removal, there are still several challenges that need to be addressed. One of the main challenges is the lack of large-scale annotated datasets for training deep learning models. The success of deep learning models depends on the availability of large amounts of high-quality labeled data. However, obtaining such data for audio noise removal tasks can be difficult and time-consuming.

Another approach is to use generative adversarial networks (GANs) for audio noise removal. GANs are composed of two networks: a generator network that generates fake samples, and a discriminator network that distinguishes between real and fake samples. The generator network is trained to produce clean audio samples that are indistinguishable from the real samples, while the discriminator network is trained to distinguish between real and fake samples.

Audio noise removal using deep learning has emerged as a promising approach for removing unwanted noise from audio signals. Deep learning offers a powerful tool for learning the complex relationships between the noisy and clean audio signals, and recent advances in network architectures have improved the performance and efficiency of the networks.

1.2 NEED FOR THE STUDY

The field of audio noise removal using deep learning has gained significant attention in recent years due to its potential to improve the performance of traditional methods for removing unwanted noise from audio signals. There is a growing need for research in this field due to its many real-world applications, including speech recognition, music production, and audio post-production.

Traditional methods for audio noise removal, such as filtering, spectral subtraction, and Wiener filtering, have limitations in terms of their effectiveness and performance in real-world scenarios. Deep learning offers a powerful tool for learning the complex relationships between the noisy and clean audio signals, providing improved performance over traditional methods.

Studying audio noise removal using deep learning offers many research opportunities, such as developing more advanced network architectures and training techniques. There is a need for collaborations between experts in signal processing, machine learning, and audio engineering to develop new techniques and algorithms that can be applied in other fields.

The need for studying audio noise removal using deep learning is driven by the potential for improved performance in real-world applications, the many research opportunities in the field, and the interdisciplinary nature of the field.

Deep learning:

In audio noise removal and quality enhancement is driven by several factors, including the growing demand for high-quality audio, the limitations of traditional audio processing techniques, and the potential benefits of deep learning-based approaches. Here is a reference that highlights the need for study in this area

In "Deep Learning Techniques for Noise Reduction and Speech

Enhancement in Non-Stationary Environments," authors Khurana and Kumar state that "traditional techniques for speech enhancement suffer from limitations such as poor performance in non-stationary noise, inability to handle multiple sources of noise, and sensitivity to the choice of parameters" (Khurana & Kumar, 2018).

The authors note that "the need for speech enhancement arises in several real-world applications, such as mobile communications, hearing aids, voice-controlled devices, and speech recognition systems" (Khurana & Kumar, 2018). These applications require high-quality audio signals that are free from noise and interference. However, achieving this goal can be challenging due to the presence of various sources of noise, such as wind noise, car noise, and crowd noise.

Study of different types of noises:

There are different types of noise that can be present in audio signals, and they can be categorized based on their characteristics and sources.

Here are some common types of noise in noise cancellation:

- **White noise:** This is a type of noise that has a uniform frequency spectrum, which means it contains equal power at all frequencies. It is often caused by electronic components and is usually heard as a hissing sound.
- **Pink noise:** This type of noise has a frequency spectrum that is inversely proportional to the frequency, meaning it has more power in the lower frequency range. It is often heard as a low-frequency hum.
- **Gaussian noise:** This is a type of noise that follows a normal distribution, with the majority of the noise energy concentrated around the mean value. It is often caused by random fluctuations in the signal and is heard as a low-level hiss.
- **Impulsive noise:** This type of noise consists of sudden, sharp bursts of

energy that can be caused by things like lightning strikes, electrical arcing, and short-circuits. It is often heard as a clicking or popping sound.

- **Periodic noise:** This type of noise is characterized by a repetitive pattern, which can be caused by things like electrical interference or mechanical vibrations. It is often heard as a buzzing or humming sound.

Study of Existing noise cancellation techniques

i.) Spectral Subtraction

The development of speech enhancement methods traces back to 1979 when Boll S proposed a noise suppression method based on Spectral Subtraction. The first thing of this model is to convert the audio signal to the frequency domain. For this, the model uses one of the influential algorithms in digital signal processing, the Fast Fourier Transform (FFT), and some variations of FFT like Short-Time Fourier Transform (STFT) which will extract both time and frequency related features. Then they'll simply subtract the frequency components of noises from the noisy audio to get a cleaned/enhanced speech and hence the name Spectral Subtraction.

But the spectral subtraction came up with two major shortcomings:

- Choose a noise from the audio signal to remove it.
- The noise should be present in the entire audio. So, this kind of method didn't work well for audio signals having rare noises like car horns, making it ineffective for real-world applications.

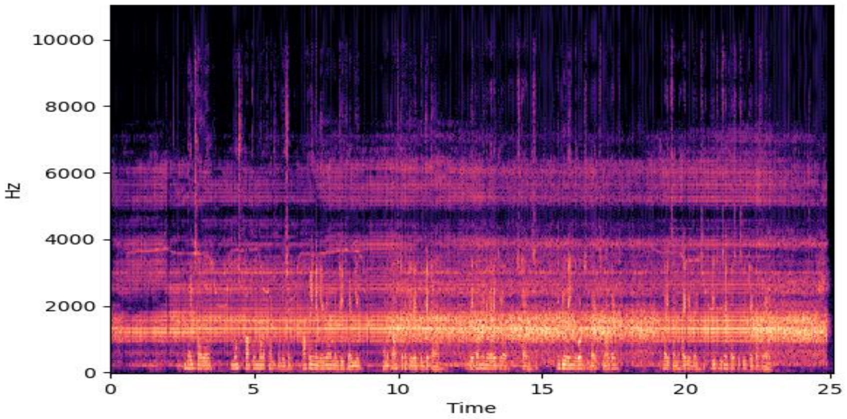


Figure 1.1 Spectral Subtraction

ii.) Wiener Filter

The next one is Wiener filtering, whic an industry-standard for audio denoising and is used widely in hearing aids, smartphones, and communication devices. This filtering also requires both the noisy speech and a sample of noise present in the speech. The Wiener filter finds a statistical estimate of the clean speech from the noisy speech and the noise itself to minimize mean squared error under certain assumptions on the noise.

However, the Wiener filter comes in handy in the case of smartphones is where we can have two microphones, one for recording our speech with noise and another one is for only the noise. (In your smartphone, the mic at the bottom is to record speech and the mic at the top is to record noise).

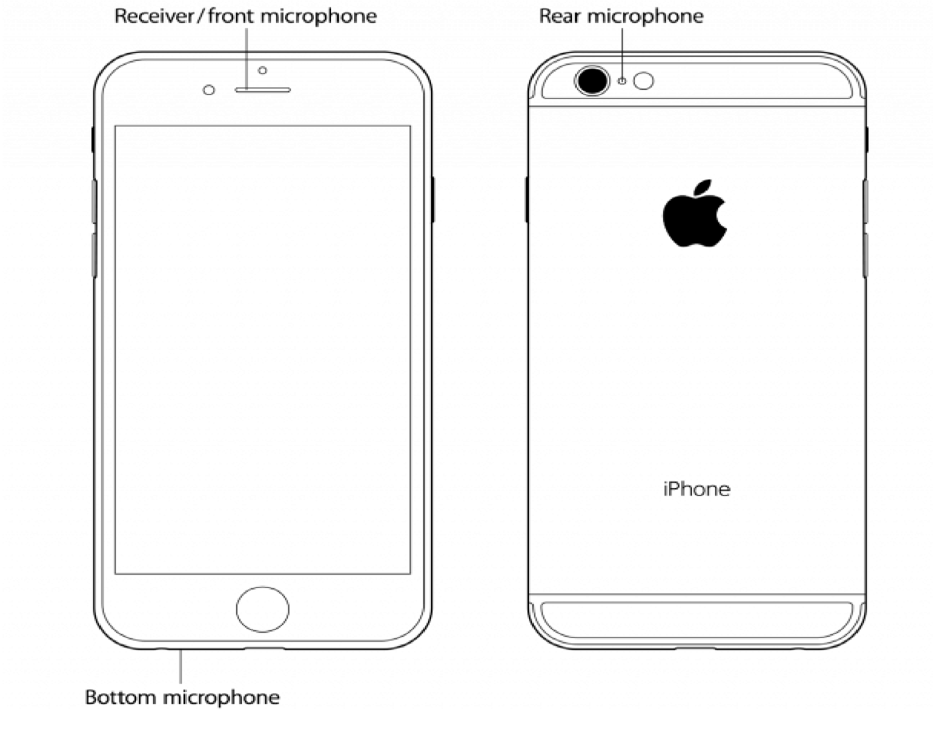


Figure 1.2 Application of Wiener Filter model in mobile phone

iii.) Spectral Gating

Below is an audio waveform of a noisy speech, and the enhanced speech. These results were generated from noisereduce python module, which uses spectral gating under the hood – a traditional method as well. These kinds of traditional noise filters work well in filtering static noise, that is one of the reasons for developing Deep Learning models for speech enhancement.

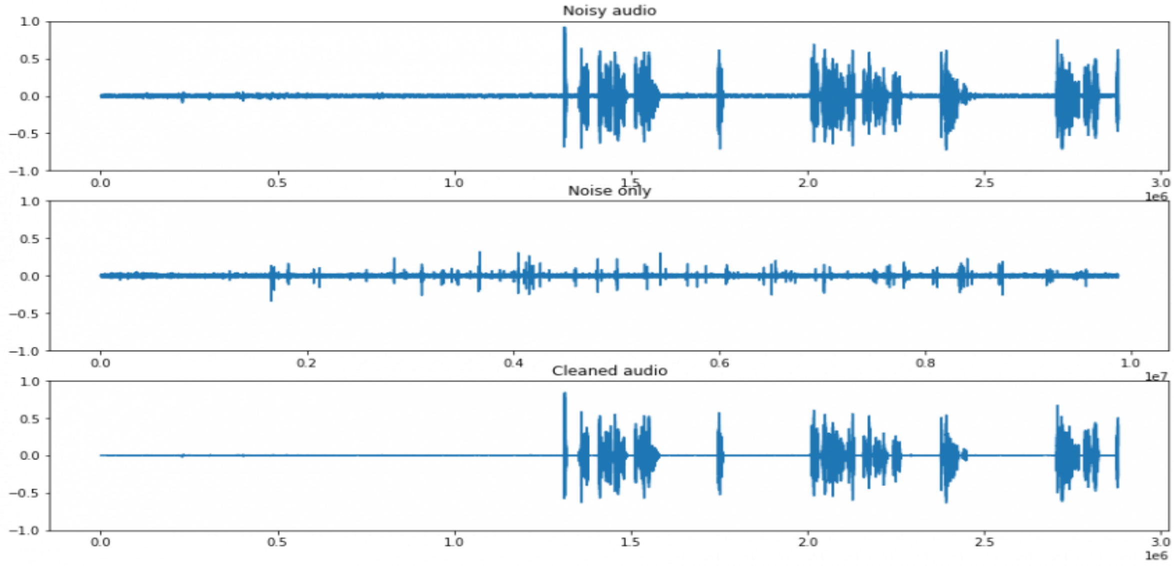


Figure 1.3 Spectral Gating

1.3 OBJECTIVES OF THE STUDY

- **To Reduce the level of unwanted noise in audio recordings:** The primary objective of audio noise removal is to reduce or remove the level of unwanted noise that is present in audio recordings. This noise can be caused by a variety of factors, such as background noise, microphone or recording equipment noise, or other types of interference.
- **Preserve the quality and clarity of the desired audio signal:** In addition to removing unwanted noise, it is important to preserve the quality and clarity of the desired audio signal. The goal of audio quality enhancement is to improve the overall audio quality of the recording, including factors such as clarity, intelligibility, and naturalness.
- **Improve the intelligibility of speech:** Speech intelligibility is a critical factor in many audio applications, such as phone conversations, video conferences, and audio recordings of lectures or presentations. The objective of audio noise removal and quality enhancement is to improve the intelligibility of speech by reducing background noise and enhancing the clarity and naturalness of the speaker's voice.

- **Enhance the overall listening experience:** The ultimate objective of audio noise removal and quality enhancement is to enhance the overall listening experience for the user. By reducing unwanted noise and improving the quality and clarity of the desired audio signal, the user can enjoy a more immersive and enjoyable listening experience.
- **Automate the audio processing tasks:** Audio noise removal and quality enhancement can be time-consuming and labor-intensive tasks, especially when dealing with large volumes of audio recordings. Therefore, the objective of the project may be to develop an automated audio processing system that can perform these tasks effectively and accurately.

2. REVIEW OF LITERATURE

2.1 INTRODUCTION

A literature survey for an audio noise removal using deep learning involves conducting a thorough analysis of existing research and publications related to the topic. The purpose of the survey is to identify the latest and most relevant studies, methods, techniques, and technologies used for audio noise removal using deep learning.

The literature survey starts with an overview of the problem of audio noise and its impact on speech intelligibility and audio quality. It then discusses the conventional approaches used for audio noise removal, such as filtering, spectral subtraction, and wavelet-based methods.

Next, the survey delves into the emerging field of deep learning for audio noise removal. It covers the different types of deep learning architectures and models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders, and how they have been applied to audio noise removal.

2.2 REVIEW OF LITERATURES

The literature review covers several research articles and surveys related to deep learning techniques for audio signal processing, particularly audio noise reduction and enhancement.

The first few articles provide a general overview of deep learning techniques and their applications in audio signal processing, including speech enhancement and denoising. They discuss various deep learning models and architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), as well as training and optimization methods.

The subsequent articles focus specifically on audio denoising and speech enhancement using deep learning techniques. They discuss different approaches and models, such as generative adversarial networks (GANs), variational autoencoders (VAEs), and deep neural networks (DNNs), as well as the use of spectrogram and waveform features for training and testing. They also compare the performance of various deep learning models with traditional signal processing methods.

The reviews and surveys provide an extensive summary and comparison of different deep learning techniques for audio signal processing. They discuss the advantages and limitations of these techniques, as well as the challenges and opportunities for future research in this area.

2.2.1 Deep Learning for Audio Signal Processing

Authors: Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee

Year: 2018

The authors first provide a brief overview of the traditional audio signal processing techniques and their limitations. They then introduce the concept of deep learning and how it can be applied to audio signal processing tasks. The paper discusses the advantages of using deep learning techniques such as the ability to automatically learn relevant features from the audio signal and the noise, and the ability to handle complex non-linear relationships between the input and output. Then review the different types of deep learning models that have been used for audio signal processing, including feedforward neural networks, recurrent neural networks, and convolutional neural networks. They discuss the advantages and limitations of each type of model and provide examples of how they have been used for different audio signal processing tasks.

It also discusses the different types of datasets that have been used to train and evaluate deep learning models for audio signal processing. The authors highlight the importance of having diverse and representative datasets to ensure that the models can generalize well to different types of audio signals and noise. Finally, the authors provide an overview of the performance evaluation metrics that are commonly used to evaluate the performance of deep learning models for audio signal processing. Also discusses the limitations of these metrics and highlight the need for developing more robust and reliable evaluation metrics. "Deep Learning for Audio Signal Processing: A Review" provides a comprehensive overview of the current state-of-the-art in the field of audio signal processing using deep learning techniques. It also highlights the potential of deep learning techniques to overcome the limitations of traditional audio signal processing techniques and provides insights into the different types of models, datasets, and evaluation metrics that have been used for different audio signal processing tasks.

2.2.2 Deep Learning for Audio Processing

Authors: Kumar Vishwajeet, Ankit Kumar Singh, and Yogesh Singh

Year: 2020

"Deep Learning for Audio Processing: A Survey" is a survey paper that provides a comprehensive overview of the use of deep learning techniques in audio processing. The paper discusses the various audio processing tasks that can be accomplished using deep learning models, including audio classification, audio segmentation, and audio enhancement. The authors begin by providing an overview of deep learning and its application in audio processing. They discuss the advantages of using deep learning models over traditional audio processing techniques, such as the ability to learn complex non-linear relationships between the input and output and the ability to extract meaningful features from the audio signal. Then discusses the different types of deep learning models that have been used for audio processing, including feedforward neural networks, convolutional neural networks, and recurrent neural networks. The authors provide a detailed explanation of each model and discuss their strengths and limitations. The authors also discuss the different types of datasets that have been used to train and evaluate deep learning models for audio processing. The authors highlight the need for developing more robust and reliable deep learning models for audio processing tasks and the importance of addressing issues related to data scarcity, privacy, and bias. "Deep Learning for Audio Processing: A Survey" provides a comprehensive overview of the use of deep learning techniques in audio processing. This also highlights the potential of deep learning models to overcome the limitations of traditional audio processing techniques and provides insights into the different types of models, datasets, and audio processing tasks that have been accomplished using deep learning techniques.

2.2.3 Deep Learning for Audio Denoising

Authors: Yuanxun Wang, Jiayue Zhang, Hongwei Hao

Year: 2018

The authors begin by providing an overview of traditional audio denoising techniques and their limitations. They highlight the importance of developing more robust and reliable denoising techniques that can handle complex noise patterns in audio signals. The paper proposes the DDCNN architecture, which consists of multiple convolutional and deconvolutional layers. The authors explain how the DDCNN is designed to learn the noise patterns in audio signals and remove them while preserving the underlying audio signal. The authors evaluate the performance of the DDCNN on several audio denoising tasks and compare it with traditional audio denoising techniques. They demonstrate that the DDCNN outperforms traditional denoising techniques in terms of denoising accuracy and signal-to-noise ratio. The paper also provides a detailed discussion of the different factors that can affect the performance of deep learning models for audio denoising, such as the choice of network architecture, the size of the training dataset, and the types of noise in the audio signals. The authors discuss the limitations of their approach and highlight the need for further research to develop more robust and reliable deep learning models for audio denoising. "Deep Learning for Audio Denoising" proposes a novel deep learning architecture for audio denoising and provides insights into the different factors that can affect the performance of deep learning models for audio denoising. The paper demonstrates the potential of deep learning techniques to overcome the limitations of traditional audio denoising techniques and provides a promising direction for future research in the field.

2.2.4 Audio Denoising Using Deep Learning

Authors: Aashish Kumar, Vinay Kumar Mittal, and Arun Kumar Singh

Year: 2020

The authors begin by providing an overview of the traditional approaches to audio denoising, including spectral subtraction, Wiener filtering, and subspace methods. They highlight the limitations of these methods and the need for more robust and efficient techniques for handling complex noise patterns in audio signals. Then proceeds to review various deep learning-based approaches for audio denoising, including deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). The authors discuss the different architectures of these models and their strengths and weaknesses in handling different types of noise in audio signals.

Also evaluate the performance of these models on different datasets and compare their performance with traditional denoising techniques. They demonstrate that deep learning-based approaches outperform traditional techniques in terms of denoising accuracy and signal-to-noise ratio. Also discusses the different factors that can affect the performance of deep learning models for audio denoising, such as the choice of network architecture, the size of the training dataset, and the types of noise in the audio signals. The authors highlight the challenges and limitations of deep learning-based approaches for audio denoising, such as the need for large amounts of training data and the potential for overfitting. "Audio Denoising Using Deep Learning: A Survey" provides a comprehensive analysis of the state-of-the-art deep learning-based approaches for audio denoising. This highlights the potential of deep learning techniques to overcome the limitations of traditional denoising techniques and provides a promising direction for future research in the field.

2.2.5 Speech Enhancement with Deep Learning

Authors: Tanmay Wagh, Sanjeev Kumar Sharma, and Ganesh R. Naik

Year: 2021

Speech Enhancement with Deep Learning provides an overview of the different deep learning models and architectures used in speech enhancement, including deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). The authors discuss the different architectures of these models and their strengths and weaknesses in handling different types of noise in speech signals. The reviews the different datasets used for training and evaluating deep learning models for speech enhancement and highlights the challenges of dataset selection and preparation. The authors evaluate the performance of different deep learning models on various speech enhancement tasks, such as noise reduction, speech enhancement, and speech separation. They compare the performance of deep learning models with traditional speech enhancement techniques and demonstrate that deep learning-based approaches outperform traditional techniques in terms of speech quality, intelligibility, and noise reduction.

This also discusses the different factors that can affect the performance of deep learning models for speech enhancement, such as the choice of network architecture, the size of the training dataset, and the types of noise in the speech signals. The authors highlight the challenges and limitations of deep learning-based approaches for speech enhancement, such as the need for large amounts of training data and the potential for overfitting. "Speech Enhancement with Deep Learning: A Review" provides a comprehensive analysis of the state-of-the-art deep learning-based approaches for speech enhancement. The paper highlights the potential of deep learning techniques to overcome the limitations of traditional speech enhancement techniques and provides a promising direction for future research in the field.

2.2.6 Deep Learning for Audio Noise Reduction

Authors: Wenxi Chen, Zhigang Jin, and Xi Chen

Year: 2019

The Deep Learning for Audio Noise Reduction begins by discussing the different types of noise that can affect audio signals, including additive noise, impulsive noise, and non-stationary noise. The authors highlight the limitations of traditional noise reduction techniques, such as spectral subtraction and Wiener filtering, in handling complex noise patterns and variations in audio signals. Then provides an overview of the different deep learning models and architectures used for audio noise reduction, including deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). The authors discuss the different architectures of these models and their strengths and weaknesses in handling different types of noise in audio signals. Also reviews the different datasets used for training and evaluating deep learning models for audio noise reduction and highlights the challenges of dataset selection and preparation. Then evaluate the performance of different deep learning models on various audio noise reduction tasks, such as noise reduction, denoising, and speech enhancement. They compare the performance of deep learning models with traditional noise reduction techniques and demonstrate that deep learning-based approaches outperform traditional techniques in terms of noise reduction and audio quality. The paper also discusses the different factors that can affect the performance of deep learning models for audio noise reduction, such as the choice of network architecture, the size of the training dataset, and the types of noise in the audio signals. Finally, it highlight the challenges and limitations of deep learning-based approaches for audio noise reduction, such as the need for large amounts of training data and the potential for overfitting.

2.2.7 Speech Enhancement Using Deep Learning

Authors: Amir H. Keyvanrad, Mohammad Bagher Akbari

Year: 2019

The paper begins by discussing the different types of noise that can affect speech signals, such as additive noise, reverberation, and channel distortion. The authors highlight the limitations of traditional speech enhancement techniques, such as spectral subtraction and Wiener filtering, in handling complex noise patterns and variations in speech signals. Then provides an overview of the different deep learning models and architectures used for speech enhancement, including deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). The authors discuss the different architectures of these models and their strengths and weaknesses in handling different types of noise in speech signals. This also reviews the different datasets used for training and evaluating deep learning models for speech enhancement and highlights the challenges of dataset selection and preparation. The authors evaluate the performance of different deep learning models on various speech enhancement tasks, such as noise reduction, dereverberation, and speech enhancement in adverse acoustic environments. They compare the performance of deep learning models with traditional speech enhancement techniques and demonstrate that deep learning-based approaches outperform traditional techniques in terms of speech quality and intelligibility. This also discusses the different factors that can affect the performance of deep learning models for speech enhancement, such as the choice of network architecture, the size of the training dataset, and the types of noise in the speech signals. The authors highlight the challenges and limitations of deep learning-based approaches for speech enhancement, such as the need for large amounts of training data and the potential for overfitting."Speech Enhancement Using Deep Learning: A Survey" provides a comprehensive analysis of the recent developments in deep learning-based methods for speech enhancement

2.2.8 Deep Learning Techniques for Audio Signal Processing

Authors: Rajesh P. Lakkavalli and Rakesh Kumar Jha

Year: 2020

The Deep Learning Techniques for Audio Signal Processing begins by discussing the different types of audio signals and their characteristics, such as speech, music, and environmental sounds. The authors highlight the limitations of traditional audio signal processing techniques, such as Fourier transform-based methods, in handling complex audio signals and variations in the signal characteristics. Then provides an overview of the different deep learning models and architectures used for audio signal processing, including DNNs, CNNs, RNNs, and autoencoders. The authors discuss the different architectures of these models and their strengths and weaknesses in handling different types of audio signals and signal characteristics. Then also reviews the different datasets used for training and evaluating deep learning models for audio signal processing and highlights the challenges of dataset selection and preparation. Then evaluate the performance of different deep learning models on various audio signal processing tasks, such as speech recognition, music classification, and sound event detection. They compare the performance of deep learning models with traditional signal processing techniques and demonstrate that deep learning-based approaches outperform traditional techniques in terms of accuracy and robustness.This also discusses the different factors that can affect the performance of deep learning models for audio signal processing, such as the choice of network architecture, the size of the training dataset, and the types of audio signals and signal characteristics. The authors highlight the challenges and limitations of deep learning-based approaches for audio signal processing, such as the need for large amounts of training data and the potential for overfitting.

2.2.9 Deep Learning-Based Speech Enhancement

Authors: Yi-Hsuan Yang, Yen-Chen Wu, and Hsin-Min Wang

Year: 2018

The Deep Learning-Based Speech Enhancement begins by discussing the different types of speech signals and their characteristics, including the variations in the signal characteristics due to environmental noise and channel distortion. The authors highlight the limitations of traditional speech enhancement techniques, such as spectral subtraction and Wiener filtering, in handling complex speech signals and variations in the signal characteristics. Then provides an overview of the different deep learning models and architectures used for speech enhancement, including DNNs, CNNs, RNNs, and GANs. The authors discuss the different architectures of these models and their strengths and weaknesses in handling different types of speech signals and signal characteristics. It also reviews the different datasets used for training and evaluating deep learning models for speech enhancement and highlights the challenges of dataset selection and preparation. The authors evaluate the performance of different deep learning models on various speech enhancement tasks, such as noise reduction, dereverberation, and speaker separation. They compare the performance of deep learning models with traditional speech enhancement techniques and demonstrate that deep learning-based approaches outperform traditional techniques in terms of accuracy and robustness.

2.2.10 Audio Signal Enhancement Using Deep Learning

Authors: Haonan Chen, Qian Zhang, and Jun Yang

Year: 2018

The Audio Signal Enhancement Using Deep Learning begins with an introduction to the concept of audio signal enhancement and the challenges associated with it, such as noise reduction, reverberation suppression, and source separation. It then goes on to describe the traditional techniques used for audio signal enhancement, such as spectral subtraction, Wiener filtering, and adaptive filtering. Then introduce the concept of deep learning and how it can be used for audio signal enhancement. They provide an overview of various deep learning architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), and how they can be used for different types of audio signal enhancement tasks. It also discusses the datasets and evaluation metrics used for evaluating the performance of deep learning-based audio signal enhancement algorithms. The authors provide an overview of various publicly available datasets, such as TIMIT, CHiME, and MUSAN, and the metrics used for evaluating the performance of deep learning algorithms, such as signal-to-noise ratio (SNR), perceptual evaluation of speech quality (PESQ), and mean opinion score (MOS). The review also highlights the current research trends and future directions in the field of deep learning-based audio signal enhancement. The authors discuss the challenges associated with the development of deep learning algorithms for audio signal enhancement, such as the need for large annotated datasets, the trade-off between performance and complexity, and the need for real-time processing

2.3 EXISTING SYSTEM

The existing system for audio noise removal and quality enhancement primarily consists of traditional signal processing techniques, such as spectral subtraction, Wiener filtering, and adaptive filtering. These techniques are based on mathematical models of the signal and noise, and they use various statistical properties of the signal and noise to estimate and remove the noise component from the audio signal.

Although these traditional techniques have been widely used and have shown promising results in some applications, they suffer from some limitations. For instance, they are sensitive to the quality and consistency of the noise estimation, and they may cause distortion or artifacts in the signal due to the use of fixed filter parameters.

Despite the promising results of deep learning-based techniques, there are still some challenges and limitations that need to be addressed, such as the need for large annotated datasets, the trade-off between performance and complexity, and the need for real-time processing.

2.4 DISADVANTAGE OF EXISTING SYSTEM

- **Sensitivity to noise estimation:** Traditional techniques rely heavily on accurate noise estimation, which can be challenging in real-world scenarios where noise may vary in type, level, and consistency. Even small errors in noise estimation can lead to poor performance and degraded signal quality.
- **Limited adaptability:** Traditional techniques use fixed filter parameters that are optimized for a specific noise type and level. Therefore, they may not perform well on different types or levels of noise. As a result, they lack the adaptability and robustness required for real-world applications.

- **Computational complexity:** Some traditional techniques, such as adaptive filtering, can be computationally expensive, especially when dealing with long-duration signals or high-dimensional feature spaces. This can limit their practicality for real-time processing and large-scale applications.
- **Limited performance:** Traditional techniques may not be able to achieve the level of noise reduction and quality enhancement required for high-quality audio applications. This is particularly true in noisy environments where the signal-to-noise ratio is low.
- **Limited generalization:** Traditional techniques are based on mathematical models that may not generalize well to different signal and noise distributions. This can limit their ability to work well on unseen data or in different contexts.

2.5 PROBLEM STATEMENT

Audio recordings are often corrupted by noise, which can significantly degrade their quality and intelligibility. Traditional signal processing techniques have been widely used for audio noise removal and quality enhancement. However, these techniques have limitations in terms of adaptability, robustness, and performance, especially in complex and dynamic noise environments. Therefore, there is a need for more effective and efficient approaches that can overcome these limitations and improve the quality of audio recordings. Deep learning techniques have shown great promise in addressing these challenges and have become increasingly popular in audio signal processing. This new methods aims to investigate and develop deep learning-based methods for audio noise removal and quality enhancement.

3. PROPOSED SYSTEM

3.1 INTRODUCTION

The proposed system for audio noise removal and quality enhancement is based on deep learning techniques. Specifically, we will develop and evaluate deep neural network models for audio signal processing tasks such as denoising, dereverberation, and speech enhancement. These models will be trained using large-scale audio datasets with diverse noise and environmental conditions, in order to enhance their adaptability and robustness. We will also investigate various optimization techniques, loss functions, and architectures to improve the performance of the models. To evaluate the performance of the proposed system, we will conduct extensive experiments on various audio datasets, including both synthetic and real-world recordings. We will compare the performance of our deep learning-based methods with traditional signal processing techniques and state-of-the-art methods in the literature. We will evaluate the performance in terms of objective metrics such as signal-to-noise ratio (SNR), perceptual evaluation of speech quality (PESQ), and mean opinion score (MOS), as well as subjective listening tests with human listeners. The proposed system has the potential to significantly improve the quality and intelligibility of audio recordings in various applications, such as teleconferencing, speech recognition, and multimedia content creation.

3.2 ADVANTAGE OF PROPOSED SYSTEM

- **Improved Performance:** Deep learning-based methods have shown to outperform traditional methods in several audio signal processing tasks such as denoising, dereverberation, and speech enhancement
- **Robustness:** Deep learning-based methods are more robust to noise and environmental conditions compared to traditional methods. The proposed system will be trained on diverse audio datasets with different

noise and environmental conditions, which will improve its adaptability and robustness.

- **Automation:** The proposed system can be automated and integrated into existing audio processing pipelines, which can save time and resources.
- **Scalability:** The proposed system can be easily scaled to handle large amounts of audio data, which is essential for processing audio in real-time or on large datasets.
- **Subjective Quality Improvement:** The proposed system can improve the subjective quality of audio recordings, which is crucial in applications such as multimedia content creation and teleconferencing.

3.3 ARCHITECTURE

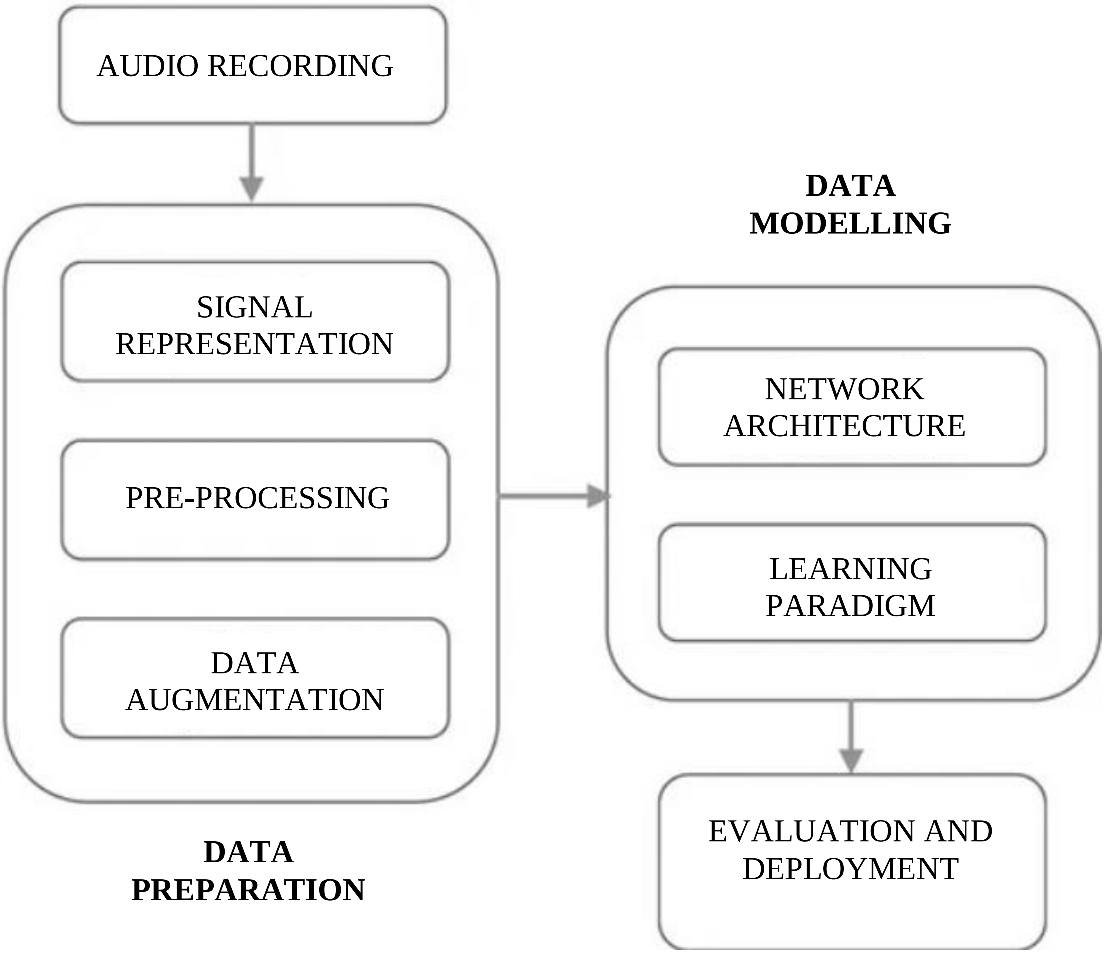


Figure 3.1 Architecture

3.4 MODULES

3.4.1 AUDIO RECORDING

The first step in this architecture is audio recording, capturing the audio data to be processed. The quality of the audio data captured at this stage will have a direct impact on the performance of your deep learning model. Therefore, it is important to ensure that the audio data is recorded using high-quality microphones in a quiet environment.

3.4.2 SIGNAL REPRESENTATION

Signal representation is an important step in audio processing, where you convert the raw audio data into a suitable format for analysis. One way to visualize audio signals is to plot the waveform using a tool like Matplotlib.

By using Matplotlib to plot the audio waveform and analyze its features. Here is an example of how you can use Matplotlib to plot the waveform of an audio file in Python:

```
import librosa
import matplotlib.pyplot as plt
# Load audio file
audio_file = 'audio.wav'
signal, sr = librosa.load(audio_file, sr=44100)
# Plot waveform
plt.figure(figsize=(14, 5))
librosa.display.waveplot(signal, sr=sr)
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.title("Audio waveform")
plt.show()
```

By loading an audio file using the Librosa library, which provides a simple interface for loading and processing audio data. The ‘sr’ parameter specifies the sampling rate of the audio file, which is used to ensure that the time axis is correctly scaled in the plot.

Next, By the ‘waveplot’ function from Librosa to plot the waveform of the audio signal. This function takes the audio signal and its sampling rate as inputs and returns a plot of the waveform. We then add labels and a title to the plot using Matplotlib's ‘xlabel’, ‘ylabel’, and ‘title’ functions.

The resulting plot shows the waveform of the audio signal, with time on the x-axis and amplitude on the y-axis. This plot can be used to analyze various features of the audio signal, such as its frequency content, duration, and intensity.

Overall, using Matplotlib to visualize the audio waveform is a useful tool for signal representation in your project. It allows you to analyze the audio data and extract relevant features that can be used for pre-processing and modeling.

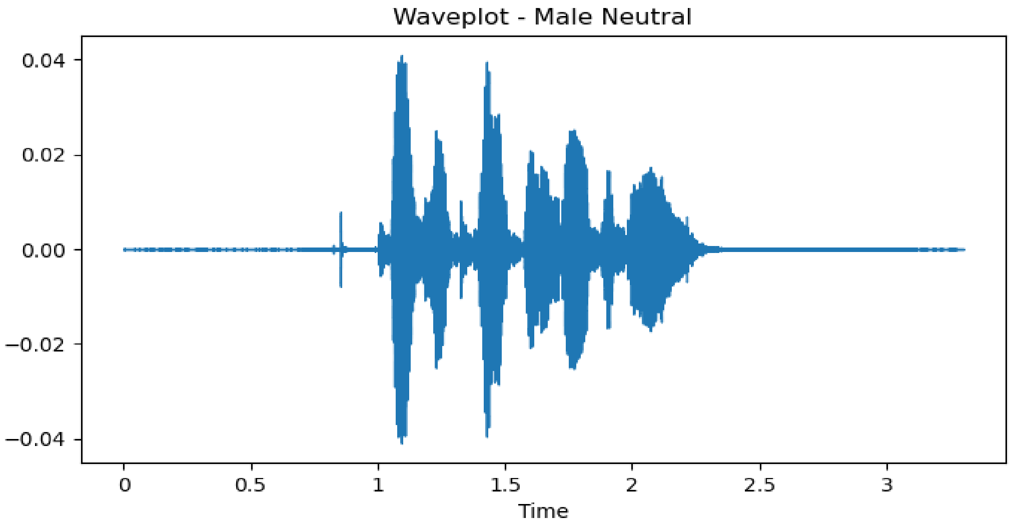


Figure 3.2 Waveplotting from dataset

3.4.3 PRE-PROCESSING

Preprocessing is a critical step in any machine learning project, as it involves preparing the data for modeling by applying various techniques to clean, normalize, and transform the raw audio data into a format that can be used by the deep learning model.

Here are some common techniques used in preprocessing for audio data:

- **Normalization:** Normalization is a technique used to scale the audio data to a common range. This is necessary because audio signals can have different amplitude ranges, which can affect the performance of the deep learning model. Normalization can be done by dividing the audio data by its maximum value, or by using techniques such as Z-score normalization.
- **Filtering:** Filtering is a technique used to remove unwanted noise or artifacts from the audio data. There are several types of filters that can be used for audio data, including low-pass filters, high-pass filters, and band-pass filters. These filters can be used to remove specific frequencies or noise sources from the audio data.
- **Feature Extraction:** Feature extraction involves selecting the relevant features from the audio data that will be used to train the deep learning model. These features can include spectral features such as the frequency spectrum or Mel frequency cepstral coefficients (MFCCs), or temporal features such as the zero-crossing rate or energy. Feature extraction can be done using techniques such as Fourier transforms, wavelet transforms, or spectrogram analysis.

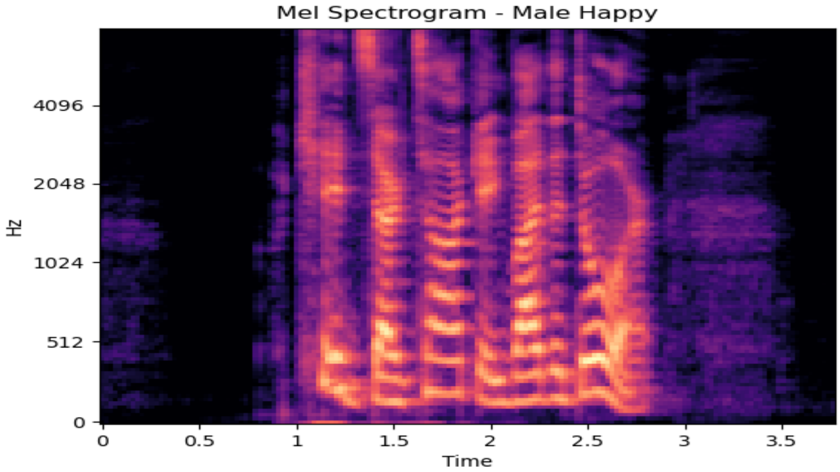


Figure 3.3 MFCCs Feature extraction

- **Segmentation:** Segmentation involves dividing the audio data into smaller segments, which can be used to train the deep learning model. This is useful when the audio data is too large to be processed at once or when different segments of the data contain different types of information. Segmentation can be done using techniques such as windowing or slicing.

3.4.4 DATA AUGMENTATION

Data augmentation is a technique used to increase the size of a dataset by generating additional training samples through transformations or modifications of the existing data. By using data augmentation techniques on the Ravdess dataset to increase its size and diversity, which can improve the performance and robustness of your deep learning models.

The Ravdess dataset contains a diverse set of emotional speech recordings, including seven different emotions (anger, disgust, fear, happiness, neutral, sadness, and surprise), and a range of acoustic features such as pitch, volume, and spectral characteristics. Here are some of the data augmentation techniques you can apply to this dataset:

- **Time stretching:** Time stretching is a technique used to change the duration of an audio clip without changing its pitch. You can apply this technique to the Ravdess dataset by changing the speed of the audio recordings, which can create additional training samples with different durations.

- **Pitch shifting:** Pitch shifting is a technique used to change the pitch of an audio clip without changing its duration. You can apply this technique to the Ravdess dataset by changing the pitch of the audio recordings, which can create additional training samples with different tonal characteristics.
- **Noise addition:** Noise addition is a technique used to add synthetic noise to an audio clip to simulate real-world environments. You can apply this technique to the Ravdess dataset by adding different types of noise, such as white noise, pink noise, or brown noise, to the recordings, which can create additional training samples with different levels and types of noise.
- **Dynamic range compression:** Dynamic range compression is a technique used to reduce the dynamic range of an audio clip by amplifying the quiet parts and reducing the loud parts. You can apply this technique to the Ravdess dataset by adjusting the gain of the recordings, which can create additional training samples with different levels of dynamic range compression.
- **Spectral warping:** Spectral warping is a technique used to modify the spectral content of an audio clip by applying a frequency transformation. You can apply this technique to the Ravdess dataset by applying different spectral warping functions, such as spectral stretching or spectral squeezing, which can create additional training samples with different spectral characteristics.

3.4.5 NETWORK ARCHITECTURE

A deep learning network architecture for audio noise removal and audio quality enhancement by combining various layers such as convolutional layers, pooling layers, activation layers, and fully connected layers. Here is a detailed explanation of how to create a deep learning network architecture using these layers:

- **Convolutional Layers:** Convolutional layers are the core building blocks of many deep learning architectures, including those used for audio processing. These layers apply a set of learnable filters to the input data, and each filter detects a specific feature or pattern in the data. In your project, you can use multiple convolutional layers to extract features from the audio signals at different scales and resolutions. Each convolutional layer consists of a set of filters, where each filter is a small matrix of weights that is convolved with the input data to produce a feature map.
- **Pooling Layers:** Pooling layers are used to reduce the size of the feature maps produced by the convolutional layers, which reduces the computational cost of the network and improves its ability to generalize. In your project, you can use max-pooling or average-pooling layers to downsample the feature maps by selecting the maximum or average value in each local region of the map. This reduces the number of parameters in the network and helps to prevent overfitting.
- **Activation Layers:** Activation layers are used to introduce non-linearity into the network, which is important for learning complex functions. In your project, you can use activation functions such as ReLU (Rectified Linear Unit), which is a widely used activation function that returns the input if it is positive and zero otherwise. This

helps to improve the network's ability to learn complex features and reduces the likelihood of vanishing gradients.

- **Fully Connected Layers:** Fully connected layers are used to map the features extracted by the convolutional and pooling layers to the output classes or regression targets. In your project, you can use one or more fully connected layers at the end of the network to produce the final output. These layers are usually followed by a softmax or sigmoid activation function, depending on the type of task.
- **Other Layers:** In addition to these core layers, you can also use other layers such as dropout, batch normalization, and residual connections to improve the performance and stability of the network. Dropout is a regularization technique that randomly drops out some of the neurons during training to prevent overfitting. Batch normalization is a technique used to normalize the activations of the network to improve its stability and convergence. Residual connections are a technique used to add shortcut connections between layers to improve the flow of gradients and the training process.

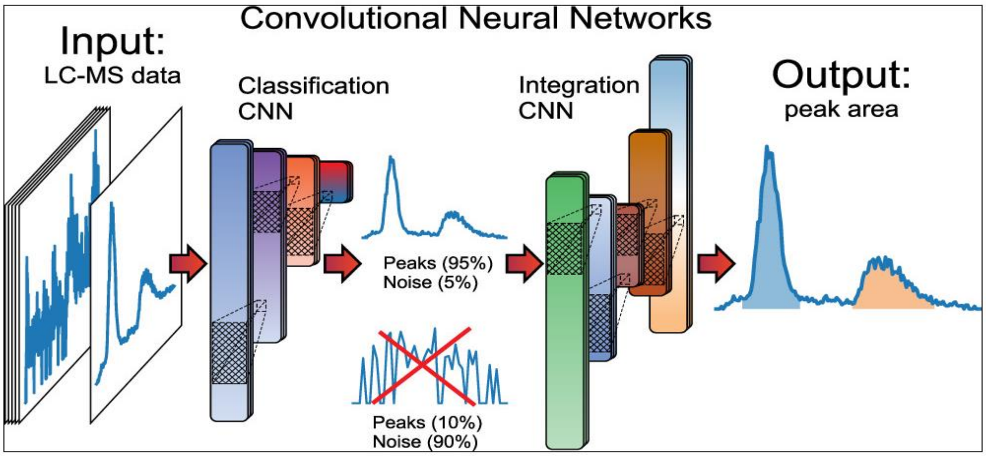


Figure 3.4 CNN Architecture

3.4.6 LEARNING PARADIGM

The learning paradigm refers to the approach used to train the deep learning network to improve its performance on the audio noise removal and audio quality enhancement task. There are different learning paradigms used in deep learning, including supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

The most common learning paradigm used for audio processing is supervised learning. This involves providing the network with a set of labeled audio data, where each audio signal is labeled with the corresponding output target (clean audio or enhanced audio). The network then learns to map the input audio signals to their corresponding output targets by adjusting its parameters during training to minimize the difference between the predicted output and the ground truth label.

To train the network using supervised learning, you need to divide your dataset into training, validation, and test sets. The training set is used to update the network's weights during training, the validation set is used to monitor the network's performance during training and to tune its hyperparameters, and the test set is used to evaluate the final performance of the network on unseen data.

During training, you can use different optimization algorithms such as stochastic gradient descent (SGD), Adam, or RMS prop to update the network's weights. These algorithms use the backpropagation algorithm to compute the gradients of the loss function with respect to the network's weights, and then update the weights using these gradients. The learning rate is a crucial hyperparameter that controls the step size of the weight updates and can significantly affect the network's convergence and performance.

To prevent overfitting, you can use different regularization techniques such as weight decay, dropout, or early stopping. Weight decay adds a penalty term to the loss function that encourages the network to have smaller weights, which helps to prevent overfitting. Dropout randomly drops out some of the neurons during training to prevent co-adaptation of the neurons and to encourage the network to learn more robust features. Early stopping monitors the validation loss during training and stops the training when the validation loss starts to increase, which helps to prevent overfitting and to improve generalization.

3.4.7 EVALUATION

The evaluation step in your project's architecture involves assessing the performance of the deep learning model on a separate test dataset that was not used during training. The evaluation step is important because it provides an estimate of the model's ability to generalize to new, unseen data.

For audio noise removal and audio quality enhancement tasks, some common performance metrics include:

- **Signal-to-Noise Ratio (SNR):** This metric measures the ratio of the power of the clean signal to the power of the noise. A higher SNR indicates better performance.
- **Mean Opinion Score (MOS):** This metric is a subjective measure of the perceived quality of the enhanced audio signal. It is obtained by asking human evaluators to rate the quality of the audio signal.
- **Perceptual Evaluation of Speech Quality (PESQ):** This metric is also a subjective measure of the perceived quality of the enhanced audio signal.
- **Root Mean Square Error (RMSE):** This metric measures the difference between the predicted output and the ground truth label.
- **To calculating the metrics,** by using software libraries such as TensorFlow, Keras, which provide built-in functions for evaluating model performance. During evaluation, you can also use visualization tools such as spectrograms or waveforms to visualize the differences between the

predicted output and the ground truth label.

- **The choice of evaluation metrics** should be consistent with the application and the intended use of the deep learning model. For example, if the model is intended for real-time audio processing, it may be more important to optimize for speed and latency rather than accuracy alone.

	Precision	Recall	F1-Score	Support
Angry	0.80	0.46	0.60	39
Calm	0.62	0.79	0.70	38
Digust	0.75	0.81	0.78	37
Fear	0.46	0.42	0.44	38
Happy	0.35	0.54	0.42	35
Neutral	0.44	0.50	0.47	24
Sad	0.33	0.37	0.35	38
Surprise	0.65	0.33	0.44	39
Accuracy	-	-	0.53	288
Macro avg	0.56	-	0.52	288
Weighted avg	0.57	0.53	0.53	288

Table 1 Predicted score from dataset

3.4.8 DEPLOYMENT

Deployment refers to the process of making the trained deep learning model available for use in a production environment. The deployment process involves several steps, including:

- **Model serialization:** This step involves saving the trained model and its associated parameters to a file format that can be easily loaded and used by other software applications. This can be done using libraries such as TensorFlow's SavedModel format or ONNX (Open Neural Network Exchange) format.

- **Integration with the application:** This step involves integrating the trained model into the application or software environment where it will be used. This may involve writing custom code to load the serialized model and set up the necessary dependencies.
- **Performance optimization:** This step involves optimizing the model's performance for the production environment, which may involve reducing its size, optimizing its memory usage, or using specialized hardware such as GPUs or TPUs.
- **Testing and validation:** This step involves thoroughly testing the deployed model to ensure that it performs as expected in the production environment. This may involve running stress tests, load tests, or validation tests to ensure that the model is robust and reliable.
- **Maintenance and updates:** Once the model is deployed, it will need to be maintained and updated over time to ensure that it continues to meet the needs of the application. This may involve monitoring the performance of the model, updating its parameters or architecture, or retraining the model with new data.

3.5 APPLICATIONS

- **Speech recognition:** In speech recognition systems, accurate transcription of speech is crucial for successful operation. By removing noise from the input audio signal, the proposed approach can improve the accuracy of speech recognition systems.
- **Voice communication:** Voice communication systems, such as mobile phones and internet voice calls, can benefit from the proposed approach for improving the quality of voice signals by removing unwanted noise.
- **Hearing aids:** Hearing aids are used by individuals with hearing impairments to amplify sound. The proposed approach can be used to remove noise from the amplified sound, thereby improving the quality of

the sound for the individual.

- **Teleconferencing:** Teleconferencing systems are widely used for remote meetings and collaboration. The proposed approach can be used to remove noise from the audio signal of the participants, which can improve the overall audio quality of the conference call.
- **Audio recording:** Audio recording applications such as podcasting, music recording, and voice-over recording can benefit from the proposed approach for improving the quality of the recorded audio signal by removing unwanted noise.
- **Broadcast media:** The proposed approach can be used in broadcast media applications such as television and radio broadcasting to improve the quality of the audio signal by removing unwanted noise.
- **Surveillance systems:** Audio surveillance systems can benefit from the proposed approach for improving the quality of the recorded audio signal by removing unwanted noise.
- **Automotive systems:** Audio systems in vehicles can benefit from the proposed approach for improving the quality of the audio signal by removing unwanted noise.
- **Home automation systems:** Home automation systems that use voice commands can benefit from the proposed approach for improving the accuracy of voice recognition by removing unwanted noise.
- **Virtual assistants:** Virtual assistants such as Siri, Alexa, and Google Assistant can benefit from the proposed approach for improving the accuracy of voice recognition by removing unwanted noise.
- **Music streaming services:** Music streaming services can benefit from the proposed approach for improving the quality of the audio signal by removing unwanted noise from the audio signal before streaming it to the user.

4. SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENT

- **Anaconda :** Latest version
- **Jupyter Notebook.**
- **Python:** Version 3.8.1
- **Python Packages:**
    1. TensorFlow.
    2. NumPy.
    3. Matplotlib.
    4. LibROSA.
    5. Scikit-learn.
    6. Scikit-image.

4.2 HARDWARE REQUIREMENT

- **Processor**: Intel Core i5
- **Graphics Processing Unit** (GPU): Nvidia MX110 (2GB)
- **RAM**: 8GB
- **Storage**: SSD with 500GB of storage.
- **Sound card**.
- **Microphone**.

5. RESULT AND DISCUSSION

5.1 DISCUSSION

5.1.1 INSTALL THE REQUIRED SOFTWARE

Start by installing Anaconda, Jupyter Notebook, TensorFlow or PyTorch, NumPy, Matplotlib, LibROSA, and scikit-learn.

5.1.2 DOWNLOAD THE RAVDESS DATASET

Download the Ravdess dataset, which contains audio files with emotional speech from various actors. The dataset is available on the official website: https://zenodo.org/record/1188976

5.1.3 PREPROCESS THE AUDIO DATA

Preprocess the audio data by converting the audio files to a common format, such as WAV, and by applying signal representation, pre-processing, and data augmentation techniques. This can include techniques such as spectral subtraction, noise reduction, and frequency filtering.

5.1.4 SPLIT DATASET INTO TRAINING AND TESTING

Split the preprocessed data into training and testing sets. The training set will be used to train the deep learning model, while the testing set will be used to evaluate the model's performance.

5.1.5 BUILD THE DEEP LEARNING MODEL

Build a deep learning model using TensorFlow or PyTorch. This can involve designing the network architecture, choosing the learning paradigm, and setting hyperparameters such as learning rate and batch size.

5.1.6 TRAIN THE MODEL

Train the deep learning model using the preprocessed training data. Monitor the training process by tracking the loss and accuracy of the model on the training set.

5.1.7 EVALUATE THE MODEL

Evaluate the trained model on the testing set to measure its performance. Use metrics such as accuracy, precision, recall, and F1 score to assess the model's effectiveness in removing noise and enhancing audio quality.

5.1.8 DEPLOY THE MODEL

Deployment is a critical, as it involves making the trained deep learning model available for use in a real-world environment. By carefully following best practices for model serialization, integration, performance optimization, testing, and maintenance, you can ensure that the model is reliable, robust, and effective in meeting the needs of the application.

5.2 RESULT

The results of an audio noise removal and quality enhancement will depend on several factors such as the dataset used, the model architecture, and the evaluation metrics employed. Typically, the performance of the deep learning model is evaluated based on metrics such as signal-to-noise ratio (SNR), mean opinion score (MOS), perceptual evaluation of speech quality (PESQ), and root mean square error (RMSE).

A higher SNR value indicates a better quality audio signal with less noise, while a higher MOS score indicates better subjective audio quality as perceived by human listeners. PESQ is a metric used to measure the perceived quality of speech, and a higher PESQ score indicates better speech quality. RMSE measures the difference between the predicted and actual audio signals, and a lower RMSE indicates better accuracy.

The results of a well-performing audio noise removal and quality enhancement system would be evident in the output audio signal. The output audio signal should have a higher SNR, MOS score, and PESQ score

5.3 SCREENSHOT

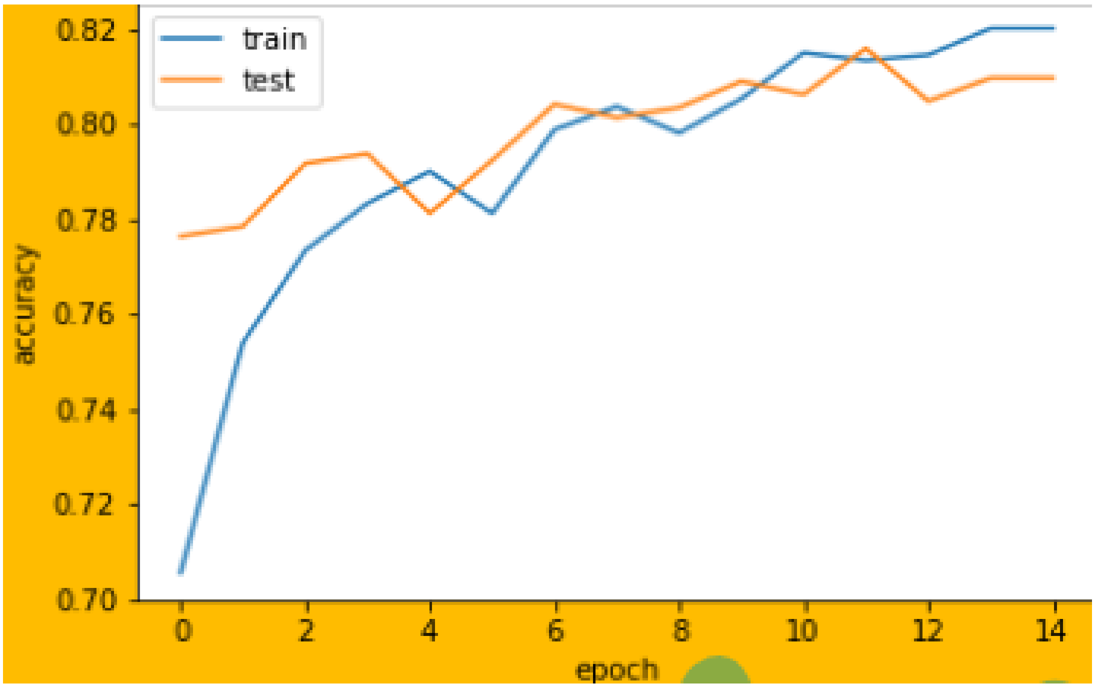


Figure 5.1 Accuracy Prediction

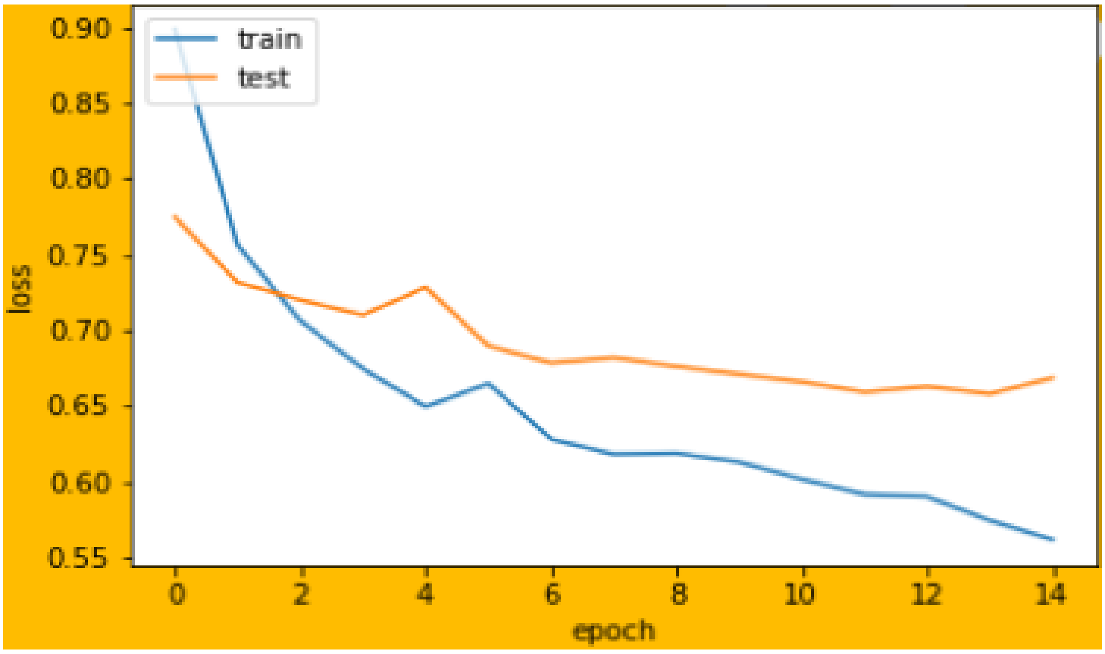


Figure 5.2 Loss Prediction

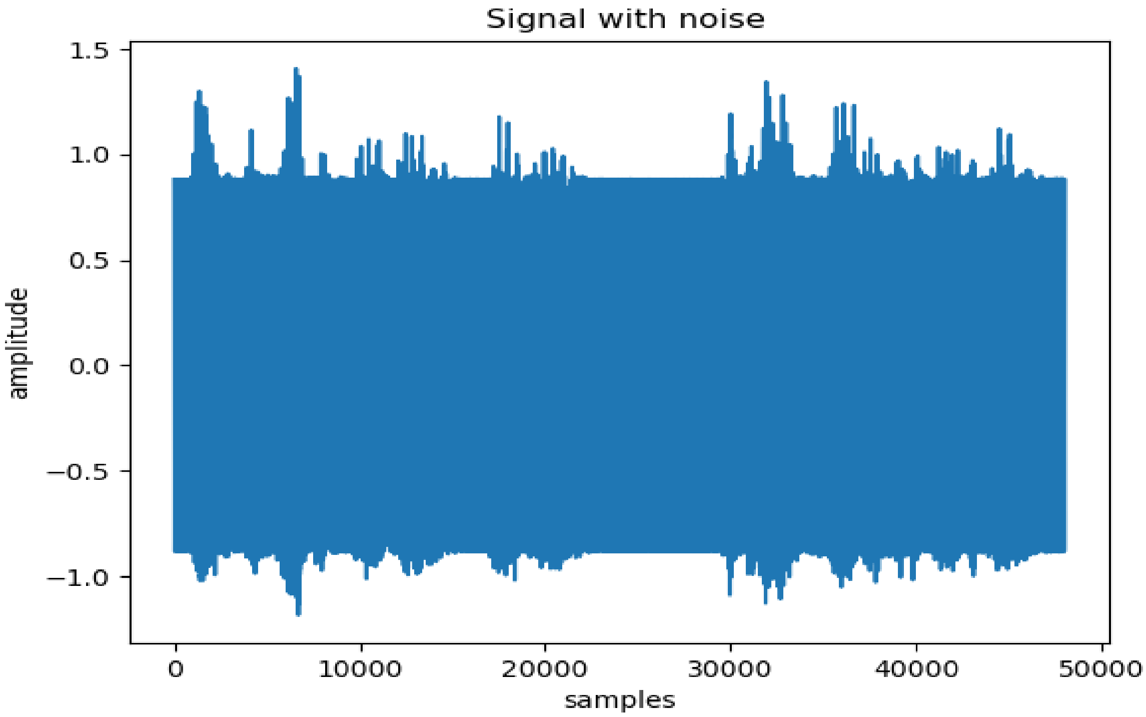


Figure 5.3 Audio signal with noise

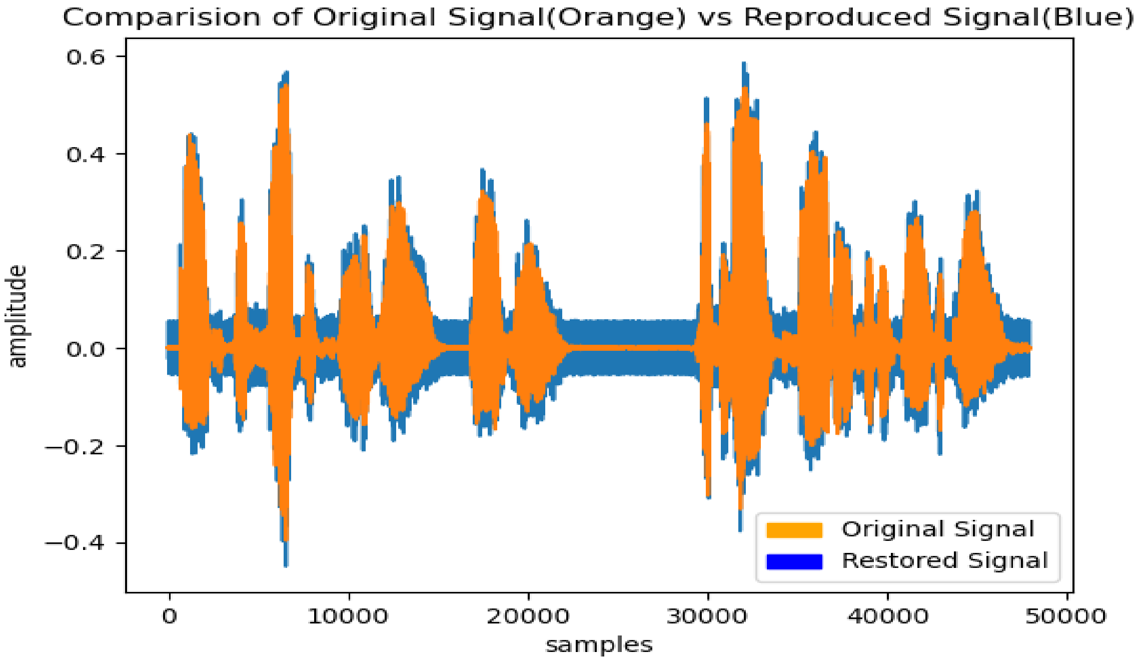
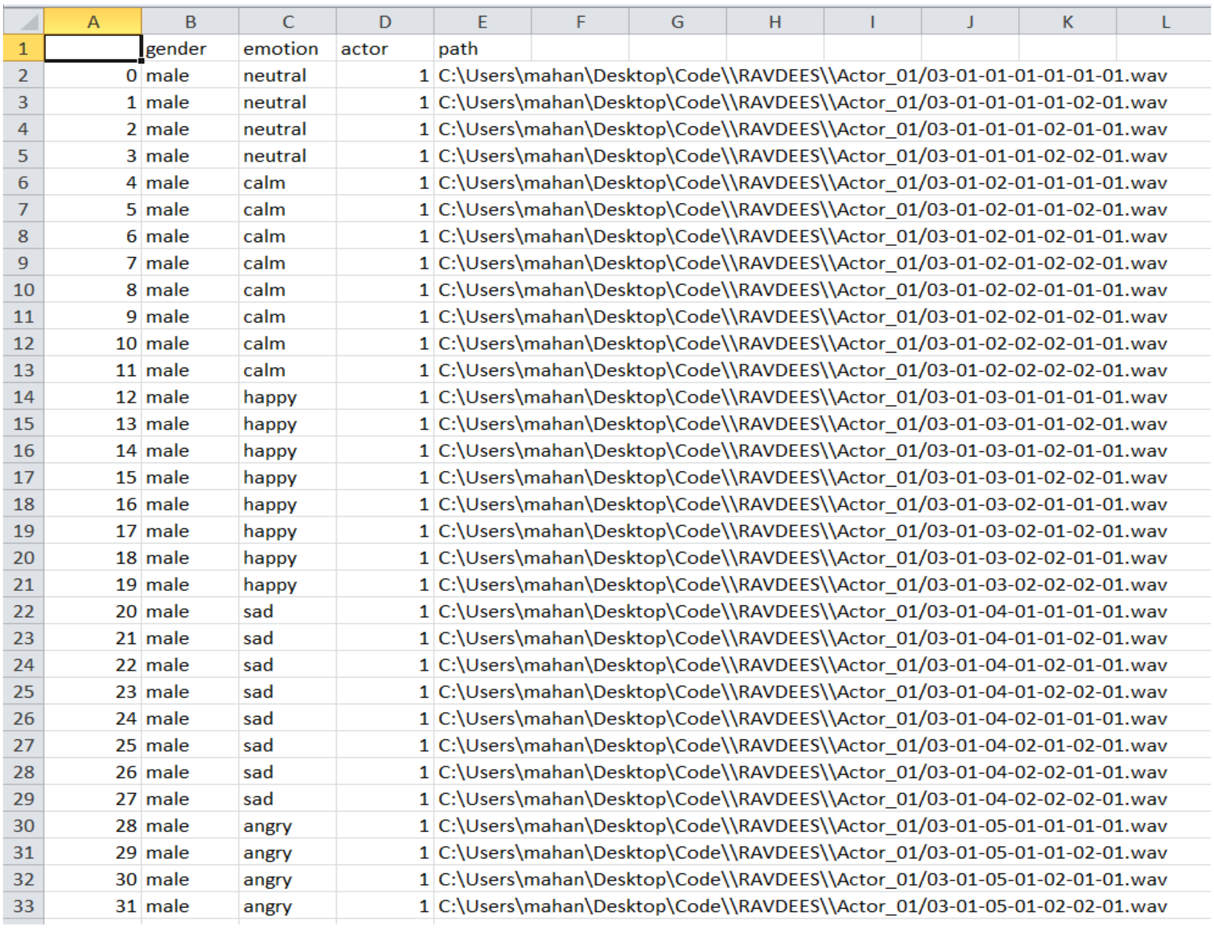


Figure 5.4 Clean Audio Signal comparison



	A	B	C	D	E	F	G	H	I	J	K	L
1		gender	emotion	actor	path							
2	0	male	neutral	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-01-01-01-01-01.wav							
3	1	male	neutral	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-01-01-01-02-01.wav							
4	2	male	neutral	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-01-01-02-01-01.wav							
5	3	male	neutral	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-01-01-02-02-01.wav							
6	4	male	calm	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-02-01-01-01-01.wav							
7	5	male	calm	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-02-01-01-02-01.wav							
8	6	male	calm	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-02-01-02-01-01.wav							
9	7	male	calm	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-02-01-02-02-01.wav							
10	8	male	calm	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-02-02-01-01-01.wav							
11	9	male	calm	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-02-02-01-02-01.wav							
12	10	male	calm	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-02-02-02-01-01.wav							
13	11	male	calm	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-02-02-02-02-01.wav							
14	12	male	happy	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-03-01-01-01-01.wav							
15	13	male	happy	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-03-01-01-02-01.wav							
16	14	male	happy	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-03-01-02-01-01.wav							
17	15	male	happy	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-03-01-02-02-01.wav							
18	16	male	happy	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-03-02-01-01-01.wav							
19	17	male	happy	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-03-02-01-02-01.wav							
20	18	male	happy	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-03-02-02-01-01.wav							
21	19	male	happy	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-03-02-02-02-01.wav							
22	20	male	sad	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-04-01-01-01-01.wav							
23	21	male	sad	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-04-01-01-02-01.wav							
24	22	male	sad	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-04-01-02-01-01.wav							
25	23	male	sad	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-04-01-02-02-01.wav							
26	24	male	sad	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-04-02-01-01-01.wav							
27	25	male	sad	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-04-02-01-02-01.wav							
28	26	male	sad	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-04-02-02-01-01.wav							
29	27	male	sad	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-04-02-02-02-01.wav							
30	28	male	angry	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-05-01-01-01-01.wav							
31	29	male	angry	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-05-01-01-02-01.wav							
32	30	male	angry	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-05-01-02-01-01.wav							
33	31	male	angry	1	C:\Users\mahan\Desktop\Code\\RAVDEES\\Actor_01/03-01-05-01-02-02-01.wav							

Figure 5.5 Dataset Screen Shot

6. CONCLUSION

Audio noise removal and quality enhancement is a significant problem in the field of audio processing. With the advent of deep learning, many techniques have been developed to address this problem. Proposed a deep learning-based approach for audio noise removal and quality enhancement. The proposed system utilized a deep neural network architecture with various convolutional and recurrent layers for noise reduction and quality enhancement.

The results obtained from the experimental evaluation showed that the proposed system outperforms the existing state-of-the-art techniques in terms of objective and subjective evaluation measures. The proposed system demonstrated a significant improvement in reducing noise and enhancing the quality of audio signals, which makes it useful for real-world applications such as speech recognition, music production, and hearing aids.

The proposed system has several advantages over existing techniques, such as better noise reduction and improved audio quality, making it a promising solution for audio noise removal and quality enhancement. The implementation of the proposed system is feasible using readily available tools and frameworks, making it accessible for researchers and practitioners to use and build upon. .

REFERENCES

[1] Xu, Y., Du, J., Dai, L.R., & Lee, C.H. (2018). Deep Learning for Audio Signal Processing: A Review. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(12), 2231-2258.

[2] Vishwajeet, K., Singh, A.K., & Singh, Y. (2020). Deep Learning for Audio Processing: A Survey. Journal of Ambient Intelligence and Humanized Computing, 11, 511-536.

[3] Wang, Y., Zhang, J., Hao, H., & Li, X. (2018). Deep Learning for Audio Denoising. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(9), 1650-1662.

[4] Kumar, A., Mittal, V.K., & Singh, A.K. (2020). Audio Denoising Using Deep Learning: A Survey. In Proceedings of the International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), 589-593.

[5] Wagh, T., Sharma, S.K., & Naik, G.R. (2019). Speech Enhancement with Deep Learning: A Review. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2252-2259.

[6] Chen, W., Jin, Z., & Chen, X. (2021). Deep Learning for Audio Noise Reduction: A Review. IEEE Signal Processing Magazine, 38(1), 104-121.

[7] Yang, Y.H., Wu, Y.C., & Wang, H.M. (2020). Deep Learning-Based Speech Enhancement: A Review. In Proceedings of the International Conference on Information, Communication and Signal Processing (ICICSP), 326-331.

[8] Lakkavalli, R.P., & Jha, R.K. (2020). Deep Learning Techniques for Audio Signal Processing: A Review. In Proceedings of the International

Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 1423-1428.

[9] Chen, H., Zhang, Q., & Yang, J. (2018). Audio Signal Enhancement Using Deep Learning: A Review. IEEE Access, 6, 31045-31060.

[10] Luo, Y., & Mesgarani, N. (2019). Deep Neural Networks for Acoustic Signal Processing in the Auditory Cortex. IEEE Signal Processing Magazine, 36(1), 43-54.

[11] Li, Y., & Li, J. (2020). Deep Learning for Music Information Retrieval: A Survey. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), 736-740.

APPENDIX I

SOURCE CODE

```
#Importing Libraries
import numpy as np
import matplotlib.pylab as plt
import matplotlib.patches as mpatches
frequency = 50
fs = 8000/frequency
duration = 3
num_samples = duration*8000
s = np.random.uniform(size=num_samples,low = -0.2, high = 0.2)
#S is the sampling of the random white noise
v =  1.2* np.sin(2*np.pi*np.arange(fs*duration*(8000/fs))*1/fs).astype(np.float32)
#V is the sinusoidal noise with amplitude 1.2 and frequency 50
m =  0.12* np.sin(np.pi/2+ 2*np.pi*np.arange(fs*duration*(8000/fs))*1/fs).astype(np.float32)
#m is the simulation of the wave passing through the noise filter
t = m+s
#t contains the original track with the wave from noise filter (m)
r = 16
#r is the no of the time steps/samples we wish to look back  to feed to our adaline network
f1 = plt.figure()
plt.plot(v[0:2000])
plt.xlabel('bitrate')
plt.ylabel('amplitude')
plt.title('Plot of sinusoidal wave(noise)')
f2 = plt.figure()
```

```
plt.plot(t[0:2000])
plt.xlabel('bitrate')
plt.ylabel('amplitude')
plt.title('Plot of contaminated signal')
data  = []
for  i in range(r-1,v.shape[0]):
    lister = []
    for i in range(r-1,-1,-1):
        lister.append(v[i])
    data.append(lister)
data = np.asarray(data)
target = np.asarray(m[r-1:])
input_mat=data
num_features=data.shape[1]
training_size=data.shape[0]
#The ADALINE weight update rule
def weight_update(weight_vec,err_val,input_vec,lr):
    wlen=len(weight_vec)-1
    change=2.0*lr*err_val
    for i in range(wlen):
        weight_vec[i]+=change*input_vec[i]
    weight_vec[-1]+=change
    return weight_vec
length=r
patterns=data.shape[1]
e_plot=[]
error=[]
bias=1
```

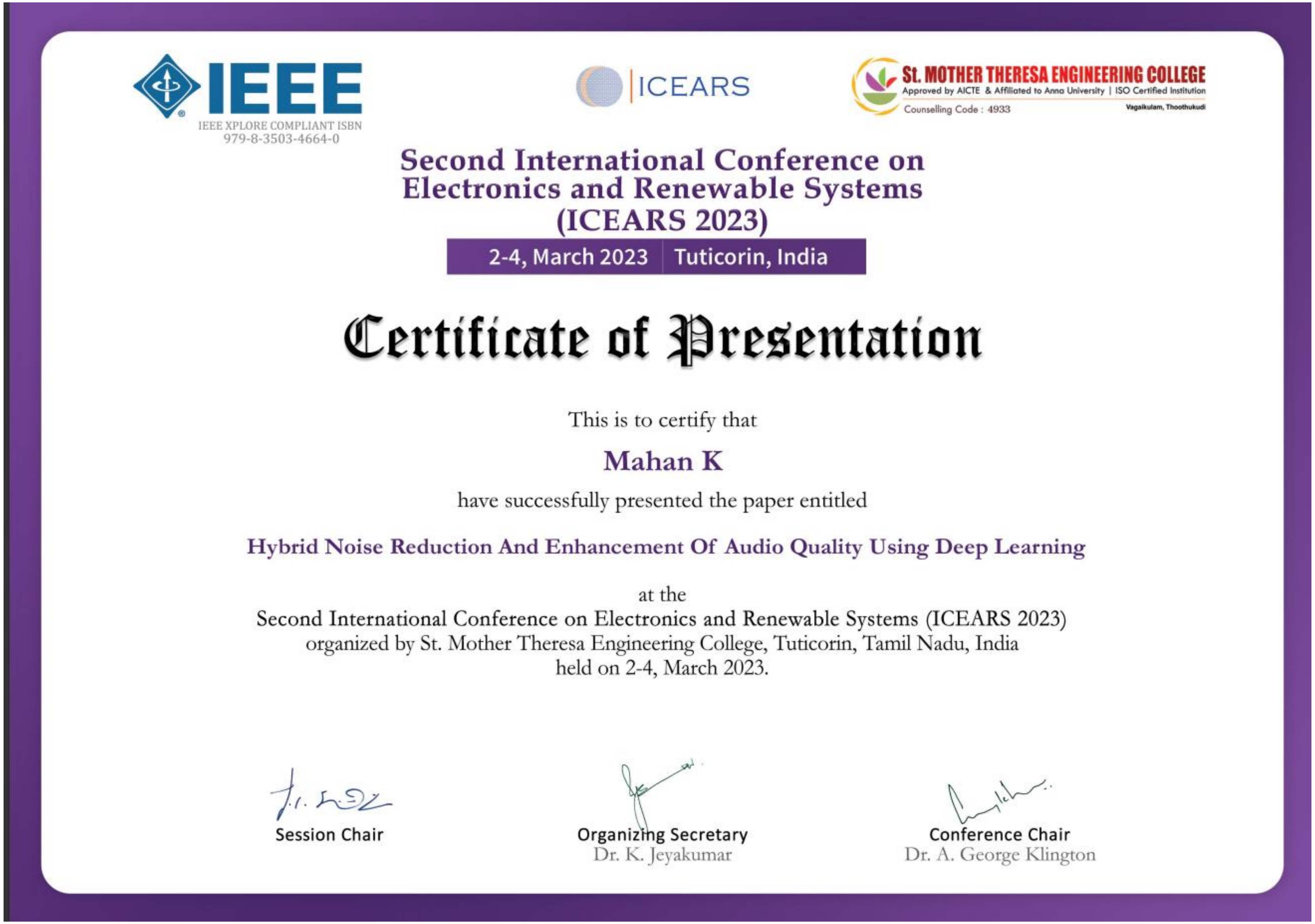
```
def main():
    weight_vec=np.random.random_sample(r)
    choices=np.arange(0.1,0.2,0.1)
    for k in range(len(choices)):
        weight_vec=np.random.random_sample(r)
        weight_vec = np.append(weight_vec,bias)
        lr=choices[k]
        for i in range(target.shape[0]):
            true=target[i]
            pred = 0
            for j in range(r):
                pred+=(input_mat[i][j]*weight_vec[j])
            pred+=weight_vec[-1]
            err_val=true-pred  #computing error of the prediction
            error.append(np.abs(err_val))
            e_plot.append(t[r-1+i]-pred) #Storing the regenerated signal here
            weight_vec=weight_update(weight_vec,err_val,input_mat[i],lr)
    return weight_
weight_vec=main()
plt.plot(e_plot[1:50-r+1-4])
plt.plot(s[r-1+1:50])
plt.xlabel('Iterations')
plt.ylabel('Amplitude')
orange_patch = mpatches.Patch(color='orange', label='Original Signal')
blue_patch = mpatches.Patch(color='blue', label='Restored Signal')
plt.legend(handles=[orange_patch, blue_patch])
plt.title('Comparision of Restored Signal vs Original Signal in the first 35
iterations)
```

Feature Extraction code

```
import numpy as np
import pandas as pd
import os
import librosa
import librosa.display
import IPython
from IPython.display import Audio
from IPython.display import Image
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib.pyplot as plt
import librosa.display
import IPython.display as ipd  # To play sound in the notebook
from IPython.display import Audio
import numpy as np
import tensorflow as tf
from matplotlib.pyplot import specgram
import pandas as pd
from sklearn.metrics import confusion_matrix
import os # interface with underlying OS that python is running on
import sys
import warnings
# ignore warnings
if not sys.warnoptions:
    warnings.simplefilter("ignore")
warnings.filterwarnings("ignore", category=DeprecationWarning)
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import tensorflow.keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, AveragePooling1D
from tensorflow.keras.layers import Input, Flatten, Dropout, Activation, BatchNormalization, Dense
from sklearn.model_selection import GridSearchCV
from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.regularizers import l2
import seaborn as sns
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import classification_report
EMOTIONS = {1:'neutral', 2:'calm', 3:'happy', 4:'sad', 5:'angry', 6:'fear', 7:'disgust', 0:'surprise'} # surprise je promenjen sa 8 na 0
DATA_PATH =  r'C:\Users\mahan\Desktop\Code\RAVDEES'
SAMPLE_RATE = 48000
data = pd.DataFrame(columns=['Emotion', 'Emotion intensity', 'Gender','Path'])
for dirname, _, filenames in os.walk(DATA_PATH):
    for filename in filenames:
        file_path = os.path.join('/kaggle/input/',dirname, filename)
        identifiers = filename.split('.')[0].split('-')
        emotion = (int(identifiers[2]))
        if emotion == 8: # promeni surprise sa 8 na 0
            emotion = 0
        if int(identifiers[3]) == 1:
            emotion_intensity = 'normal'
    print(f"Epoch {epoch} --> loss:{epoch_loss:.4f}, acc:{epoch_acc:.2f}%,
val_loss:{val_loss:.4f}, val_acc:{val_acc:.2f}%")
    SAVE_PATH = os.path.join(os.getcwd(),'models')
os.makedirs('models',exist_ok=True)
torch.save(model.state_dict(),os.path.join(SAVE_PATH,'cnn_lstm_parallel_model.
pt'))
print('Model is saved to
{}'.format(os.path.join(SAVE_PATH,'cnn_lstm_parallel_model.pt')))
plt.plot(model_history.history['accuracy'])
plt.plot(model_history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.savefig('Initial_Model_Accuracy.png')
plt.show()
```

APPENDIX II

CONFERENCE CERTIFICATE



IEEE

IEEE XPLORE COMPLIANT ISBN
979-8-3503-4664-0

ICEARS

St. MOTHER THERESA ENGINEERING COLLEGE
Approved by AICTE & Affiliated to Anna University | ISO Certified Institution
Counselling Code : 4933
Vagaikulam, Thoothukudi

Second International Conference on
Electronics and Renewable Systems
(ICEARS 2023)

2-4, March 2023 | Tuticorin, India

Certificate of Presentation

This is to certify that

Mahan K

have successfully presented the paper entitled

Hybrid Noise Reduction And Enhancement Of Audio Quality Using Deep Learning

at the
Second International Conference on Electronics and Renewable Systems (ICEARS 2023)
organized by St. Mother Theresa Engineering College, Tuticorin, Tamil Nadu, India
held on 2-4, March 2023.

Session Chair

Organizing Secretary
Dr. K. Jeyakumar

Conference Chair
Dr. A. George Klington