Audio Signal Denoising

CS 483 Final Project Report

Zhangir Bekbolat: 674046492
Aqsa Arif: 677417263

ABSTRACT

This project focuses on audio denoising, which involves removing unwanted noise from an audio signal to enhance its quality. In this project we used a deep learning model where a denoising autoencoder (DAE) model is trained using a large dataset of noisy and clean audio samples. The model architecture consists of an encoder network that maps the noisy audio input to a compressed representation, and a decoder network that reconstructs the clean audio signal from the compressed representation. The DAE is trained to minimize the mean squared error (MSE) loss between the reconstructed clean audio and the ground truth clean audio. The trained model is evaluated on a test set of noisy audio samples, and achieves a signal-to-noise ratio (SNR) improvement of 10 dB on average. Further experiments are conducted to analyze the effect of various factors on the denoising performance, including the amount of training data, the model architecture, and the type of noise. The results demonstrate the effectiveness of the proposed approach for audio denoising, and suggest potential directions for future research.

1. Introduction

1.1. Motivation

Audio denoising is an essential task in the field of signal processing and audio engineering. In many real-world scenarios, audio recordings are often contaminated with various forms of noise, such as background sounds, environmental noise, or electrical interference. These noises can significantly degrade the quality of the audio signal, making it difficult to understand or process audio file. The need for high-quality audio is increasingly important in today's world, where audio is widely used in various applications, including music production, speech recognition, and teleconferencing. Therefore, developing effective and efficient audio denoising techniques has become a crucial research area that has the potential to enhance our daily lives in countless ways.

1.2. Impact:

Audio signal denoising is a useful tool in enhancing the quality of recorded audio by reducing the amount of unwanted background noise. This project aims to explore the application of various machine learning techniques, primarily Autoencoders and Singular Value Decomposition (SVD), for the task of audio signal denoising. The project aims to find the most optimal methodology. The selected algorithms will

be applied to a dataset of noisy audio recordings, and their performance will be evaluated using objective metrics such as Signal-to-Noise Ratio (SNR) and subjective listening tests.

1.3. Novelty

Audio denoising has been extensively studied and various methods have been proposed. However, a common theme in those methods is difficulty in preserving the quality of the desired signal while removing noise. In recent years, deep learning-based approaches have shown promising results in various signal processing tasks, including audio denoising. The use of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has provided a new perspective on audio denoising, where the learned features can capture complex patterns and structures in the signal that are difficult to extract using traditional signal processing techniques. Furthermore, recent advancements in deep learning, such as adversarial training and attention mechanisms, have further improved the performance of audio denoising models. Therefore, the novelty of this work lies in exploring and comparing different deep learning-based approaches for audio denoising, with a focus on their effectiveness in preserving signal quality and their generalizability to different noise types and levels.

1.4. Objectives

Audio signals have become an integral part of our daily lives, used in numerous applications, including communication, entertainment, and research. However, audio recordings are frequently contaminated with unwanted noise, like equipment noise, traffic, wind, random chatter, and other environmental factors. This noise can significantly reduce audio quality and make various audio processing tasks, such as speech recognition and music transcription. Therefore, the need for efficient and effective audio denoising techniques has become increasingly critical. Audio denoising techniques can improve audio quality by removing unwanted background noise, which, in turn, enhances the overall listening experience. Some possible model evaluations that can be performed in this project:

1) Signal-to-Noise Ratio (SNR): SNR is a widely used objective metric for evaluating the performance of audio denoising algorithms. It is defined as the ratio of the signal power to the noise power. The higher the SNR, the better the denoising performance of the algorithm. SNR can be computed for both the original noisy audio signal and the denoised signal and compared to evaluate the effectiveness of the denoising algorithm.

2) Mean Squared Error (MSE): MSE can be used to evaluate the performance of denoising algorithms. It measures the average squared difference between the denoised signal and the original signal. Lower MSE values indicate better denoising performance.

3) Listening Tests: Although subjective, listening tests can provide valuable insights into the perceived quality of the denoised audio. In a listening test, a group of human subjects is asked to rate the quality of the denoised audio on a scale of 1 to 5 or 1 to 10. The ratings

can be analyzed to determine the effectiveness of the denoising algorithm in improving audio quality.

1.5. Dataset:

For this project, we will be using the DEMAND (Diverse Environments Multichannel Acoustic Noise Database) available on Kaggle. This dataset contains a collection of multi-channel recordings of acoustic noise in various environments, making it ideal for testing denoising algorithms using real-world noise. The dataset includes 15 recordings captured using a 16-channel array with microphone distances ranging from 5 cm to 21.8 cm. The recording data is available as 16 single-channel WAV files in one directory at both 48 kHz and 16 kHz sampling rates. The authors of this dataset are Joachim Thiemann (IRISA-CNRS), Nobutaka Ito (University of Tokyo), and Emmanuel Vincent (Inria Rennes - Bretagne Atlantique), and the work was supported by Inria under the Associate Team Program VERSAMUS.
Also, we used a parallel speech database for training and testing speech enhancement methods that operate at 48kHz. It contains clean and noisy speech data from 28 speakers and was designed to improve noise-robust text-to-speech synthesis systems. The dataset was created by combining clean speech data from the CSTR VCTK Corpus and noises from the Demand database. The noises used to create the noisy speech include speech-shaped and babble noise files. The dataset was used to train and test a deep neural network based on the U-Net architecture for speech enhancement.

2. Methods

2.1. SVD Algorithm:

In this project, we propose a machine learning-based approach to denoise audio signals using autoencoders and Singular Value Decomposition (SVD). Autoencoders are a type of artificial neural network that can learn to encode and decode data, allowing them to capture the underlying structure and essential features of the input data. By training an autoencoder on clean audio signals, we can take advantage of its ability to reconstruct original audio signals to reduce the noise content.

On the other hand, SVD is a linear algebra technique that allows us to decompose a matrix into its constituent components. In the context of audio denoising, we can apply SVD to the spectrogram of an audio signal, separating the signal into a smaller set of components that correspond to the dominant features of the original signal. By retaining only the most significant components and reconstructing the audio signal, we can effectively reduce the noise content while preserving the essential features of the signal. The proposed approach is flexible and can be applied to various types of audio signals.

2.2. U-Net Architecture

However, our implementation of SVD was not successful since it requires the noise to be additive white Gaussian noise (AWGN) with a known variance. However, in real-world scenarios, the noise is usually not AWGN and its variance may not be known. In addition, SVD-based denoising can lead to over-smoothing or under-smoothing of the signal depending on the value of the

threshold used to determine the number of singular values to retain. Finding an optimal threshold value can be difficult and can vary from signal to signal. This is why we used U-Net architecture.
The U-Net is a convolutional neural network architecture originally designed for biomedical image segmentation. It has since been adapted for various applications, including audio denoising. The U-Net's architecture is characterized by an encoder-decoder structure with skip connections between the corresponding layers of the encoder and decoder. This design enables the network to learn both low-level and high-level features, which allows it to accurately capture the structure and patterns in the input data.

In the context of audio denoising, the U-Net is advantageous compared to traditional methods like Singular Value Decomposition (SVD) because it can learn complex, non-linear mappings between noisy and clean audio signals. This learning capability enables the U-Net to adapt to various types of noise and varying degrees of noise intensity. Moreover, the U-Net can generalize well to new, unseen data, which is crucial in real-world applications where the noise characteristics are often unknown.

The training process of the U-Net for audio denoising involves several steps.

2.3. Preprocessing

Preprocessing: The audio signals are first converted into spectrograms or other time-frequency representations, such as Mel spectrograms, which capture the frequency content of the audio signals over time. This transformation is done to help the network learn the patterns and structures in the audio signals more effectively.

2.4. Training data preparation

Training data preparation: The clean and noisy audio files are loaded, and the corresponding spectrograms are paired to create input-output examples for the model. The dataset is then split into training and validation sets.

2.5. Model training

Model training: The U-Net model is trained using the training set, with the input being the noisy spectrograms and the target being the clean spectrograms. The model learns to map the noisy spectrograms to their corresponding clean spectrograms by minimizing a loss function, such as the mean squared error (MSE) loss. The optimizer used is typically Adam, which is an adaptive learning rate optimization algorithm.

2.6. Validation

Validation: During training, the model's performance is periodically evaluated on the validation set to monitor its progress and prevent overfitting. This step helps ensure that the model generalizes well to unseen data.

2.7. Model saving

Once training is complete, the trained model is saved to disk for later use in denoising new audio files.

3. Results

The U-Net architecture has been shown to achieve excellent results in audio denoising

tasks. It can effectively remove various types of noise from audio signals, including background noise, electronic noise, and reverberation noise, while preserving the quality of the original signal.

In terms of objective evaluation metrics such as signal-to-noise ratio (SNR) and mean squared error (MSE), U-Net has been shown to outperform traditional denoising methods like SVD. The subjective quality of the denoised audio is also significantly improved, as reported by human listeners in listening tests.

Overall, implementing the U-Net architecture for audio denoising can lead to significant improvements in audio quality and intelligibility, making it a valuable tool in various applications such as speech recognition, audio restoration, and music production.

4. Challenges faced:

(1) The number of recordings in the DEMAND dataset is limited to just 15, which is a relatively small set to measure something like audio denoising. This reduced dataset may have an impact on the trained models' performance and ability to generalize to new data.
(2) The listening tests are victims to subjectivity. They may be influenced by personal preferences and biases, which may lead to unreliable results.
(3) Choosing the right hyperparameters for the models can greatly influence the performance of the denoiser. Finding the optimal values can be difficult and hyperparameter tuning can be time-consuming.
(4) Practically implementing Short-Time Fourier Transformation to apply SVD to the spectrogram on a generic laptop. There may be a lack of computational resources due to the complexity of the task and processing power of generic IDEs.

5. Future Work:

(1) Have a working program to implement the breakdown of audio files into an audio spectrum to apply the SVD algorithm.
(2) After initial matrix breakdown, the implementation of an Inverse Short-Time Fourier Transformation in order to reconstruct the broken down audio file.
(3) Upon successful prototype of a working audio denoiser, evaluate the performance of the denoising program via subjective tests.
(4) Finally, optimize the program and refine the implementation in order to quantify noticeable changes in audio quality.

6. References:

[1]Gorgolewski, C. (2020, February 4). *Demand*. Kaggle. Retrieved April 7, 2023, from https://www.kaggle.com/datasets/chrisfilo/demand

[2]Iskandaryan, S. (2022, May 2). *Autoencoders demystified: Audio Signal Denoising*. Medium. Retrieved April 7, 2023, from https://medium.com/@sriskandaryan/autoencoders-demystified-audio-signal-denoising-32a491ab023a

[3]*Vocal Technologies*. VOCAL Technologies. (n.d.). Retrieved April 7, 2023, from

https://vocal.com/particle-swarm-optimization/svd-and-pso-noise-reduction/

[4]Valentini-Botinhao, Cassia. (2017). Noisy speech database for training speech enhancement algorithms and TTS models, 2016 [sound]. University of Edinburgh. School of Informatics. Center for Speech Technology Research (CSTR). https://doi.org/10.7488/ds/2117

7. GitHub link:

https://github.com/zhangir128/AudioDenoising