המכללה האקדמית להנדסה בכרמיאל

בראודה

Software Engineering Department
Braude College

Capstone Project Phase A

Speech Denoising: a Noise2Noise Approach

23-2-R-10

Students:

Omri Cohen, omri.cohen2@e.braude.ac.il

Eliav Shabat, Eliav.shabat@e.braude.ac.il

Supervisors

Dr. Renata Avros

Prof. Zeev Volkovich

Table of content

1. Abstract

The document presents a theoretical study focused on removing noise from audio samples. The project was conducted in collaboration with Rafael, with the goal of filtering noise from audio files captured by a BNET walkie-talkie. The purpose of this noise reduction is to facilitate Rafael's future work in speaker identification using machine learning techniques.

The initial chapters provide an in-depth exploration of noise removal using machine learning, emphasizing the problem statement and our proposed solution - the Noise2Noise model. A comprehensive introduction to the denoising technique is presented, accompanied by a mathematical background that covers key concepts such as Deep Neural Networks (DNNs), audio file functions, and loss functions.

The core component of the project revolves around the N2N model. This section delves into the process of constructing databases using the UrbanSound8K dataset, elucidates the utilization of the 'U-Net architecture' within the framework of complex numbers, and explains the training process of the model. Various parameters, including Signal-to-Noise Ratio (SNR), Perceptual Evaluation of Speech Quality (PESQ) [3], and more, are employed during the training phase. An architectural diagram of the model is provided alongside the project's future plans.

Lastly, an iterative model for the testing process is being described, which encompasses multiple iterations of model training and the use of metrics such as noise removal accuracy and speaker identification accuracy percentages. These metrics serve as quantitative measures to evaluate the effectiveness of the developed model.

Overall, this document serves as a detailed account of the theoretical study conducted for noise removal in audio samples. It encompasses a range of topics, from the theoretical foundations to the practical implementation and evaluation of the proposed Noise2Noise model.

The project book will provide readers with a deeper understanding of the challenges associated with denoising audio.

2. Introduction

Before dig-in to noise removal solution let first address the problem – Speaker Recognition within limited environment and technology. Important capability which a few of defense companies are interested in.

The recognition is performed by a set of characteristics comparison of speakers from audio recordings or a predefined database. Based on the audio that is being transferred on the communication channel the recognition model will face another issue - noise.

For example - chatter in the background, wind, static interference and in certain radios there is a unique noise while talking which appears as background noises in the recordings.

Those noises interference along with the attempt to perform speaker recognition with high accuracy create a challenges to extract clear speech and

speaker unique characteristic.
At "Rafael" defense company there is an AI model whose its goal is to identify speakers using recordings of a BNET walkie-talkie. It was found that BNET produces a unique noise in the background when someone is talking over the radio and that the model does not perform well for these noisy recordings. The book offers a solution to this problem which includes learning the noise that a BNET radio causes to the speech itself and cleaning the noise from the recordings for a proper use of the recognition model. Performing a denoising process which includes the use of neural networks and sophisticated techniques to reduce and eliminate unwanted noise while maintaining the integrity of the speaker's voice and voice characteristics.
Noise reduction algorithms perform an analysis of spectral characteristics of the audio signal and selective attenuation of the noise components, thus help us to get a cleaner audio speech signal.

3. Background and Related Work

The project endeavors to apply cutting-edge deep learning techniques, specifically a novel Noise2Noise (N2N) approach, to significantly enhance the clarity of audio data in high-noise environments.
This solution defies traditional denoising techniques by training a network to map from one instance of noise to another, bypassing the need for noise-free audio samples. This distinctive feature makes it uniquely suited to real-world scenarios where clean training data may not be readily available.
Despite the inherent complexities of such an implementation, the potential advantages and efficacy of this approach are noteworthy. By enhancing the reliability and clarity of audio communications in high-noise environments, this solution opens new horizons for advancements in audio denoising, especially under circumstances where only noisy data are available.
The first phase is converting speech samples into spectrograms, then a visual representation of the audio signal in the frequency domain. By utilizing this technique, the goal is to demonstrate its effectiveness in reducing both synthetic noises and complex real-world noise distributions commonly encountered in urban environments.
The first part of this section is an introduction to denoising, and the second part is Mathematical background. later we define concepts that are used when solving the problem.

3.1. Introduction to denoising

The denoising process with neural networks typically involves a two-step approach. First, a large dataset of clean and noisy military radio recordings is used to train the neural network. The network learns to identify the underlying patterns and relationships between the noisy input and the corresponding clean output. This training process allows the network to effectively denoise new, unseen recordings.

Second, during the denoising phase, the trained neural network takes in a noisy military radio recording as input. It applies a series of complex mathematical operations (such as FFT or STFT) and transformations (Audio signal to Spectrogram), using the learned patterns, to separate the speech signal from the noise. The network's layers, consisting of multiple interconnected nodes to analyze and process the input spectrogram, exploiting the correlations between adjacent frequency and time components.

As the input progresses through the network layers, it gradually suppresses the noise components while preserving the essential and unique characteristics of the speech signal. This process relies on the network's ability to differ relevant features and distinguish between meaningful speech information and undesirable noise artifacts mainly with the help of the skip-connections (which will be explained later). The resulting output is a clean recording that closely resembles the original speech signal, enhancing the clarity and intelligibility of the military radio communication.

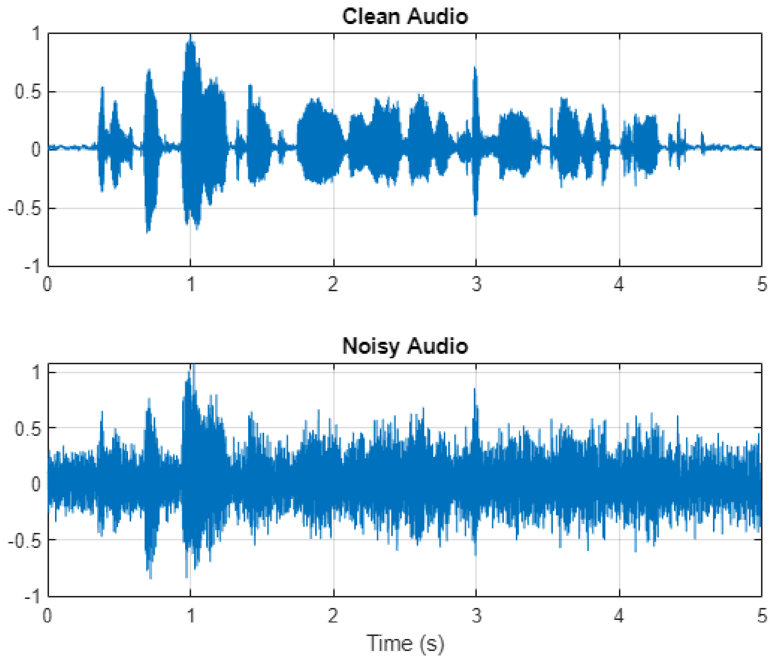


Figure 1: Example of comparison between noisy waveform and clean waveform. [4]

3.2. Mathematical background

3.2.1. Deep Neural Network (DNN)

The DNN is characterized by parameters: $argmin\ \theta$, loss function L, input x, output $f\theta(x)$, and target y. The DNN learns to denoise the input audio by solving an optimization problem, given by equation:

1 Equation

$$argmin_{\theta}\ E_{(x,y)}\{L(f_{\theta}(x), y)\}$$

The result of the loss function represents the difference between the network's output $f_{\theta}(x)$ and the target y. The main goal is to reduce it. $argmin$ returns the input that produces the smallest output of the function E. The objective is to find the pair of denoised input ($f_{\theta}(x)$- x) and target (y) with the smallest distance.

3.2.2. Noisy Audio Samples

A noisy audio sample is a clean audio sample with noise overlayed on it. Consider the clean audio as y, two noisy audio samples $x1$ and $x2$ are created by randomly sampling from independent noise distributions N and M, following conditions 1 and 2:

2 Equation

$$x_1 = y + n{\sim}N$$

3 Equation

$$x_2 = y + m{\sim}M$$

Condition 1: The noises added to the input and target are sampled from zero-mean distributions and are uncorrelated to the input.
Condition 2: The correlation between the noise in the input and in the target are close to zero.

3.2.3. Noise2Clean (N2C) Techniques

To realize how N2N works it's important to first understand the N2C (commonly used technique in denoising model in the training stage) - Traditional N2C techniques use noisy inputs and clean targets in the training stage. They have access to clean training audio targets and commonly employ a Mean Squared Error (MSE) loss function to solve the following optimization problem:

4 Equation

$$argmin_{\theta}\ L_{2,n2c} = argmin_{\theta}\ E_{(x_1,y)}\{(f_{\theta}(x_1) - y)^2\}$$

Here, the goal is to minimize the expected value of the squared difference between the network's output and the clean signal.

3.2.4. Noise2Noise (N2N) Approach

The N2N approach, does not have the option of using clean training audio for the targets. Instead, it employs noisy inputs and noisy targets during the training stage. The optimization problem in this case is given by:

5 Equation

$$L_{2,n2n} = E_{(x_1,x_2)}\{(f_{\theta}(x_1) - x_2)^2\} =$$

$$= E_{(x_1,x_{2,m\sim M})}\left\{\left((f_{\theta}(x_1) - (y + m)\right)^2\right\} =$$

$$= E_{(x_1,x_{2,m\sim M})}\{(f_{\theta}(x_1) - y)^2\} - E_{(x_1,x_{2,m\sim M})}\{2m(f_{\theta}(x_1) - y)\} + E_{(m\sim M)}\{m^2\} =$$

$$= L_{2,n2c} + Var(m) + (E_{(m\sim M)}\{m\})^2$$

The 1st term in the equation is N2C loss function, the 2nd and 3rd term are variance of m and estimated m^2, those 2 elements are getting close to zero.

The bigger the size of the noisy training dataset increases the closer the N2N MSE (Mean-Square Equation) value is getting to the N2C MSE value.

6 Equation

$$lim_{|TrainingDataSet|\to\infty}\ L_{2,n2c} = L_{2,n2n}$$

4. Expected achievements

The anticipated goals of this endeavor involve the creation of a sophisticated learning model capable of enhancing audio signals obtained from military radio communication, even with limited clean training data. The model's primary objective is to eliminate noise from the signal and enhance the speaker's sound quality, thereby optimizing the real-time performance of the Speech Recognition module developed by RAFAEL. The project's results will entail a comprehensive summary encompassing theoretical insights, algorithm implementation via code, and the visualization of achieved outcomes.

5. Research process

5.1. The project process

Part A:

After discussing the task with the RAFAEL team, we searched for possible models to perform denoising considering the existing limitations - lack of human resources assigned to the task, lack of time and a small database of clean recordings intend to train the model.

We first conducted initial research about neural networks for denoising audio signals and what components are required to this mission.

After that we compared between some models promised to do such that. We finally chose the N2N model and made sure that this method is possible to implement, meets the requirements and solves the problem effectively.

We did an in-depth reading of the article about N2N and wrote a book that describes the theory of the research and a proposal for its implementation for the purpose of solving the RAFAEL mission.

Part B:

In this section, the attention will be directed towards actualizing, executing, and assessing the model outlined in section A.

Initially, gathering of a database specifically designed to facilitate the learning process of the networks, utilizing the RAFAEL B-NET radio.

Subsequently, the network will be trained utilizing Python framework and neural network libraries, leveraging the collected dataset.

Ultimately, apply the model to effectively remove noise from the BNET radio signals and carry out a thorough evaluation of its performance.

5.1.1. N2N Advantages (against N2C)

Noise2Noise (N2N) approach has some advantages over the traditional Noise2Clean (N2C) approach, particularly in contexts where clean training data is hard to obtain or unavailable. These advantages include:

- **Requires relatively small amount of clean data:** The most significant advantage of N2N is that it requires relatively small amount of clean data (noise-free). In many real-world scenarios, obtaining clean data can be challenging, if not impossible. This is particularly true in situations with inherent noise, such as in military radio communication, medical imaging, or industrial quality inspection.

- **Potential for Robustness:** Because N2N is trained on noisy data, it might be more robust to noise in real-world usage compared to N2C models, which are typically trained on clean data and may not handle noise as well.

- **Cost Efficiency:** Collecting and preparing clean training data can be expensive and time-consuming. By using noisy data instead, N2N can potentially reduce the cost and effort involved in preparing the training dataset (very significant feature when working small developments team).

However, it is important to note that N2N has its own set of challenges and limitations. For instance, the model requires pairs of noisy observations of the same underlying clean signal. Moreover, the added noise must satisfy certain conditions (being zero-mean and uncorrelated with the signal) for the approach to work effectively. Finally, the performance of N2N may depend on the nature of the noise and its relationship to the underlying clean signal.

5.1.2. Potential challenges

- Dataset creation – The algorithm requires training dataset, the data will be collected from RAFAEL.
  Collecting sufficient amount of data (recordings) and reliable one.
  Due to security aspects, we can't use data that is already exist because of the risk in leaking high risk information.
  There for we will need to create our own data - something that requires a lot of time and effort or alternatively use existing one but then we need it to be close as possible to the real use-case as the BNET communication system that RAFAEL has.

- Coding environment – Training the algorithm may require a work environment with powerful enough hardware.
  Therefore, it is necessary to find both a convenient development environment and a sufficiently strong environment for training and verification.

- Noise characteristic – The model was originally trained on Gaussian-like noises and in communication through BNET the noise will be different than simply a white noise.
  We hope to face those challenges and solve them in the best way possible.

5.2. Product - N2N algorithm

5.2.1 Datasets and Data Generation for N2N training and evaluation

This section presents creation of an optional dataset, which can be used in further experiments.

Definitions:

1. **Pydub** - A Python library used for manipulating audio files by truncating, repeating, overlapping, or applying various transformations.
2. **SNR** - a measure of the relative strength of a desired signal compared to the background noise present in a given system or environment.
3. **PESQ** - An objective measurement method used to assess the quality of speech signals by comparing the original and degraded versions based on human auditory perception.

4. **UrbanSound8K** – A dataset that contains 8732 labeled sound excerpts of urban sounds from 10 different classes.

In the context of training the DC-Unet model for denoising audio signals, we will follow a specific process utilizing the Noise2Noise algorithm. To generate the necessary training samples, we will record audio samples using radio communication, specifically through the Walkie-Talkie. These samples will be used to create a dataset for training and evaluating the denoising model.

To ensure a comprehensive and diverse dataset, we will combine different sources of audio. One of the sources is a clean and noisy parallel speech database, specifically designed for training and testing speech enhancement methods. This dataset contains recordings from twenty-eight speakers and operates at a sampling rate of 48kHz. Additionally, we will incorporate the UrbanSound8K dataset, which consists of labeled sound excerpts from various urban sounds.

Using a Python library called Pydub, we will manipulate the audio files by applying transformations such as truncation, repetition, overlapping, and more. The Signal-to-Noise Ratio (SNR), a measure of signal strength relative to background noise, will guide our generation process. We aim to achieve an average SNR of 5dB in the training files. However, if the resulting files obtain high Perceptual Evaluation of Speech Quality (PESQ) scores, indicating high quality, they will not be suitable for denoising verification. Therefore, adjustments will be made to create a blind denoising scenario where the original SNR of the clean audio and the noise remain random values between 0 and 10.

To create the training audio files, we will overlay random noise samples onto the clean audio files, ensuring that the noise covers the entire speech segment. Target training audio files will be generated using clean audio and random noise samples from different categories than those used for the input files. This approach will result in training sets with varied SNR values, providing a challenge for denoising techniques while maintaining intelligible speech for human evaluation.

The testing dataset will follow a similar generation process as the training datasets. The noisy audio files will serve as the testing inputs, while the underlying clean audio files will act as the testing references. This setup ensures that the denoising techniques are evaluated in realistic scenarios, closely resembling real-world applications.

By creating benchmark datasets that include noise in both the input and target, we can comprehensively evaluate denoising techniques. The utilization of the UrbanSound8K dataset, along with various noise generation strategies, offers a diverse range of challenges for denoising algorithms. This comparative analysis will inform further advancements in denoising techniques, leading to improved noise reduction in various real-world applications.

5.2.2. Network architecture

The Noise2Noise approach, as demonstrated in the provided study, employs the Deep Complex U-Net (DCUnet-20) [2] architecture – a complex-valued masking framework that extends upon the U-Net architecture. This network consists of 20 layers and has achieved state-of-the-art results on the VOICEBANK+DEMAND speech enhancement benchmark. The superior speech enhancement metrics result from the DCUnet-20's ability to more precisely understand and recreate both phase and magnitude information from spectrograms.

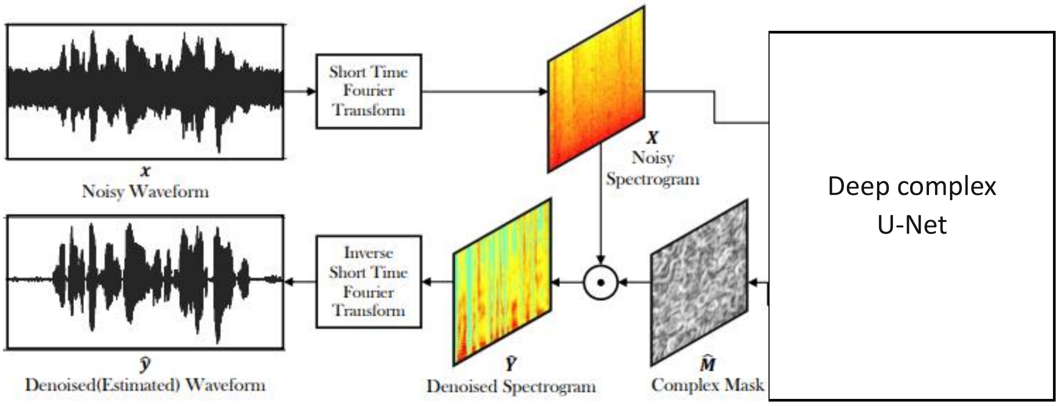


Figure 2 Image represents a use of U-Net as the core of the denoising process. [1]

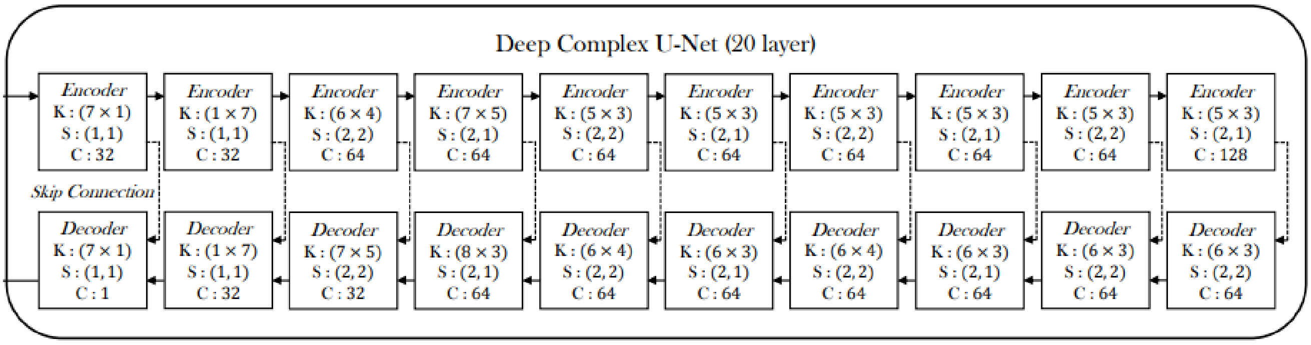

Deep Complex U-Net (20 layer)
Encoder K : (7 × 1) S : (1, 1) C : 32
Encoder K : (1 × 7) S : (1, 1) C : 32
Encoder K : (6 × 4) S : (2, 2) C : 64
Encoder K : (7 × 5) S : (2, 1) C : 64
Encoder K : (5 × 3) S : (2, 2) C : 64
Encoder K : (5 × 3) S : (2, 1) C : 64
Encoder K : (5 × 3) S : (2, 2) C : 64
Encoder K : (5 × 3) S : (2, 1) C : 64
Encoder K : (5 × 3) S : (2, 2) C : 64
Encoder K : (5 × 3) S : (2, 1) C : 128
Skip Connection
Decoder K : (7 × 1) S : (1, 1) C : 1
Decoder K : (1 × 7) S : (1, 1) C : 32
Decoder K : (7 × 5) S : (2, 2) C : 32
Decoder K : (8 × 3) S : (2, 1) C : 64
Decoder K : (6 × 4) S : (2, 2) C : 64
Decoder K : (6 × 3) S : (2, 1) C : 64
Decoder K : (6 × 4) S : (2, 2) C : 64
Decoder K : (6 × 3) S : (2, 1) C : 64
Decoder K : (6 × 3) S : (2, 2) C : 64
Decoder K : (6 × 3) S : (2, 2) C : 64

Figure 3 Image represents U-Net architecture. [1]

The process begins with the conversion from the time-domain waveform into the time-frequency domain using STFT (=Short Time Fourier Transform).
This transform outputs a complex matrix spectrogram, which can be factorized into a real-valued phase component and a complex-valued magnitude component. The STFT is computed with FFT (=Fast Fourier Transform) size of

3072, a number of bins equating to 1536, and a hop size of 16ms. The spectrogram is then normalized to ensure it follows by Parseval's energy-conservation property, meaning that the energy in the spectrogram equals the energy in the original time-domain waveform.

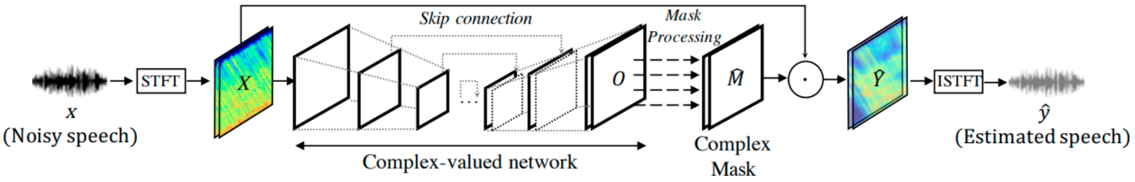


Figure 4 Illustration of speech enhancement framework with DCUnet. [2]

Traditional real-valued neural architectures such as U-Net only extract information from the magnitude spectrogram, discarding useful data from the phase spectrogram. This is due to the inability of conventional real-valued convolutional neural layers to process the complex-valued phase information. DCUnet overcomes this limitation by adopting a complex-valued convolutional neural network capable of processing both phase and magnitude spectrograms. This network architecture results in better precision during phase estimation and reconstruction of the enhanced audio.

The DCUnet-20 can be best described as a complex-valued convolutional autoencoder utilizing stride (residual) connections. It incorporates complex convolution layers, complex batch normalization, complex weight initialization, and activation functions.
The strided complex convolutional layers help to prevent spatial information loss during the downsampling process, while the strided complex deconvolutional layers restore the size of the input during upsampling. The encoding and decoding stages of the network consist of a complex convolution with specified kernel sizes, stride sizes, and output channels as described by Firgure 3, followed by complex batch normalization, and finally, a leaky CReLU (LeCReLU) activation function.
This activation function is applied to both the real and imaginary parts of the neuron.

7 Equation

$$LeCReLU = LeCReLU\big(R(z)\big) + iLeCReLU\big(T(z)\big)$$

The loss function used in this architecture is the novel weighted SDR loss function. In this function, the noisy speech at a given time step, the target source, and the estimated source are all considered.

x – noisy speech
T – time step
y – target source
$\hat{y}$ – estimated source
α – energy ratio between target source and noise and defined as follows:

$$\alpha = \frac{\|y^2\|}{\|y^2\| + \|x - y\|^2}$$

The loss function is:

8 Equation

$$loss_{wSDR}(x, y, \hat{y}) = -\alpha \frac{< y, \hat{y} >}{\|y\|\|\hat{y}\|} - (1 - \alpha) \frac{< x - y, x - \hat{y} >}{\|x - y\|\|x - \hat{y}\|}$$

A speech spectrogram is computed by multiplying the estimated mask with the input spectrogram. The estimated mask is then calculated using a novel polar coordinate-wise complex-valued ratio mask. Finally, an ISTFT (=Inverse Short Time Fourier Transform) is applied to convert the estimated time-frequency domain spectrogram back into its time-domain waveform representation.

The distinctive feature of this network, distinguishing it from others, lies in the utilization of skip-connections and the complex nature of both input data and network parameters, such as convolution filters. Skip-connections involve the concatenation of layers from the Encoder with their corresponding "parallel" layers from the Decoder. This strategy preserves spatial information from the input signal and mitigates information loss during the encoding process.

In the speech enhancement context, the DCU-Net architecture comprises two components: the encoder and the decoder. The encoder focuses on extracting high-level features from the input signal and compressing them into a lower-dimensional representation. On the other hand, the decoder's role is to reconstruct the output signal using the compressed representation generated by the encoder. The skip-connections between the encoder and decoder facilitate the combination of low-level features from the encoder with high-level features from the decoder, thereby enhancing the network's performance by enabling the learning of more intricate representations of the input signal.

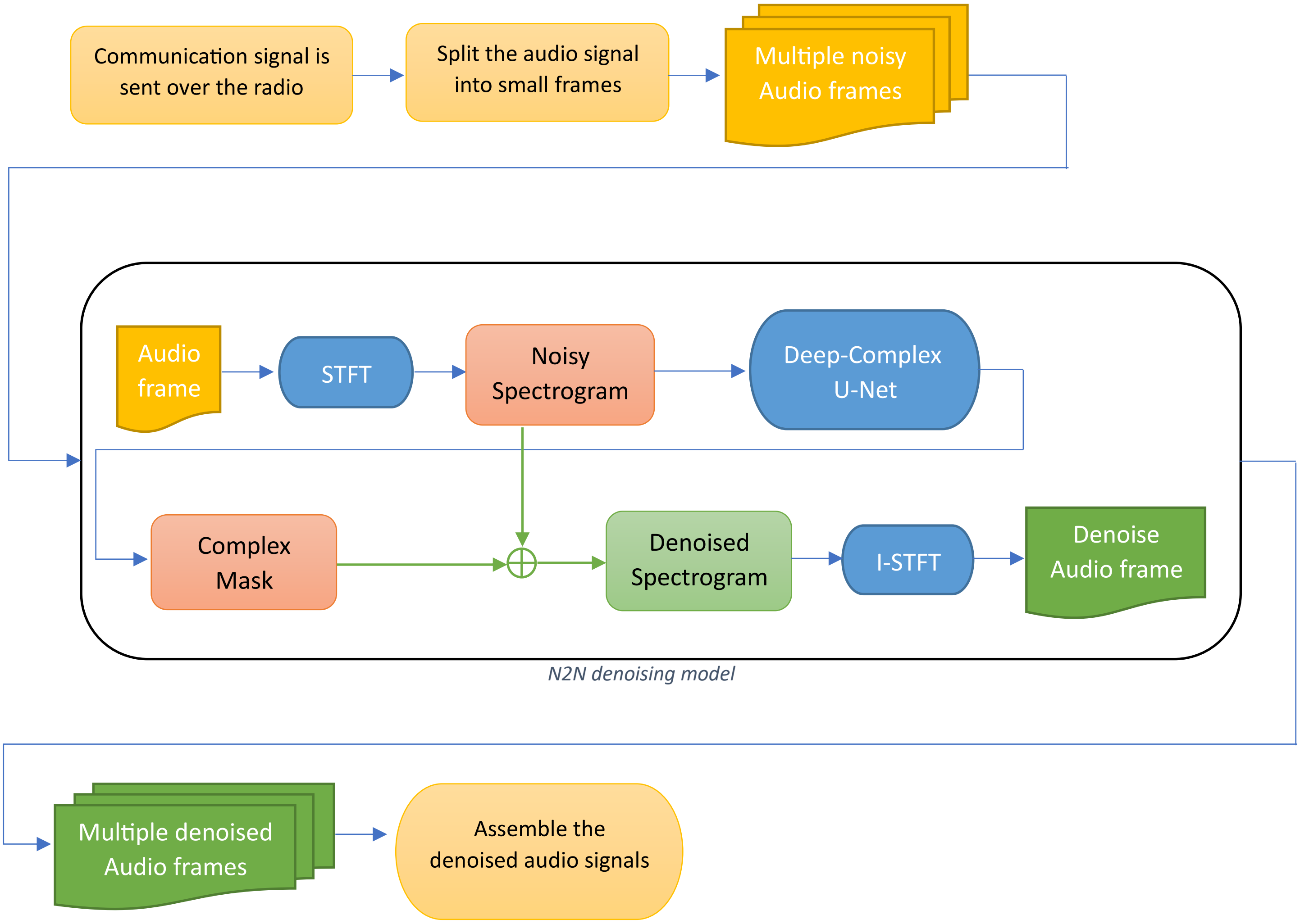
5.2.3. Training process

The model was trained using both noisy inputs and noisy targets. This means that it learned to remove noise from inputs that already had noise in them, without relying on any clean data.

To evaluate the performance of these models, several metrics were used. These metrics measure different aspects of the output quality. The first metric is SNR, which stands for Signal-to-Noise Ratio. It quantifies the ratio of the desired speech signal to the unwanted noise. The second metric is SSNR, or Segmented Signal-to-Noise Ratio, which is a variation of SNR that takes into account specific segments of the speech signal. The third and fourth metrics are wide-band and narrow-band PESQ scores. PESQ stands for Perceptual Evaluation of Speech Quality and measures how closely the generated speech matches the original clean speech in terms of perceived quality. The wide-band and narrow-band refer to different frequency ranges analyzed by PESQ. Finally, the fifth metric is STOI, which stands for Short Term Objective Intelligibility. STOI evaluates the intelligibility or clarity of the generated speech.

These metrics provide a comprehensive assessment of the denoising models' performance. They not only measure their ability to reduce unwanted noise but also provide an objective measure of the quality of the speech produced by these models.

5.2.4. Flow diagram


Communication signal is sent over the radio
Split the audio signal into small frames
Multiple noisy Audio frames
Audio frame
STFT
Noisy Spectrogram
Deep-Complex U-Net
Complex Mask
Denoised Spectrogram
I-STFT
Denoise Audio frame
N2N denoising model
Multiple denoised Audio frames
Assemble the denoised audio signals

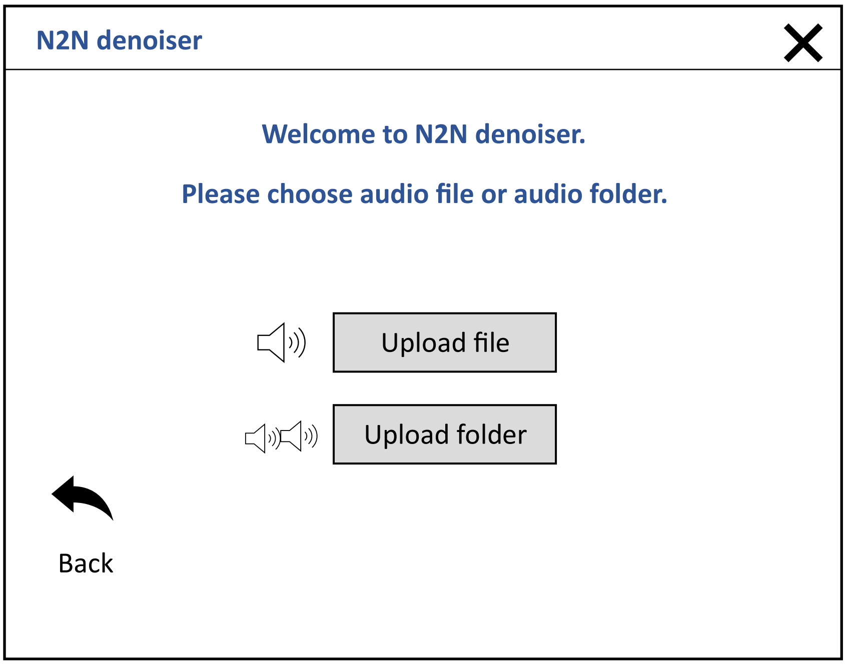
5.2.5. Gui

In part B of the project, a simple user interface will be created for removing noise from audio files for the convenience of the users.

The interface will be written using Python and "tkinter" library and will include the following functions for the user:

1. Uploading a file from file explorer
2. Uploading a file directory from file explorer
3. Removing noise from a file or file directory.
4. Listen to the denoised audio – the output.

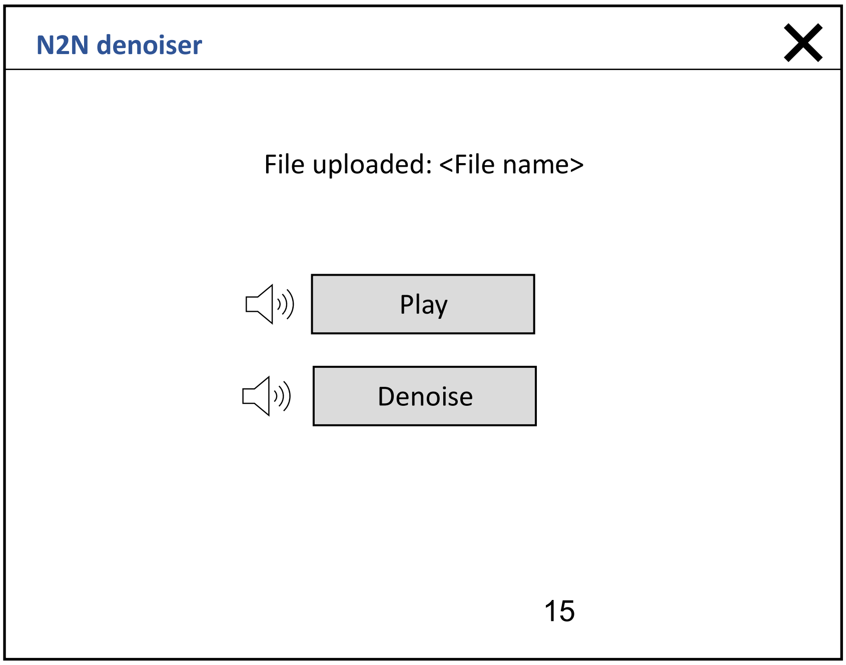
Example:



N2N denoiser

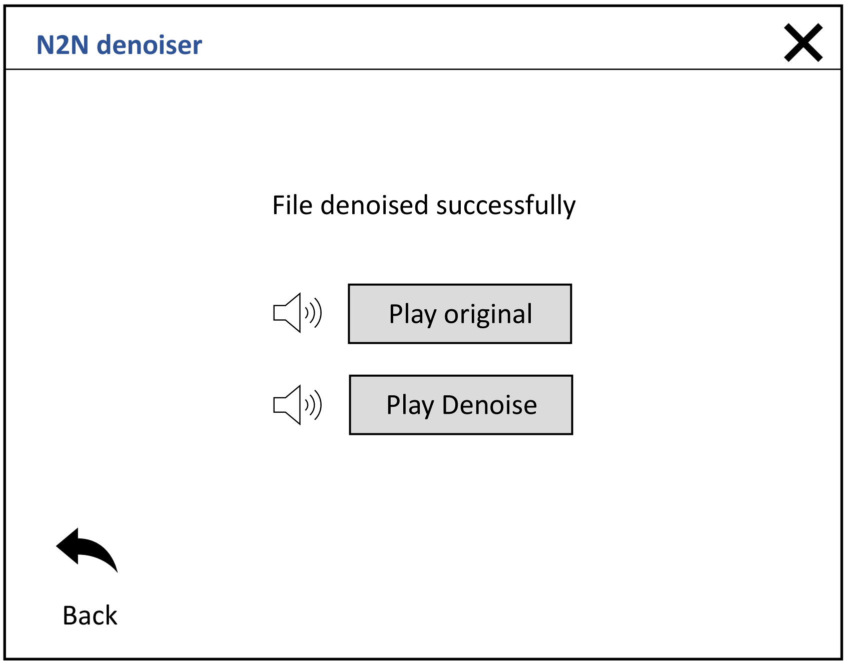
Welcome to N2N denoiser.

Please choose audio file or audio folder.



N2N denoiser

File uploaded: <File name>



6. Evaluation/Verification Plan

Iterative Product Evaluation and Verification Plan for N2N:

Define general metrics to measure success:

- Accuracy of noise removal: Measure the model's ability to remove noise from audio signals accurately.
- Signal-to-Noise Ratio (SNR): Quantify the improvement in SNR after denoising.
- Speech recognition accuracy: Measure the accuracy of speech recognition on the denoised audio signals (by RAFAEL).

Iteration 1:

- Training and Validation:
  - Train the initial Noise2Noise deep learning model using a dataset of noisy audio inputs and corresponding noisy targets.
  - Validate the model's performance on a separate validation dataset.
- Evaluation:
  - Measure the accuracy of noise removal by comparing the denoised audio with the original noisy audio using metrics like Mean Squared Error (MSE) or Signal-to-Noise Ratio (SNR).
  - Evaluate the speech recognition accuracy on the denoised audio samples using a pre-trained speech recognition model (using PESQ).
- Verification:
  - Visually assess the quality and level of noise removal by comparing the denoised audio with the original noisy audio.

- Verify the impact of denoising on speech recognition accuracy by comparing it with the accuracy on the noisy audio.

Iteration 2 and Beyond:

- Training and Validation:
  - Further train the Noise2Noise model using additional noisy audio inputs and corresponding noisy targets.
  - Perform cross-validation to evaluate the model's generalization ability.
- Evaluation:
  - Measure the accuracy of noise removal using appropriate metrics and compare with the previous iteration.
  - Assess the impact on speech recognition accuracy and compare with the previous iteration.
- Verification:
  - Continuously visually assess the quality of denoised audio.
  - Verify the impact on speech recognition accuracy and compare with the previous iteration.

Success Criteria and Iteration Termination:

- success criteria: noise removal accuracy > 90%, speech recognition accuracy > 90%, SNR improvement > 10dB).
- Terminate the iterative development process when the success criteria for all metrics are met.
- If the success criteria are not met, continue iterating by fine-tuning the model, adjusting hyperparameters, or augmenting the training data.

Note: The iterative development process for the Noise2Noise deep learning denoising model follows an agile and data-driven approach. Regular evaluations, verifications, and adjustments are essential to iteratively enhance the model's performance and meet the defined success criteria. The specific implementation and techniques used may vary based on the project's context and available resources.

7. Future plans

We aim to implement these principles in practice and develop a system that can denoise military radio communications using a Deep Neural Network and the Noise2Noise approach. We will use a loss function that seeks to maximize the similarity between the input and the target, such as Signal-to-Distortion Ratio (SDR) or Signal-to-Noise Ratio (SNR)-based losses and train the network on noisy audio samples as both the input and the target.

8. References

[1] Madhav Mahesh Kashyap, Anuj Tambwekar, Krishnamoorthy Manohara, S Natarajan, “Speech Denoising without Clean Training Data: a Noise2Noise Approach", Department of Computer Science and Engineering, PES University, India, Sep 2021.

[2] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, “Phase-aware speech enhancement with deep complex u-net,” in International Conference on Learning Representations, 2018.

[3] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, “Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs,” in 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221), vol. 2, 2001, pp. 749–752 vol.2.

[4] https://github.com/pheepa/DCUnet