Hearing Aid Speech Enhancement Using U-Net Convolutional Neural Networks

Paul Kendrick[1]

[1]Music Tribe

kenders2000@gmail.com

Abstract

This paper presents an entry into the Clarity-2021 challenge for hearing-aid speech enhancement (CEC1). A U-Net Convolution Neural Network was trained using the provided training data to predict clean spectrograms. The enhanced signal was then adapted to a listener's audiogram using a hearing aid model that includes frequency equalization and dynamics processing. The average MBSTOI over the dev dataset was 0.56. The method showed a significant improvement over the baseline algorithm.

Index Terms: speech enhancement, clarity challenge, u-net, convolution neural network

1. Introduction

The Clarity-2021 challenge [1] for hearing-aid speech enhancement was set up as a response to the increasing numbers of people with hearing loss [2]. A dataset was provided for training, development and evaluation of a speech enhancement algorithm. The challenge requires a system that doesn't look ahead more than 5 ms. This paper describes a submission which uses the U-Net CNN architecture to enhance spectrograms. The U-Net was originally introduced for biomedical imaging [3] and more recently for audio source separation [4]. Section 2.2 describes how the signal is first enhanced in a listener-agnostic way using the U-Net. Section 2.3 describes the hearing aid model used to adapt the cleaned signal to the listener. Code is available on GitHub [5].

2. Methodology

The speech enhancement algorithm was spilt into two steps. First, a U-Net CNN predicts a clean signal from spectrograms of the 3 channels of each hearing aid (no listener adaption). Second, standard hearing aid processing algorithms are applied to adapt the signal to a listener via the audiogram. The model operates at a sampling frequency of 16 kHz.

2.1. Datasets

The Clarity-2021 dataset consists of 10,000 unique examples, split into train (6000), dev (2500), and eval (1500). Examples consisted of a single speech source captured in a simulated acoustic space with a competing noise source (competing talker or domestic noise). A Binaural Head Related Impulse response simulates the sound at three microphones of a behind-the-ear hearing aid. Each example is around 6 s long and the interferer precedes the onset of the target by 2 s and follows the offset by 1 s. Clean targets were provided for the train and dev sets.

2.2. Listener-agnostic noise reduction

Figure 1 shows the speech enhancement system. All 6 input channels are used to predict the clean signal at left ear. To meet the 5 ms lookahead limit, the input is processed using a 80 sample *input frame* with a 70 sample hop. To provide the network with more context, previous samples are prepended to the signal, until a 96,000 sample (6 s) *input processing window* is defined. Zero padding is used when no signal history is available. As 'the interferer always precedes the onset of the target by 2 s and follows the offset by 1 s' [1], this additional context ensures that the network has access to the some of the isolated interferer.

A Short-time Fourier transform (STFT) is computed from each *input processing window* (window length 1024, hop 256, hanning window). No logarithmic amplitude scaling or perceptual frequency binning is applied; the resulting STFT dimensions are 376x513. A trained U-Net is used to predict the clean magnitude STFT at the left ear from all 6 noisy magnitude STFTs (see 2.2.1). The cleaned magnitude STFT and the noisy phase STFT from channel 1 of the left ear are combined and the inverse STFT computed. This results in 96,000 samples referred to as the *output processing window*.

The last 80 samples (5 ms) of each *output processing window* is the current *output frame*. The 1st half of a 20-sample hanning window is applied to the 1st 10 samples of the *output frame*, the 2nd half of the hanning window is applied to the last 10 samples. The signal is synthesized by overlapping and adding the *output frames* with an overlap of 10 samples (0.625ms). Overlap-add is required to prevent artifacts due to jumps in level between adjacent frames. This occurs as the U-Net operates independently on each *input processing window*; it has no memory of past outputs which results in a discontinuity in the level between adjacent *output frames*.

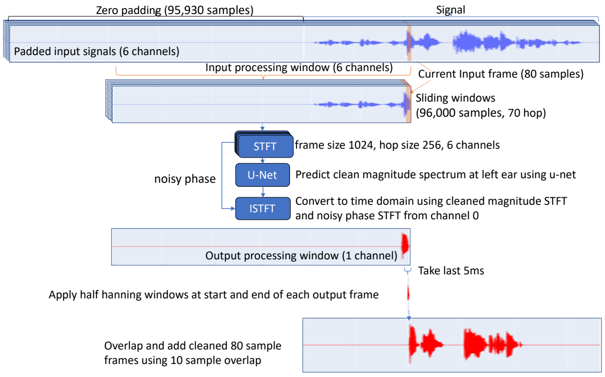
Zero padding (95,930 samples)
Signal
Padded input signals (6 channels)
Input processing window (6 channels)
Current Input frame (80 samples)
Sliding windows
(96,000 samples, 70 hop)
STFT
frame size 1024, hop size 256, 6 channels
noisy phase
U-Net
Predict clean magnitude spectrum at left ear using u-net
ISTFT
Convert to time domain using cleaned magnitude STFT
and noisy phase STFT from channel 0
Output processing window (1 channel)
Take last 5ms
Apply half hanning windows at start and end of each output frame
Overlap and add cleaned 80 sample
frames using 10 sample overlap

Figure 1: *Speech enhancement overview. Single channel shown.*

The U-Net is trained to output the signal at the left ear. However, by simply swapping the ears, right-for-left, for each hearing aid channel, the head is effectively mirrored and the clean magnitude response at the right ear can now be predicted with the same U-Net. The algorithm lookahead is 70 samples (4.375 ms).

2.2.1. *U-Net training*

The U-Net is similar to the deconvolution network [6] where each convolutional layer halves the size of the input but doubles the number of filters. This produces a small but deep representation which is then decoded using up-sampling layers back to the original size. The U-Net includes skip connections between the encoder and decoder at equivalent resolution levels. This allows low-level information to flow from the encoder to decoder. Attention gates on skip connections learn to suppress irrelevant regions while highlighting useful features [7]. The implementation is based on the Customizable U-Net keras implementation [8]. Figure 2 gives an overview.

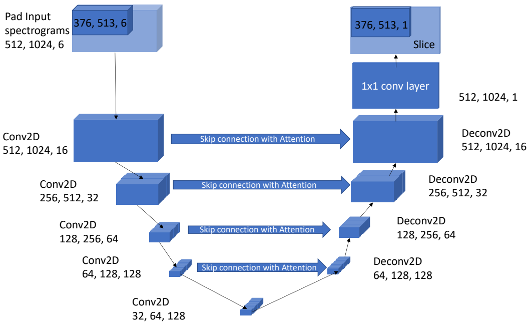

Figure 2: *U-Net, dimensions (frames, bins, channels)*

The input spectrogram is zero-padded so each dimension is a power of two so the resolution can be continuously halved/doubled through the U-net. Each Conv2D layer in the encoder consists of; a 2D convolution layer, a batch norm layer, a dropout layer (dropout rate of 0.3), a 2D convolution layer and a batch norm layer. Between each encoder layer, max-pooling with a pool size (2, 2) performs down-sampling. Deconv2D blocks consist of a 2D transposed convolution layer for up-sampling, concatenated with the skip connection; a 2D convolution layer, a batch norm layer, a 2D convolution layer and a batch norm layer. 1x1 convolution maps the output to a single channel. The output is sliced to remove the padding.

The U-net was trained on the Clarity training dataset [2]. For the training data an *input frame* size of 80 samples was used with an *input processing window* of 96,000 samples. This results in 1372 sample input processing windows. To reduce memory 10 % of the *input processing windows* were retained at random for training. This results in a 378x376x513x6 STFT tensor for each example. The target magnitude STFTs are calculated in the same way, but from the clean utterance before processing. The cross-correlation between the noisy signal (channel 1, left ear) and the clean utterance is used to ensure the targets are synchronized with the hearing aid signals. This results in a 378x376x513x1 STFT tensor for each target. The U-Net was optimized using mean absolute error loss, Adam optimization and a learning rate of 0.001 (TensorFlow 2.4.1). Training was terminated after 10 Epochs which took 22 Days using a Nvidia GTX 1080ti with 11Gb of memory, an intel i7 cpu, and 32 Gb ram.

2.3. Signal adaption to listener's audiogram

A simple hearing aid model was employed consisting of a 4-band filter bank processor, a 2-channel compressor and a soft clipping processor.

2.3.1. *Filter bank*

A filter-bank was applied to the cleaned signal. Gain was applied to each band prior to recombination. An audiogram with pure tone sensitivity levels at 250 Hz, 500, 1 kHz, 2 kHz, 3 kHz, 4 kHz, 6 kHz and 8 kHz was provided for each listener. The gain to apply to each band was calculated using a simple heuristic:

$$G_{ij} = \min(G_{max}, T_{ij} - T_{best}) \qquad (1)$$

G_{ij} is the gain in band j at ear i, G_{max} is the maximum allowed gain (30 db), T_{ij} is the threshold in band j at ear i, and T_{best} is the threshold of the most sensitive band (both ears). The result is a whitening of the frequency response for that listener, although extreme gains are prevented. As each gain is relative to T_{best}, some overall additional gain may be required depending on the average loss. Due to the 16 kHz sampling rate, the highest frequency band used was centered on 6 kHz but the gain is adjusted as the average power in the 6 & 8 kHz bands. FIR filters were used to ensure a linear phase response. Due to the 70-sample look-ahead employed by the speech enhancement, any further processing can only look ahead up to 10 samples. This limits the filter length. A type-I FIR filter has a symmetric impulse response with the center of symmetry at tap (N-1)/2; where N is odd. An FIR filter with 11 taps was used and a zero-phase filter realized by looking 5 samples into the future. This filter cannot affect frequencies below around 1.5 kHz, therefore the 2 kHz band is taken as the lowest frequency band, where the gain is the average power of the 250 Hz, 500, 1 kHz and 2 kHz bands. This results in in four bands:

- A low-pass filter with a cut-off of 2500 Hz
- A band-pass filter centered at 3 kHz with cut-off frequencies of 2.5 kHz and 3.5 kHz
- A bandpass filter centered at 4 kHz with cut-off frequencies of 3.5 kHz and 5 kHz
- A high-pass filter with a cut-off of 5 kHz

Most of the speech energy will be within the 2500 Hz band and there is significant overlap between adjacent bands; as such this the system is compromised by the lookahead limit. In future models the gain could be applied directly to the spectrograms prior to reconstruction to negate this limitation. FIR filters were designed using the window technique (hanning window). Once gains are applied to each band, the signal is recombined, and the 5 sample look ahead applied. When combined with the U-Net noise reduction, this results in a total system look-a-head of 4.6875ms; well within the 5ms limit.

Two channel compression [9] [1] was applied after the filter bank (one channel per ear) using a -6 dB threshold, a ratio of 5:1, attack time of 4ms and a 75ms release time.

Soft clipping [1] was applied to prevent sample values of greater than ±1 using the following:

$$f(x) = \begin{cases} x > 1, & (21-1)/21 \\ -1 \geq x \leq 1, & x - x^{21}/21 \\ x < -1, & (21-1)/21 \end{cases} \qquad (4)$$

3. Results and Discussion

The mean MBSTOI over the development dataset was 0.56 and the median was 0.57. The baseline method evaluated over the development dataset showed a mean and median MBSTOI of 0.41.

The inference time for the generation of a single example was around 6 minutes (Nvidia GTX 960 with 4Gb of memory, an intel i7 cpu, and 32 Gb ram). This is extremely high however it should be noted that the U-Net predicted a full 6 s long spectrogram every 70 samples and an inverse STFT is calculated for the full 6 s. As most of this signal is discarded, there is clear scope for significant optimization.

The optimal parameters for the hearing aid model were found by repeatedly evaluating a subset of the dev set (the first ten scenes or 30 sound files). A subset was used due to time constraints and as such the scope of the investigation was relatively small.

Table 1: *Results of optimising the hearing aid parameters, T_{best} method refers to whether this is calculated per-ear or over both ears.*

	T_{best} method	G_{max}	Clipping degree	MBSTOI
Baseline				0.31
U-Net (no hearing aid)				0.54
U-Net (with hearing aid)	Per-ear	40	3	0.55
U-Net (with hearing aid	Per-ear	40	21	0.55
U-Net (with hearing aid)	Both	40	21	0.56
U-Net (with hearing aid	Both	30	21	0.57

Table 1 shows that even without the hearing aid model the U-net outperforms the baseline method by a MBSTOI of 0.23. The degree of clipping had little impact. Setting the gain relative to the best threshold band over both ears appears to perform better compared with using the best threshold per ear. A lower limit of 30 dB on the maximum gain performed slightly better than 40dB. Overall, the hearing aid only marginally increased the MBSTOI by around 0.03, this is most likely due to the compromised filter bank design imposed by the lookahead limitation. Figure 3 compares MBSTOI performance of the proposed algorithm, at different SNRs, with the baseline.

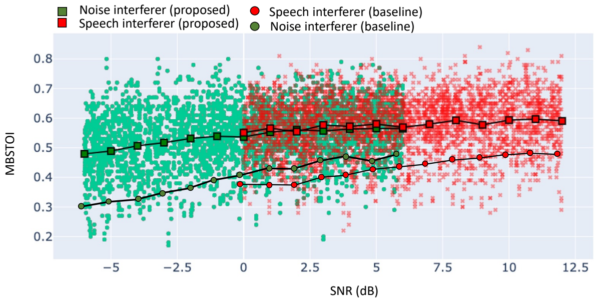

Figure 3: *Comparison of MBSTOI performance with baseline.*

For both speech and noise interferers there is a weak positive correlation between MBSTOI and SNR (speech: $\tau = 0.12$, $p < 0.001$; noise: $\tau = 0.27$, $p < 0.001$). The correlation is weaker compared to the baseline results (speech: $\tau = 0.35$, $p < 0.001$; noise: $\tau = 0.49$, $p < 0.001$). This indicates that the proposed method is more robust to higher levels of interferers. Figure 3 also shows that the MBSTOI performance is similar for speech and noise interferers at the same SNR, whereas for the baseline, the performance is lower when the interferer is speech.

4. Conclusions and Further work

In this paper an approach for speech enhancement for hearing aid users using a U-Net showed an improvement in the MBSTOI measure compared with the baseline method.

The hearing aid model provided only a marginal increase in the MBSTOI measure; a better approach would incorporate the frequency adaption into the network. The method efficiency could be improved upon by some simple optimizations; the CNN output scope could be restricted to only the last 80 samples, to reduce the model size; and the inverse STFT optimised. The resulting model was relatively small (2 million parameters) some further hyperparameter tuning may yield improvements. The loss function used was mean absolute error, loss functions that take into account the human perceptual system will likely yield better results. It would be interesting to include the STFT forward and backward transforms as layers at the start and end of the U-Net, this has the advantage of defining a loss function in the time-domain which would include phase. For the same reason it would be interesting to compare the performance of the U-Net with the wave-U-net, which has a very similar structure but takes the raw waveform as input, and all convolutions are 1D.

5. References

[1] S. Graetzer, J. Barker, T. J. Cox, M. Akeroyd, J. F. Culling, G. Naylor, E. Porter, and R. Viveros Muñoz, “Clarity-2021 challenges: Machine learning challenges for advancing hearing aid processing,” *in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*, Brno, Czech Republic, 2021.

[2] Hearing loss is on the rise! “https://www.who.int/deafness/world-hearing-day/World-Hearing-Day-Infographic-EN.pdf”, accessed: 2021-6-22

[3] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, LNCS, Vol.9351: 234--241, 2015

[4] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, A. and T. Weyde, “Singing Voice Separation with Deep U-Net Convolutional Networks” *Proceedings of the 18th ISMIR Conference*, Suzhou, China, October 23-27, 2017

[5] U-Net speech enhancement for hearing aids, “https://github.com/kenders2000/u_net_speech_enhancement”, accessed: 2021-9-10

[6] H. Noh, S. Hong, and B. Han. “Learning deconvolution network for semantic segmentation.” *In Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[7] O. Oktay, Jo Schlemper, L. L. Folgoc, M. J. Lee, M. Heinrich, K. Misawa, K. Mori, Steven G. McDonagh, N. Hammerla, Bernhard Kainz, Ben Glocker, D. Rueckert, “Attention U-Net: Learning Where to Look for the Pancreas”, *1st Conference on Medical Imaging with Deep Learning* (MIDL 2018), Amsterdam, The Netherlands, 2018.

[8] Keras U-Net, “https://github.com/karolzak/keras-unet”, accessed: 2021-6-22

[9] D. Giannoulis, M. Massberg, and J. D. Reiss. "Digital Dynamic Range Compressor Design: A Tutorial and Analysis." *Journal of Audio Engineering Society*. Vol. 60, Issue 6, 2012, pp. 399-408.