



Speech Denoising without Clean Training Data: a Noise2Noise Approach

Madhav Mahesh Kashyap*, Anuj Tambekar*, Krishnamoorthy Manohara, S Natarajan

Department of Computer Science and Engineering, PES University, India

madhavkashyap99@gmail.com, anujtam@gmail.com, krishnam3103@gmail.com, natarajan@pes.edu

Abstract

This paper tackles the problem of the heavy dependence of clean speech data required by deep learning based audio-denoising methods by showing that it is possible to train deep speech denoising networks using only noisy speech samples. Conventional wisdom dictates that in order to achieve good speech denoising performance, there is a requirement for a large quantity of both noisy speech samples and perfectly clean speech samples, resulting in a need for expensive audio recording equipment and extremely controlled soundproof recording studios. These requirements pose significant challenges in data collection, especially in economically disadvantaged regions and for low resource languages. This work shows that speech denoising deep neural networks can be successfully trained utilizing only noisy training audio. Furthermore it is revealed that such training regimes achieve superior denoising performance over conventional training regimes utilizing clean training audio targets, in cases involving complex noise distributions and low Signal-to-Noise ratios (high noise environments). This is demonstrated through experiments studying the efficacy of our proposed approach over both real-world noises and synthetic noises using the 20 layered Deep Complex U-Net architecture.

Index Terms: Speech Denoising, Speech Enhancement, Noise Reduction, Deep Learning, Data Collection, Noise2Noise

1. Introduction

Deep Learning [1] has revolutionized the domains of Computer Vision and Speech, Language and Audio Processing. The recent surge in popularity of deep learning has resulted in a multitude of new data-driven techniques to tackle challenges in the domains of speech and audio, such as removing noise from speech in order to enhance speech intelligibility. Its primary strength comes from its ability to leverage massive amounts of data to find relationships and patterns, and its ability to learn varying representations of the data. However, this strength is also one of the alleged pitfalls of deep learning, in that it is often a sub-optimal solution when dealing with insufficient or noisy and corrupted data. In the audio domain, this entails the collection of a large amount of perfectly clean recordings, a proposition which is often challenging in areas that are home to low-resource languages due to the large upfront costs of creating facilities that have the necessary soundproofing and equipment required for such a task.

However, the pioneering work of Lehtinen et al [2], disproved one of these dependencies - it is possible to train convolutional neural networks to denoise images, without ever being shown clean images. This paper is a natural extension of Noise2Noise in the audio domain, by demonstrating that it is possible to train deep speech denoising networks, without ever having access to any kind of clean speech. Additionally, our findings indicate that for complex noise distributions at

low Signal-to-Noise (SNR) ratios, using noisy training data can yield better results. This can incentivize the collection of audio data, even when the circumstances are not ideal to allow it to be perfectly clean. We believe that this could significantly advance the prospects of speech denoising technologies for various low-resource languages, due to the decreased costs and barriers in data collection. The source code for our Noise2Noise speech denoiser is available on GitHub¹ under the MIT License.

2. Background and Theory

2.1. Motivation and Related Work

The motivation for this work stems from [2], where the authors show that it is possible to denoise images using only noisy images as a reference, provided two key conditions hold.

- **Condition 2.1** The noises added to the input and target are sampled from zero-mean distributions and are uncorrelated to the input.
- **Condition 2.2** The correlation between the noise in the input and in the target is close to zero.

The first condition ensures that the median or mean of the target distribution stays the same, despite the presence of noise; while the second ensures that the network does not learn a mapping from one noise type to the other, but rather learns a robust generalization aimed to remove the noise.

In this paper, the Noise2Noise technique is applied in the audio space, by converting speech samples into spectrograms, and its efficacy is demonstrated on both synthetic noises and complex real-world world noise distributions that one may encounter in urban environments. Recent work showcases that self-supervised approaches using a combination of noisy targets alongside clean targets can improve speech denoising performance [3, 4]. The experiments performed in this work indicate that even in fully supervised training regimes, the presence of clean speech is not a requirement when dealing with deeper networks and sufficient samples. This allows deep networks to be trained in the removal of complex noises without any requirement or dependence on speech data devoid of noise.

2.2. Theoretical Background

Consider a Deep Neural Network (DNN) with parameters θ , loss function L , input x , output $f_{\theta}(x)$, and target y . The DNN learns to denoise the input audio by solving the optimization problem shown in Eqn 1:

$$\operatorname{argmin}_{\theta} \mathbb{E}_{(x,y)} \{L(f_{\theta}(x), y)\} \quad (1)$$

A noisy audio sample is a clean audio sample with noise overlaid on it. Consider the clean audio y . 2 noisy audio samples

¹<https://github.com/madhavmk/Noise2Noise-audio-denoising-without-clean-training-data>

* equal contribution

x_1 and x_2 are created by randomly sampling from independent noise distributions N and M , following conditions 2.1 and 2.2

$$x_1 = y + n \sim N \text{ and } x_2 = y + m \sim M \quad (2)$$

In this work, techniques using noisy inputs and clean targets in the training stage are described as Noise2Clean (N2C) techniques. Traditional Noise2Clean DNN approaches [5, 6, 7, 8] have access to clean training audio targets, and commonly employ a L_2 loss function to solve the following optimization:

$$\operatorname{argmin}_{\theta} L_{2,n2c} = \operatorname{argmin}_{\theta} \mathbb{E}_{(x_1,y)} \{(f_{\theta}(x_1) - y)^2\} \quad (3)$$

Our Noise2Noise (N2N) approach does not have the luxury of using clean training audio for the targets. Instead, it employs noisy inputs and noisy targets during the training stage.

$$L_{2,n2n} = \mathbb{E}_{(x_1,x_2)} \{(f_{\theta}(x_1) - x_2)^2\} \quad (4)$$

$$= \mathbb{E}_{(x_1,x_2,m \sim M)} \{(f_{\theta}(x_1) - (y + m))^2\} \quad (5)$$

$$= \mathbb{E}_{(x_1,x_2,m \sim M)} \{(f_{\theta}(x_1) - y)^2\} - \mathbb{E}_{(x_1,x_2,m \sim M)} \{2m(f_{\theta}(x_1) - y)\} + \mathbb{E}_{m \sim M} \{m^2\} \quad (6)$$

$$= L_{2,n2c} + \operatorname{Var}(m) + \mathbb{E}_{m \sim M} \{m\}^2 \quad (7)$$

$\mathbb{E}_{m \sim M} \{m\} = 0$ due to Condition 2.1. This causes the second term in Eqn 6 and the third term in Eqn 7 to equal 0. Mathematically, the expectation of the m^2 is equal to the variance of m plus the square of the expectation of m . This fact is used to expand the third term in Eqn 6. The variance of the sample distribution $\operatorname{Var}(m)$ is equal to the variance of the population divided by the sampling size. Hence as the size of the noisy training dataset increases, the Noise2Noise $L_{2,n2n}$ loss value tends to equal the Noise2Clean $L_{2,n2c}$ loss value.

$$\lim_{|TrainingDataSet| \rightarrow \infty} L_{2,n2n} = L_{2,n2c} \quad (8)$$

A similar derivation proves we get equivalent results if we instead employ a L_1 loss function for the optimization.

$$\operatorname{argmin}_{\theta} L_{1,n2c} = \operatorname{argmin}_{\theta} \mathbb{E}_{(x_1,x_2)} \{|f_{\theta}(x_1) - x_2|\} \quad (9)$$

$$\lim_{|TrainingDataSet| \rightarrow \infty} L_{1,n2n} = L_{1,n2c} \quad (10)$$

We can also extend the other conclusion of [2] from the pixel domain to the time domain. If the same audio clip had varying uncorrelated noises and was averaged, the average would result in the true audio. Hence, any loss function that aims to maximize the similarity between the input and the target, such as SDR or SNR-based losses, is also appropriate for Noise2Noise based training. This leads us to the following theorem :

Theorem 2.3 *Deep neural networks can be trained to denoise audio by employing a technique that uses noisy audio samples as both the input as well as the target to the network, subject to the noise distributions being zero mean, independent of the true signal and uncorrelated.*

In the following sections, we show the results of practically applying Theorem 2.3 on real-world speech samples, and on synthetic and real-world noise distributions.

3. Experimental Setup

3.1. Datasets and Data Generation

Due to the lack of a pre-existing benchmark dataset containing noise in both the input and target, a collection of datasets was generated in order to compare the performance of Noise2Clean training with respect to Noise2Noise training. The clean speech files for these datasets came from the 28 speaker version of [9] - 26 speakers are used for training, and the other 2 unseen speakers are used for evaluation. All 10 noise categories of the UrbanSound8K dataset [10] were used. This dataset was chosen for its collection of samples from numerous real-world noise categories.

Separate training and testing datasets are created for each UrbanSound8K noise category N . For each noise type N , the input training audio file is generated by overlaying a random noise sample from N with repetition on top of a clean audio file. Computing the number of repetitions and then scaling the noise to reach the target average SNR of 5dB resulted in files with Perceptual Evaluation of Speech Quality (PESQ) scores that were already too high to be good candidates to verify the efficacy of our denoising approach. Instead, the volume of the noise is adjusted such that the original SNR of the clean audio and the noise is a random number in the range 0 to 10 (inclusive of both), resulting in a blind denoising scenario. The noise is then overlapped over the clean audio using PyDub [11], which truncates or repeats the noise such that it covers the entire speech segment. Next, a corresponding target training audio file is generated using the same underlying clean audio file, and a random noise sample from a category that is not N . Due to this method, the UrbanSound8K training sets do not have an average SNR of 5dB; but nevertheless possess many highly noisy samples where the speech can be discerned by the human ear, while still posing a significant challenge for denoising techniques (see the Baseline metrics in Figure 2).

The Mixed category dataset was created by picking a random noise category for the input file, while picking another random noise category for the target file, ensuring both don't use the same noise category N . The White noise category dataset was generated by using random additive white gaussian noise with SNR scaled randomly in the range 0 to 10, on both the input and target training files.

The testing dataset was generated in the same fashion. The testing input is the noisy audio file, whereas the testing reference is the underlying clean audio file.

3.2. Network Architecture

We demonstrate the effectiveness of this Noise2Noise approach using the 20 layered Deep Complex U-Net [12] (DCU-Net-20) architecture. This complex-valued masking framework is an extension upon the popular U-Net [13] architecture and has achieved state of the art results on the VOICE-BANK+DEMAND [14, 15, 16] speech enhancement benchmark. Superior speech enhancement metrics are achieved as a result of its ability to more precisely understand and recreate both phase and magnitude information from spectrograms.

First, the time domain waveform is converted into the time-frequency domain using the Short Time Fourier Transform (STFT). This transform outputs a linearly scaled, complex matrix spectrogram, factorizable into a real-valued phase component and a complex-valued magnitude component. The STFT is computed with a FFT size of 3072, number of bins equaling 1536, and hop size of 16ms. Normalization is then carried out

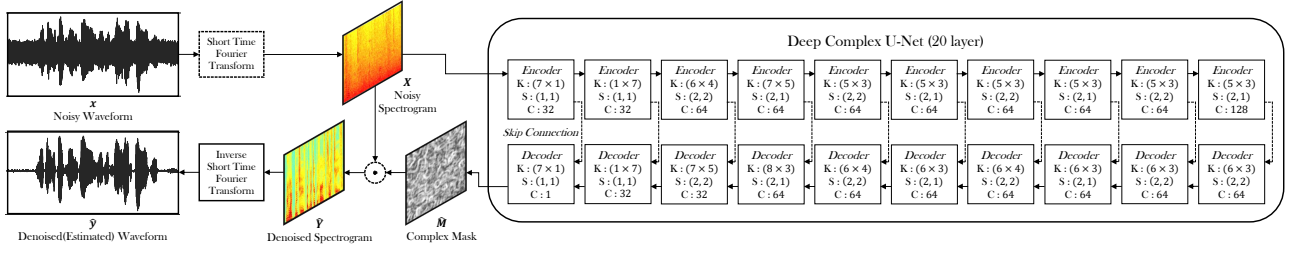


Figure 1: Speech denoising framework using the DCUnet-20 model. K denotes kernel size, S denotes stride and C denotes the output channel size

to ensure compliance with Parseval’s energy-conservation property [17], meaning that the energy in the spectrogram equals the energy in the original time domain waveform.

Real-valued neural architectures such as U-Net extract information from only the magnitude spectrogram, discarding useful data from the phase spectrogram. This is because the complex-valued phase information cannot be processed by conventional real-valued convolutional neural layers. DCUnet overcomes this limitation by instead opting for a complex-valued convolutional neural network capable of processing both phase and magnitude spectrograms. This results in better precision during phase estimation and reconstruction of the enhanced audio. Figure 1 describes the 20 layered DCUnet framework employed in our work. It is best described as a complex-valued convolutional autoencoder utilizing stride (residual) connections. Complex convolutional layers, complex batch normalization, complex weight initialization, and $\mathbb{C}ReLU$ are applied as described in [18].

Strided complex convolutional layers prevent spatial information loss when downsampling. Strided complex deconvolutional layers restore the size of input when upsampling. The Encoding and Decoding stages consist of a complex convolution with kernel sizes, stride sizes, and output channels as described by Figure 1, followed by complex batch normalization, and finally a leaky $\mathbb{C}ReLU$ ($Le\mathbb{C}ReLU$) activation function. The $Le\mathbb{C}ReLU$ is a modified $\mathbb{C}ReLU$ where the leaky $ReLU$ [19] ($LeReLU$) activation function is applied on both the real and imaginary parts of the neuron. Where $z \in \mathbb{C}$:

$$Le\mathbb{C}ReLU = LeReLU(\Re(z)) + i LeReLU(\Im(z))$$

We apply the novel weighted SDR loss function ($loss_{wSDR}$) introduced by [12]. Let x denote noisy speech with T time step, y denote target source and \hat{y} denote estimated source. If α is the energy ratio between target source and noise, then $loss_{wSDR}(x, y, \hat{y})$ is defined as:

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

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

The estimated speech spectrogram $\hat{Y}_{t,f}$ is computed by multiplying the estimated mask $\hat{M}_{t,f}$ with the input spectrogram $X_{t,f}$. The novel polar coordinate-wise complex-valued ratio mask is detailed in [12], and is estimated as follows.

$$\hat{Y}_{t,f} = \hat{M}_{t,f} \cdot X_{t,f} = |\hat{M}_{t,f}| \cdot |X_{t,f}| \cdot e^{i(\theta_{\hat{M}_{t,f}} + \theta_{X_{t,f}})}$$

$$\hat{M}_{t,f} = \hat{M}_{t,f}^{magnitude} \cdot \hat{M}_{t,f}^{phase}$$

$$\text{where } \hat{M}_{t,f}^{magnitude} = \tanh(O_{t,f}) \text{ and } \hat{M}_{t,f}^{phase} = \frac{O_{t,f}}{|O_{t,f}|}$$

An Inverse Short Time Fourier Transform (ISTFT) is then applied to convert the estimated time-frequency domain enhanced spectrogram into its time domain waveform representation.

3.3. Training and Evaluation Methodology

A DCUnet-20 model is trained using noisy training inputs and clean training targets - this model is denoted by N2C (Noise2Clean). Another identical DCUnet-20 model is trained using noisy training inputs and noisy training targets (as described in the dataset generation section above) - this model is denoted by N2N (Noise2Noise). As such, the N2N denoiser is never exposed to any clean data during training. For each N , the following five metrics are computed - SNR, Segmented SNR (SSNR), wide-band and narrow-band PESQ scores [20], and Short Term Objective Intelligibility (STOI) [21]. These scores give a reflection of not just the ability to remove signal disturbance but also provide an objective measure of the quality of speech produced. All the models were trained with a Nvidia K80 GPU, with a batch size of 2 till convergence (roughly 4 epochs). The implementations of DCUnet-20², PESQ³ and STOI⁴ were based on open source repositories.

4. Results

The results are tabulated in Table 1. The mean and standard deviation of the SNR, SSNR, narrow-band PESQ score (PESQ-NB), wide-band PESQ score (PESQ-WB), and STOI on the test set are reported. Each row corresponds to a noise category with the Baseline numbers indicating the values before denoising, N2C indicating the performance of the traditional Noise2Clean approach, and N2N indicating the performance of our proposed Noise2Noise approach. A green highlighted cell denotes the better performer (higher mean) among N2C and N2N for a given noise category and metric. The violin plot in Figure 2 compares the PESQ-NB metric density distribution shifts pre and post-denoising using the N2C and N2N methods.

N2C performs marginally better than N2N on all metrics for White noise, and on the SSNR metric for Engine Idling. However these performance differences are marginal, likely due to limited phase information in case of White noise. We hypothesize that N2C performs better/on-par with N2N in case of stationary noises like Engine Idling. In every other category (eg. Siren) and metric, N2N performs better than N2C, due to the ability of the network to generalize better [3] and avoid getting

²<https://github.com/pheepa/DCUnet>

³<https://github.com/ludlows/python-pesq>

⁴<https://github.com/mpariente/pystoi>

Table 1: Denoising performance results for N2C and N2N based DCUnet-20 networks. A number next to a category denotes its class number in the UrbanSound8K dataset

| Noise Category Name | Metric | SNR | SSNR | PESQ-NB | PESQ-WB | STOI |
|----------------------|------------|--------------------|--------------------|-------------------|-------------------|-------------------|
| White | Baseline | 4.589 \pm 2.903 | -4.572 \pm 2.352 | 1.526 \pm 0.173 | 1.095 \pm 0.048 | 0.557 \pm 0.173 |
| | N2C | 17.323 \pm 3.488 | 4.047 \pm 4.738 | 2.655 \pm 0.428 | 1.891 \pm 0.359 | 0.655 \pm 0.179 |
| | N2N (ours) | 16.937 \pm 3.973 | 3.752 \pm 4.918 | 2.597 \pm 0.462 | 1.840 \pm 0.375 | 0.650 \pm 0.180 |
| Mixed | Baseline | 0.629 \pm 3.849 | -4.775 \pm 4.040 | 1.800 \pm 0.460 | 1.251 \pm 0.318 | 0.554 \pm 0.201 |
| | N2C | 3.645 \pm 3.676 | -1.109 \pm 3.315 | 1.795 \pm 0.285 | 1.281 \pm 0.147 | 0.533 \pm 0.183 |
| | N2N (ours) | 3.948 \pm 5.285 | -0.711 \pm 4.049 | 2.114 \pm 0.459 | 1.455 \pm 0.292 | 0.593 \pm 0.206 |
| Air Conditioning (0) | Baseline | 1.172 \pm 3.560 | -5.351 \pm 2.690 | 1.921 \pm 0.450 | 1.212 \pm 0.207 | 0.593 \pm 0.187 |
| | N2C | 4.174 \pm 3.608 | -1.433 \pm 3.124 | 1.980 \pm 0.232 | 1.386 \pm 0.165 | 0.578 \pm 0.180 |
| | N2N (ours) | 4.656 \pm 5.612 | -0.800 \pm 3.687 | 2.440 \pm 0.386 | 1.658 \pm 0.298 | 0.641 \pm 0.178 |
| Car Horn (1) | Baseline | 1.085 \pm 3.868 | -4.138 \pm 5.103 | 1.839 \pm 0.536 | 1.336 \pm 0.464 | 0.558 \pm 0.196 |
| | N2C | 4.143 \pm 3.899 | -0.415 \pm 3.664 | 1.924 \pm 0.313 | 1.370 \pm 0.208 | 0.562 \pm 0.201 |
| | N2N (ours) | 4.823 \pm 6.166 | 0.324 \pm 4.558 | 2.445 \pm 0.481 | 1.770 \pm 0.410 | 0.634 \pm 0.199 |
| Children Playing (2) | Baseline | 0.883 \pm 3.655 | -4.951 \pm 3.013 | 1.795 \pm 0.397 | 1.224 \pm 0.210 | 0.571 \pm 0.182 |
| | N2C | 3.830 \pm 3.580 | -1.403 \pm 3.201 | 1.854 \pm 0.235 | 1.332 \pm 0.152 | 0.550 \pm 0.171 |
| | N2N (ours) | 4.348 \pm 5.370 | -0.636 \pm 3.776 | 2.177 \pm 0.378 | 1.512 \pm 0.248 | 0.620 \pm 0.178 |
| Dog Barking (3) | Baseline | 0.481 \pm 5.024 | -2.881 \pm 6.020 | 1.924 \pm 0.570 | 1.413 \pm 0.461 | 0.561 \pm 0.212 |
| | N2C | 3.438 \pm 3.457 | -0.684 \pm 3.767 | 1.773 \pm 0.326 | 1.326 \pm 0.190 | 0.520 \pm 0.188 |
| | N2N (ours) | 3.990 \pm 5.451 | -0.002 \pm 5.084 | 2.147 \pm 0.535 | 1.550 \pm 0.372 | 0.593 \pm 0.221 |
| Drilling (4) | Baseline | 0.412 \pm 3.952 | -5.340 \pm 3.020 | 1.585 \pm 0.292 | 1.135 \pm 0.101 | 0.524 \pm 0.191 |
| | N2C | 3.621 \pm 3.806 | -0.617 \pm 3.347 | 1.887 \pm 0.366 | 1.352 \pm 0.195 | 0.518 \pm 0.197 |
| | N2N (ours) | 3.961 \pm 5.420 | -0.403 \pm 3.888 | 2.006 \pm 0.471 | 1.413 \pm 0.249 | 0.556 \pm 0.216 |
| Engine Idling (5) | Baseline | 0.467 \pm 3.847 | -5.663 \pm 2.608 | 1.883 \pm 0.560 | 1.217 \pm 0.239 | 0.558 \pm 0.208 |
| | N2C | 3.698 \pm 3.603 | -1.403 \pm 3.010 | 1.916 \pm 0.362 | 1.284 \pm 0.155 | 0.562 \pm 0.204 |
| | N2N (ours) | 4.061 \pm 5.347 | -1.479 \pm 3.648 | 2.272 \pm 0.510 | 1.552 \pm 0.312 | 0.596 \pm 0.210 |
| Gunshot (6) | Baseline | -0.025 \pm 4.151 | -2.631 \pm 6.04 | 1.921 \pm 0.693 | 1.430 \pm 0.484 | 0.519 \pm 0.224 |
| | N2C | 3.831 \pm 3.892 | -0.449 \pm 3.901 | 2.020 \pm 0.47 | 1.458 \pm 0.284 | 0.537 \pm 0.209 |
| | N2N (ours) | 4.400 \pm 6.367 | 0.169 \pm 5.476 | 2.321 \pm 0.739 | 1.718 \pm 0.535 | 0.569 \pm 0.240 |
| Jackhammer (7) | Baseline | -0.175 \pm 4.137 | -5.808 \pm 2.703 | 1.497 \pm 0.293 | 1.097 \pm 0.072 | 0.479 \pm 0.197 |
| | N2C | 3.167 \pm 3.621 | -1.516 \pm 3.029 | 1.821 \pm 0.378 | 1.292 \pm 0.170 | 0.491 \pm 0.200 |
| | N2N (ours) | 3.381 \pm 5.020 | -1.407 \pm 3.431 | 1.898 \pm 0.456 | 1.326 \pm 0.204 | 0.516 \pm 0.229 |
| Siren (8) | Baseline | 1.341 \pm 3.692 | -5.099 \pm 3.006 | 1.822 \pm 0.327 | 1.270 \pm 0.183 | 0.601 \pm 0.182 |
| | N2C | 4.504 \pm 4.062 | -0.058 \pm 3.643 | 1.956 \pm 0.226 | 1.382 \pm 0.164 | 0.580 \pm 0.185 |
| | N2N (ours) | 5.190 \pm 6.354 | 0.606 \pm 4.455 | 2.451 \pm 0.320 | 1.758 \pm 0.299 | 0.656 \pm 0.178 |
| Street Music (9) | Baseline | 0.807 \pm 3.792 | -5.258 \pm 2.963 | 1.762 \pm 0.353 | 1.214 \pm 0.188 | 0.551 \pm 0.194 |
| | N2C | 3.662 \pm 3.594 | -1.210 \pm 3.149 | 1.891 \pm 0.290 | 1.302 \pm 0.150 | 0.564 \pm 0.193 |
| | N2N (ours) | 3.825 \pm 5.047 | -1.036 \pm 3.636 | 2.170 \pm 0.409 | 1.490 \pm 0.240 | 0.603 \pm 0.197 |

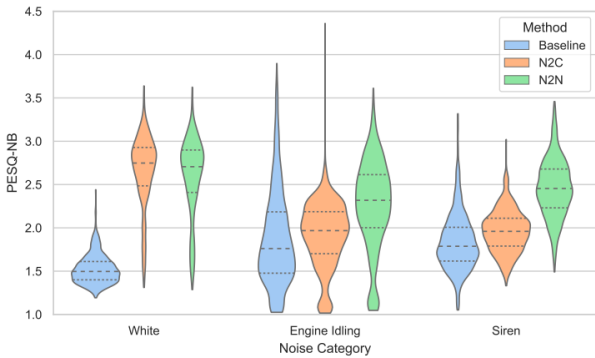


Figure 2: Violin plot comparing the distribution of PESQ-NB for certain noises, pre and post-denoising using N2C and N2N.

stuck in a local optimum [22]. The lack of this ability is why we observe a decrease in intelligibility (STOI) for N2C in Mixed and UrbanSound8K categories 0,2,3,4 and 8, despite SNR improvements.

5. Conclusion

This work proves that deep neural networks can be trained to denoise audio by employing a technique that uses only noisy audio samples as both the input as well as the target to the network, subject to the noise distributions being zero mean and uncorrelated. This is demonstrated by using the DCUnet-20 model to denoise both real-world UrbanSound8K noise categories as well as synthetically generated White noise. Furthermore we see that our proposed Noise2Noise approach in the speech domain produces superior denoising performance compared to the conventional Noise2Clean approach, for low SNR UrbanSound8K noise categories. This is a general conclusion seen across all noise categories and metrics for noises from the UrbanSound8K dataset.

A limitation of this approach is the fact that the noisy training input and target pairs need to have the same underlying clean speech. Although this type of data collection is still practical - for example having multiple microphones in various spatial locations to the noisy speech source - further research should be done to reduce this constraint. The authors hope this paper will encourage better denoising tools for low resource languages, as expensive clean data collection is no longer an obstacle.

6. References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [2] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning image restoration without clean data," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 2965–2974.
- [3] N. Alamdari, A. Azarang, and N. Kehtarnavaz, "Improving deep speech denoising by noisy2noisy signal mapping," *Applied Acoustics*, vol. 172, p. 107631, 2021.
- [4] R. E. Zezario, T. Hussain, X. Lu, H. M. Wang, and Y. Tsao, "Self-supervised denoising autoencoder with linear regression decoder for speech enhancement," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6669–6673.
- [5] Y. Shi, W. Rong, and N. Zheng, "Speech enhancement using convolutional neural network with skip connections," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2018, pp. 6–10.
- [6] Z. Zhao, H. Liu, and T. Fingscheidt, "Convolutional neural networks to enhance coded speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 663–678, 2019.
- [7] F. G. Germain, Q. Chen, and V. Koltun, "Speech Denoising with Deep Feature Losses," in *Proc. Interspeech 2019*, 2019, pp. 2723–2727. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2019-1924>
- [8] A. Azarang and N. Kehtarnavaz, "A review of multi-objective deep learning speech denoising methods," *Speech Communication*, vol. 122, 05 2020.
- [9] C. Valentini-Botinhao, "Noisy speech database for training speech enhancement algorithms and TTS models 2016[sound]." [Online]. Available: <https://doi.org/10.7488/ds/2117>
- [10] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.
- [11] J. Robert, M. Webbie *et al.*, "Pydub," 2018. [Online]. Available: <http://pydub.com/>
- [12] 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.
- [13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [14] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *2013 international conference oriental COCOSDA held jointly with 2013 conference on Asian spoken language research and evaluation (O-COCOSDA/CASLRE)*. IEEE, 2013, pp. 1–4.
- [15] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings," in *Proceedings of Meetings on Acoustics ICA2013*, vol. 19, no. 1. Acoustical Society of America, 2013, p. 035081.
- [16] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating rnn-based speech enhancement methods for noise-robust text-to-speech," in *SSW*, 2016, pp. 146–152.
- [17] S. Kelkar, L. Grigsby, and J. Langsner, "An extension of parseval's theorem and its use in calculating transient energy in the frequency domain," *IEEE Transactions on Industrial Electronics*, no. 1, pp. 42–45, 1983.
- [18] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *6th International Conference on Learning Representations, ICLR 2018*, 2018.
- [19] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.
- [20] 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.
- [21] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4214–4217.
- [22] M. Zhou, T. Liu, Y. Li, D. Lin, E. Zhou, and T. Zhao, "Toward understanding the importance of noise in training neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7594–7602.