“Sorry, I didn’t quite get that.”: Noise Detection and Removal in Speech Audio Using Deep Convolutional Neural Networks

Mustafa Majid (7860730)
Graduate School of Natural Sciences
Utrecht University
Utrecht, the Netherlands
m.m.majid@students.uu.nl

Nicolò Loddo (1531697)
Graduate School of Natural Sciences
Utrecht University
Utrecht, the Netherlands
n.loddo@students.uu.nl

Parsa Beigzadeh (1203754)
Graduate School of Natural Sciences
Utrecht University
Utrecht, the Netherlands
p.beigzadeh@students.uu.nl

Colino Sprockel (3979822)
Graduate School of Natural Sciences
Utrecht University
Utrecht, the Netherlands
c.j.sprockel@students.uu.nl

Coen Kenter (5982065)
Graduate School of Natural Sciences
Utrecht University
Utrecht, the Netherlands
c.kenter@students.uu.nl

Tom Kalkman (6209890)
Graduate School of Natural Sciences
Utrecht University
Utrecht, the Netherlands
t.a.kalkman@students.uu.nl

***Abstract*—Convolutional Neural Networks (CNNs) have been used to remove environmental noise from noisy audio files to make the audio clearer and easier to understand. Wave-U-Net is the model that achieved this with end-to-end audio source separation in the time domain. This paper will look at the performance of Wave-U-Net models on a dataset of noisy speech audio. The performance is measured using both quantitative and qualitative measures, with both giving favorable results for sound quality after the neural networks remove the noise. Performance suffers when the environmental noise is louder than the speech and sometimes resulted in clipping of the speech audio.**

***Index Terms*—Speech enhancement, denoising, convolutional neural network, Wave-U-Net.**

I. Introduction

Recorded speech audio often has background noise that affects the overall quality and usefulness of the recording. Virtual assistants such as Alexa and Siri often complain that they did not catch what a person has said if the current environment is too noisy and the device microphone is capturing unwanted audio as well. Environmental background noise drastically reduces the power of speech recognition software, which may need to remove the noise before further processing of the audio can take place.

Most human beings are usually able to focus on a single sound when hearing multiple simultaneously, making it possible to verbally communicate with each other even with background noise present in the environment. In this paper, we are interested in building a similar system that removes noise from speech audio recordings using a convolutional neural network. Noise removal from audio files is a complex problem [1]. We assume that noise removal yields superior quality audio when the noise removal system is exposed to various categories of noise. Such a system can provide improved audio quality for software used in virtual assistants, hearing aids and provide a clearer voice recording for applications such as online lectures.

II. Related work

Procedural denoisers have been used widely. Recently, neural networks are being used for audio denoising, featuring various architectures.

Abouzid et al. published research in the domain of speech reconstruction [2] focuses on denoising speech recordings using a particular type of convolutional Denoising Auto-Encoders (DAE). Auto-Encoders (AE) encode the input data into a smaller set of important features, to then try to reconstruct the original data in the decoding phase. The first development of a Denoising Auto-Encoder is reported in a paper by Vincent et al. [3] for general types of data, adding a criterion of “robustness to partial destruction of the input” to Auto-Encoders’ design. Their solution consists of providing the Auto-Encoder with raw noisy data as input and training it to output denoised data from the decoding phase by backpropagating the error not with the input as normally is the case in AEs but from the original clean data.

A better de-noising approach is used by Macartney and Weyde [4] utilizing the Wave-U-Net model: a network that focuses on audio source separation with an encoder-decoder approach introduced by Stoller et al. in 2018 [5]. The Wave-U-Net is a machine learning model architecture built upon the U-Net: a convolutional neural network for image segmentation. U-Net is an AE with the addition of *concatenations*: connections between downsampling and upsampling blocks that protects against loss of details. Over the last couple of years, U-Nets have increased in popularity in image and audio upsampling and denoising tasks. By adapting the U-Net to one-dimensional inputs and,

therefore, applicable to audio, Stoller et al. made it possible to perform audio source separation that ultimately has been used in the aforementioned research by Macartney and Weyde to denoise speech recordings. Compared to audio source separation, Macartey and Weyde found that fewer layers were necessary for speech denoising than for the music separation demonstrated by Stoller et al.

III. Data

The dataset to be used is created from merging two datasets together. The first dataset is a subset of audio recordings from the Mozilla Common Voice Corpus 7.0 database, in the English language [6]. The Corpus contains 65 gigabytes of English audio, of which a subset around 13.7 gigabytes, containing around 423 hours of audio, was taken for this research task. Each audio file contains a speech recording of a person saying a simple sentence in English. Some of the sentences can also be in the form of questions, such as "How do you know she didn't?". The context of such sentences is not given with the audio file and is also not necessary for the denoising application. The second dataset contains audio recordings of noises from the UrbanSound8K, which contains 8732 different street noise audio files, that can be categorized into ten types, including dogs barking, sirens, and car horns [7]. The resulting audio files are either trimmed to 4 second length, or extended to 4 seconds if they are shorter. The extension is done by padding a few seconds of silence to the audio files.

A. Pre-processing

Each speech audio file is merged with a random noise audio file to artificially create noisy speech audio data points for the model to learn from. This is achieved using the PyDub library[1] in Python that can be used to merge audio segments by overlaying one audio waveform over the other [8]. Additionally, the merged audio files are also either trimmed or padded to make them all the same length of four seconds, as that was the median length of the audio files before this pre-processing step. The padding of the audio file is done by simply padding with a few moments of no sound, simply to extend the audio clip to make it reach the desired length of four seconds. No effort was made to ensure the speech and noise samples to be mixed were of similar volume, resulting in some mixed samples being inaudible even to humans.

There are two datasets, and each one is to be trained on a separate model. Dataset A has 2,000 noisy speech files for training, while Dataset B has 18,000.

test to cite table: Table I

Dataset	Number of Samples	Total Length (s)
Dataset A	2000	8000
Dataset B	18000	72000

TABLE I: *Number of noisy audio samples in each training set, with total length of audio in seconds in the corresponding training set also listed.*

IV. Approach

A. Model architecture

We use the Wave U-Net architecture [5] which was used for singing voice separation. We use an implementation already written in PyTorch[2]. Two main model types were used, one trained on a dataset A, consisting of 2,000 noisy audio files (Model A), and the other on dataset B, consisting of 18,000 noisy audio files (Model B). Model A has three variants: models A-1, A-2, and A-3 were all trained on dataset A, with varying hyperparameter values. There is only one version of model B, without any hyperparameter tuning, this was due to the larger training time from the larger dataset. The audio separation performance of all models will be examined.

B. Default Wave-U-Net

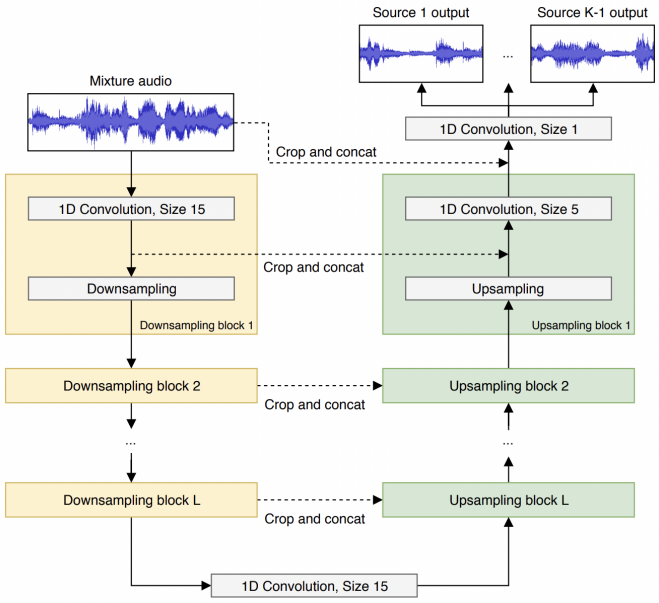

Fig. 1: *Wave-U-Net architecture that takes mixture audio as input and outputs the separated audio sources* [5].

The default Wave-U-Net consists of 6 downsampling (DS) blocks, followed by 6 upsampling (US) blocks, meaning it is a U-Net with 6 levels. Each DS block has a direct connection to the US block of the same level downstream in the network: the feature maps of this block are sent as input to the upsampling block of the same level, this is called a *concatenation*. Following these upsampling blocks there is one final 1D convolution with a *tanh* actvation function to produce the sound output. All DS blocks contain 1 layer of convolution, with zero-padding and a *leakyReLU* activation function. No pooling is performed. The number of feature channels doubles each successive level we go down in the U-Net, starting at 32

[1]PyDub can be found **here**.

[2]can be found on Github **here**.

feature channels up to 1024 feature channels at the bottom of the U. For the upsampling blocks each successive level halves the number of feature channels again. All convolutions have a stride of 4. see Figure 1 for an example of L levels. Note that we train one network per output source.

C. Performance metrics

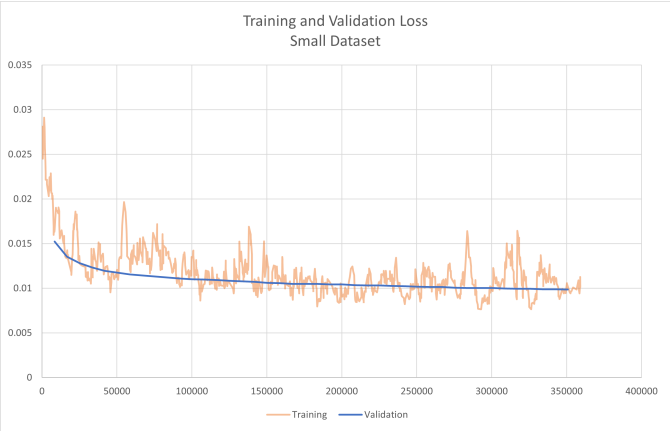

Fig. 2: *L1 training and validation loss for Model A-1 against number of iterations. The validation loss decreases fast then starts plateauing, but the training loss is fluctuating throughout after decreasing somewhat from the beginning of the training process.*

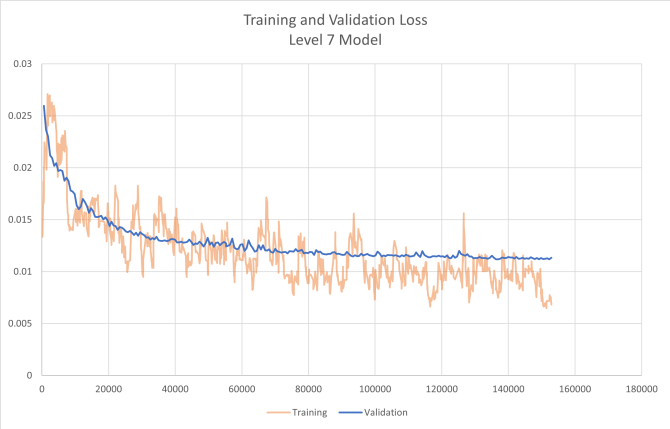

Fig. 3: *L1 training and validation loss for Model A-2 against number of iterations. The training loss and validation loss both tend to decrease with the number of iterations, but training loss is fluctuating significantly. The validation loss plateaus at a higher value than than the loss for A-1.*

Figure 2, Figure 3 and Figure 4 show the L1 training and validation losses for models A-1, A-2, and A-3, respectively, plotted against the number of iterations on the x-axes. The training loss tends to fluctuate but follows an overall decreasing trend, while validation loss tends to decrease more smoothly. Both plateau around 40,000 iterations. Figure 5 shows the L1 training and validation losses for model B, which

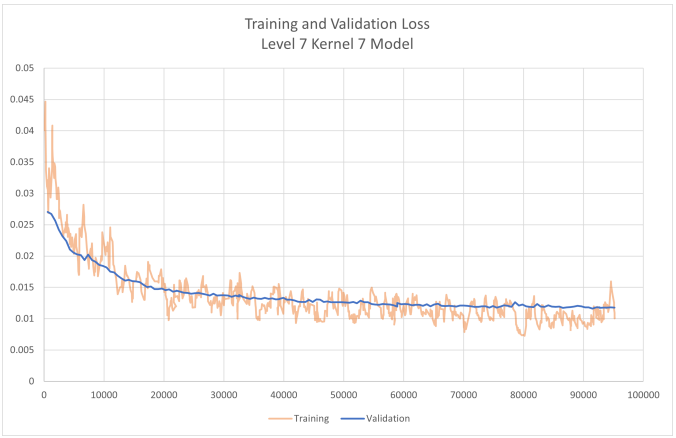

Fig. 4: *L1 training and validation loss for Model A-3 against number of iterations. The training loss and validation loss both tend to decrease with the number of iterations.*

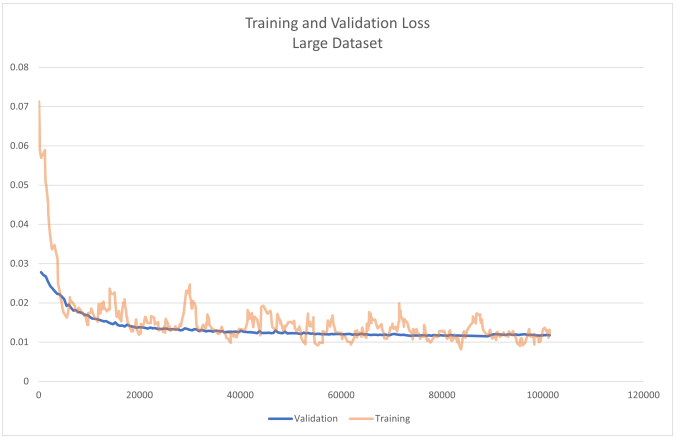

Fig. 5: *L1 training and validation loss for Model B against number of iterations. The training loss and validation loss both tend to decrease with the number of iterations, with less fluctuation in the training loss compared to the A models.*

starts to plateau a lot earlier around 15,000 iterations. In terms of loss values and trends in the graphs, the models are very similar to one another.

To see how our model is going to perform, we will do a quantitative and qualitative performance check. Our quantitative measures will consist of the median and mean of the SDR (Signal to Distortion Ratio). Our qualitative measures are going to be a bit less objective, we are going to listen to the audio files and compare the different results from our models. This will include artifacts left behind by the model and what types of noise are more or less removed.

D. Hyperparameter tuning

Hyperparameters for the model trained on the dataset A were tuned to optimize audio separation performance.

Hyperparameter tuning was not performed on model B, as training the model is very resource-intensive with respect to

Model	Dataset	Kernel	Level	batch_size
B	B	5	6	4
A-1	A	5	6	4
A-2	A	5	7	3
A-3	A	7	7	3

TABLE II: *Hyperparameters for each model.*

time and computational resources, so the default settings for the Wave-U-Net PyTorch implementation were utilized. This model was trained for 20 hours on an RTX3070 GPU with CUDA enabled, completing 35 epochs of training.

On models A-1, A-2, and A-3, various hyperparameter settings were tested, as given in Table II.

For model A-3, up to 7 levels were tested. Testing with more than 7 levels, however, was not feasible due to hardware limitations. The size of the networks generated was larger than our GPU's memory could handle. Model A-3 was trained after tuning of its hyperparameters.

V. Results

A. Qualitative analysis of model performance

Click here to find one denoising example per noise type. these were denoised using model B. Samples were randomly selected from the test set based on two conditions: firstly, the audibility of the speech in the sample must be adequate to the researcher, and secondly, a sufficient amount of noise present in the sample.

We can see that model B is consistently able to remove almost all noise from the samples, with at most a small distortion remaining in the denoised speech samples. Some exceptions will be discussed later.

The results of performing denoising on the same samples using the best model A can be found best model A can be found here.

Given that model B's training set is 9 times larger than model A's, we unsurprisingly notice that model B performs better on denoising than model A on these unseen test samples.

B. Model limitations

Listen here to some samples in which model B failed to adequately remove the noise. In an audio file where a drilling noise is playing over speech audio, the drilling noise very much overshadows the speech in the mixed audio. When the model separates the two, the voice audio does not have the loud drilling noise anymore, but the speech itself remains distorted. Additionally, the noise output from the model contains some of the speech signal, as a person talking can be heard in the noise output, which may explain why the speech output sounds distorted and not clear. It is not possible to make out what the person is saying, before or after audio source separation.

A possible cause of this is audio clipping, where the amplitude of an audio signal is flattened at a certain height because it is too loud. This flattening causes a loss of information, and potentially destroys the speech irrecoverably.

In contrast, when the background noise is not so loud that it overshadows the voice in the mixed audio, such as with street music as opposed to drilling and jackhammer noises, the resulting speech audio is quite clear. There are still some remnants of the background noise present in the output speech audio, but this does not significantly affect the clarity of the speech.

This is one example, where the noise is not that loud, but model B fails to remove didgeridoo sounds in some parts of the audio, we theorize this is because a didgeridoo sounds similar to a human voice and the model is struggling to differentiate.

The qualitative evaluation of results was done by the authors listening to the mixed audio and comparing it to the speech audio output by each model. The quality of the denoised speech audio files varies based on the type of noise that was present in the mixed audio input. If the mixed audio contained loud noises such as construction drill or gun shot sounds (class "drilling" and "gun_shots" in the in the dataset, respectively), the resulting speech audio was not very clear. However, this is most likely a limitation in the data itself as opposed to the neural network. Noises that were not loud in comparison allowed both models to produce better speech audio outputs. Model B consistently produced better-sounding audio compared to Model A, even though the latter had hyperparameters tuned to optimize performance. This is likely due to the much larger training set model B utilized to perform better audio separation.

C. Quantitative analysis of model performance

Model	Type	Mean	Median
B	Noise	12.38	10.66
	Voice	8.77	8.37356
A-1	Noise	10.59	8.98
	Voice	6.58	7.1979
A-2	Noise	10.56	9.36
	Voice	7.60	7.483327
A-3	Noise	10.47	9.37
	Voice	7.58	7.267062

TABLE III: Mean and Median for the SDR metric.

Based on the Wave-U-Net architecture, a suitable measure to evaluate the performance of audio separation models quantitatively is to use the Signal-to-Distortion ratio [5]. Following the approach from the original Wave-U-Net paper, we take the median SDR values as opposed to mean or standard deviation, since our vocal audio file data is not normally distributed, as it was picked from Mozilla Common Voice English Corpus at random.

We find that Model B achieves the highest SDR. Model A-2 and A-3 appear to be second best, followed by A-1 in last place. We find that on the small data set, models with 7 levels outperform the model with 6 levels.

VI. Further Research

Macartey and Weyde [4] found that fewer layers were necessary for speech denoising than for the music separation of the original Wave-U-Net paper. Hyperparameter tuning

on the small training set showed that models with 7 levels outperform models with 6 levels. An obvious next step is to train a new model with 7 levels on the big dataset and see how that affects performance. Availability of more powerful computational resources will help with this task.

As the U-Net architecture has been used on images and audio files, it can be further investigated how this type of model will work on signal separation of different types, such as in electrical signal source separation. Additionally, one can investigate a completely different approach using unsupervised learning models where an unlabelled mixed audio file is given to a model, and the model separates the relevant audio sources without being instructed on which audio sources to target.

VII. Conclusion

In conclusion, The U-Net architecture is very suitable for denoising. We advise the use of a U-Net denoising model as a possible step, followed by a conventional speech-to-text model for improving speech-to-text systems. Further research in this area could focus on modifying this architecture to separate various voices in situations where many people speak at the same time. Audio recording technology limitations also must be accounted for, as the volume of the sounds affects the performance of the Wave-U-Net models.

References

[1] M. Dogra, S. Borwankar, and J. Domala, "Noise removal from audio using cnn and denoiser," in *Advances in Speech and Music Technology*, Springer, 2021, pp. 37–48.

[2] H. Abouzid, C. Otman, O. Reyes, and S. Ventura, "Signal speech reconstruction and noise removal using convolutional denoising audioencoders with neural deep learning," *Analog Integrated Circuits and Signal Processing*, Sep. 2019. DOI: 10.1007/s10470-019-01446-6.

[3] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," Jan. 2008, pp. 1096–1103. DOI: 10.1145/1390156.1390294.

[4] C. Macartney and T. Weyde, *Improved speech enhancement with the wave-u-net*, 2018. arXiv: 1811 . 11307 [cs.SD].

[5] D. Stoller, S. Ewert, and S. Dixon, *Wave-u-net: A multi-scale neural network for end-to-end audio source separation*, 2018. arXiv: 1806.03185 [cs.SD].

[6] *Mozilla common voice*. [Online]. Available: https : / / commonvoice.mozilla.org/en/datasets.

[7] 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.

[8] *Pydub*. [Online]. Available: http://pydub.com/.