ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414695

SPEECH ENHANCEMENT AUTOENCODER WITH HIERARCHICAL LATENT STRUCTURE

Koen Oostermeijer[⋆] *Jun Du*[⋆] *Qing Wang*[⋆] *Chin-Hui Lee*[†]

[⋆]University of Science and Technology of China, Hefei, China
[†]Georgia Institute of Technology, Atlanta, USA
keen@mail.ustc.edu.cn, qingwang2@ustc.edu.cn, ✉ jundu@ustc.edu.cn, chl@ece.gatech.edu

ABSTRACT

A new hierarchical convolutional neural network-based autoencoder architecture called SEHAE (Speech Enhancement Hierarchical AutoEncoder) is introduced, in which the latent representation is decomposed into several parts that correspond to different scales. The model consists of three functionally different components. First, a stack of encoders generates a set of latent vectors that contain information from an increasingly larger receptive field. Second, the decoders construct the clean speech in a stage-wise and additive fashion, starting from a learned initial vector. The third component, which we call funnel networks, is tasked with "knitting" together the outputs of the previous decoder and the encoder to compute latent vectors for the next decoder. Several options for initial vectors are explored. Experiments show that SEHAE achieves significant improvements for the considered speech quality and intelligibility measures, outperforming a denoising autoencoder and other stepwise models. Furthermore, its internal workings are investigated using the intermediate results from the decoders.

Index Terms— Speech enhancement, Hierarchical autoencoder, Convolutional layers, SEHAE

1. INTRODUCTION

In speech enhancement the objective is to increase the speech quality and intelligibility of utterances corrupted by background noise. More precisely, one seeks to reduce the strength of the background noise while preserving the quality of the speech. This has applications in telecommunications [1] and hearing aids such as cochlear implants [2–4].

Popular methods from before the advent of the deep learning paradigm include classical methods such as Wiener filtering [5], spectral subtraction [6], minimum-mean square error (MMSE) based spectral amplitude estimator [7], Karhunen-Loéve transformation (KLT) [8], non-negative matrix factorization (NMF) [9].

The earliest applications of deep learning to speech enhancement made use of feedforward neural networks (FNN), i.e. multilayer perceptrons (MLP) [10–12]. These methods worked by extracting log-power spectra (LPS) features and used a sliding window to estimate the clean speech frame by frame. One drawback of this method was that the only information available to the network came from the features from within the window. This was solved by recurrent neural networks (RNN), which, per construction, have access to information from all previous frames [13–15]. A third type of network that later gained traction is convolutional neural networks (CNN) [16]. Compared to FNNs their field of reception scales better with the size of the model and compared to RNNs they are easier to parallelize. They commonly have a cone-shaped autoencoder structure [16–18] in which the input is brought down to a latent representation by the encoder and is subsequently reconstructed by the decoder, while removing the noise. Often, residual connections [19] are added between the encoder and decoder layers to improve the flow of information and gradients. Historically, CNN-based methods have been known to suffer from over-smoothing [12, 20, 21]. Several methods have been proposed to address this, such as (Frequency domain) Speech Enhancement Generative Adversarial Network, (F)SEGAN for short [22, 23] and loss function-based methods [24, 25]. The former uses a generative adversarial network (GAN) to make the output of the autoencoder more like real speech samples, while the latter focus on enhanced loss functions to make the network better able to represent the nuances of speech.

In this paper, we introduce a new way to deal with this by means of a novel hierarchical speech enhancement architecture called SEHAE (Speech Enhancement Hierarchical AutoEncoder). It is inspired by the recent Nouveau Variational Autoencoder (NVAE) [26], which uses a structure reminiscent of ours: it uses series of encoders and decoders and starts from an initial vector. However, it is intrinsically different as it does not make use of funnel networks and is made for image generation rather than denoising.

2. ARCHITECTURE

In this paper we introduce a novel speech enhancement architecture called SEHAE that operates on LPS features in the time-frequency domain. As its name implies, its defining feature is its multi-stage hierarchical structure. Instead of a single latent representation, SEHAE uses a series of encoders that output latent vectors that at each step contain increasingly more coarse-grained and global information. Because of this decomposition, the network is explicitly adept to handle the different scale structures that are present in the log-spectrogram data.

The decoding stage is done by a series of decoders that construct the clean target speech in an additive fashion, starting from an initial vector. Each individual decoder outputs a log-spectrogram, which grants us the ability to gain insight into how the algorithm performs the different stages of construction. The initial vector is in general a learnable parameter that functions as a canvas on which the rest of the reconstruction is added. The length of the vector is equal to the number of frequency bins. For each sample the vector is tiled in the time direction to make it the same size as the sample. This can be roughly interpreted as a frequency-dependent bias term.

Finally, a third type of network, referred to as a funnel network, is introduced that is tasked with knitting together the encoder and decoder outputs. There is one for each step in the encoding and de-

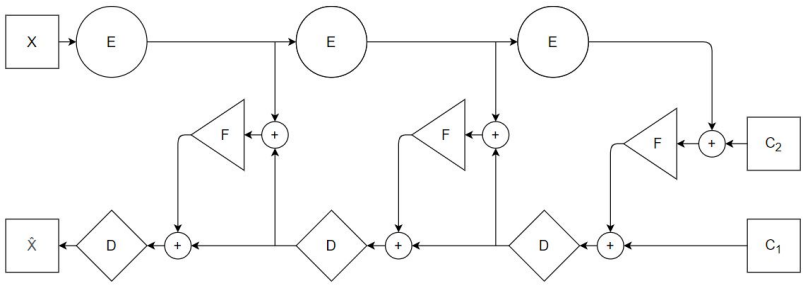


Fig. 1: SEHAE architecture. The components E, F and D denote the encoders, funnels and decoders, respectively. C_1 and C_2 are the initial vectors

coding process. At each step, the latent vector of the encoder and the output of the decoder of the previous step are concatenated and fed into the funnel network to obtain a new latent representation for the next decoder. This gets concatenated with the output of the previous decoder and used as input for the next decoder. The first funnel uses an initial vector that is in general not equal to the canvas vector of the first decoder. Fig. 1 shows the SEHAE architecture in more detail.

3. CONVOLUTIONAL LAYERS

Convolutional layers are a type of neural network layer in which the output is computed by convolving a kernel over the input data. Generally, the input and the output have a height, width and number of channels. Consider an input $X \in \mathbb{R}^{C\times H\times W}$, where C, H and W denote the number of channels, the height and the width, respectively, and a kernel $K \in \mathbb{R}^{C\times C\times H_K\times W_K}$, where the first two dimensions describe the connection strength between the different channels and the latter two are the height and width of the kernel, respectively. Furthermore, a bias term $B \in \mathbb{R}^C$ is added to each channel:

$$Y_{c,l,k} = B_c + \sum_{j,m,n} X_{j,l+m-1,k+n-1} K_{c,j,m,n}. \tag{1}$$

To prevent the output image from being smaller than the input, the outer edges of the input are padded with zeros. This is particularly useful for residual connections.

3.1. Depthwise Convolutions

For depthwise convolutions [27], there are no connections between different channels:

$$K_{i,j,m,n} = 0 \,\forall\, i \neq j. \tag{2}$$

Hence the number of input and output channels is required to be the same. This reduces number of parameters considerably and has been shown to be effective in, for example, [28].

3.2. Squeeze-and-Excite

For squeeze-and-excite layers [29], before adding the input, the intermediate result is averaged over the height and width dimensions of the image. This results in a vector with size equal to the number of channels. This vector is then fed into two FNN layers, and squeezed by a sigmoid function to lie in the range $[0, 1]$. These are weights $\omega_i, i = 1, \ldots, C$ with which the channels are multiplied. This operation allows global properties of the picture to be communicated to the next layers:

$$\omega_i = (\sigma \circ f_2 \circ \mathrm{ReLU} \circ f_1)\left(\frac{1}{HW}\sum_{m,n} X_{i,m,n}\right), \tag{3}$$

where the $f_k(x) \equiv W_k x + b_k$ are fully connected layers with linear activation function, and $\sigma(x) = 1/(1+\exp(-x))$ and $\mathrm{ReLU}(x) = \max(0, x)$ are the sigmoid and rectified linear unit (ReLU) functions, respectively.

4. MODEL SPECIFICATIONS

All convolutional layers in our model are preceded by a batch normalization (BN) layer and a Leaky ReLU activation function with negative slope 0.05. The specific architectures of each of the components were optimized by a component-wise greedy grid search.

4.1. Encoders

As previously explained, the stack of encoders generates an increasingly more long-range latent representation. The first encoder perceives a relatively small area and each consecutive encoder increases the receptive field, i.e. the area the network has access to information from. This is further increased by the funnel networks. In addition, at the end a squeeze-and-excite layer is added so as to access global averaged information. Each encoder consists of three convolutional layers, the middle one of which is of the depthwise kind. They make use of (3×3) sized kernels. Furthermore, a residual connection is added to improve information flow. The encoder unit is illustrated in Fig. 4.

4.2. Funnels

The funnel networks consist of two (3×3) convolutional layers, as shown in Fig. 3. Together with the encoders, this results in receptive fields of size $(11, 17, 23) \approx (0.35, 0.55, 0.74)$ seconds for each step, respectively.

4.3. Decoders

The decoders have a relatively small receptive field; they consist of four convolutional layers with kernel sizes (3×3), (1×1), (3×3) and (1×1), respectively. The second one is depthwise. Like the encoders, they also have a skip connection, see Fig. 2.

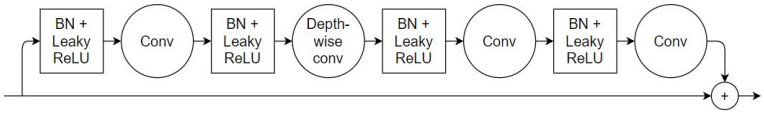


Fig. 2: Decoder unit

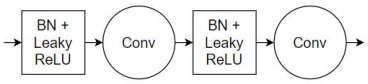


Fig. 3: Funnel unit

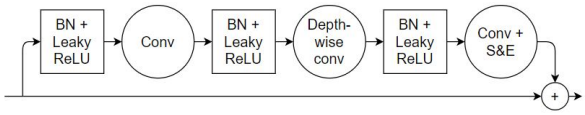


Fig. 4: Encoder unit

4.4. Canvas Vectors

In this paper we explore several options for the canvas vectors C_1 and C_2. The most general case is where they are both independent learnable parameters: $C_1 \neq C_2$. Another option is to set them equal: $C_1 = C_2$. Furthermore, we consider the case where they are both equal to the input: $C_1 = C_2 = X$. These constitute the different versions of SEHAE.

5. EXPERIMENTAL RESULTS

Our models were trained and tested on subsets of WSJ0 [30] for speech and DEMAND for noise [31], with mutually exclusive speakers and noise types, respectively. The former consisted of 82 speakers for the train set and 9 speakers for the test set. Data was generated with a 50% overlap, which yielded a 13 hour and 42 minute dataset, respectively. The noise was divided into a train set consisting of the Metro, Square, Café, Station, Restaurant, Meeting, Hallway, Park, Field and Washing noises and a test set containing the River, Cafeteria and Traffic noise types. We chose this test set such that the noise types belonged to different groups, e.g. Restaurant, Café and Cafeteria are in the same group, namely eating and drinking venues. Each noise type had a total duration of 5 minutes.

At the start of each epoch noise was shuffled and sampled anew and combined with the speech to reduce overfitting. The speech-to-noise ratios (SNRs) we trained and tested at are SNR = -5, 0, 5 dB. Sampling was done at 16 kHz after which a short-time Fourier transform was applied with step size 256 and frame length 512, resulting in 32 ms frames with 257 frequency bins, ranging from 0 to 8 kHz. These were converted to LPS features and were split into time slices of size $40 \approx 1.3$ seconds.

We used a Radam optimiser [32] and an E^{2}STOI loss function [33], which is an ESTOI-based loss function [34] that has been shown to perform better than the mean squared error loss function. For comparison we employed two objective measures, namely the Short-Time Objective Intelligibility (STOI) [35] and the Perceptual Evaluation of Speech Quality (PESQ) [36].

The different versions of SEHAE are compared against a nine-layer PL-LSTM (Progressive Learning LSTM) [37], another stepwise architecture, which uses intermediate targets to boost learning performance. Moreover, to investigate the effect of the hierarchical nature of SEHAE we have constructed an autoencoder using the encoders and decoders as described in the previous chapter. This model uses a sequence of three encoders and three decoders and has been scaled up to match the parameter count of our SEHAE models, about $4.5 \cdot 10^4$.

The STOI and PESQ results of the different systems across the three test noise types are listed in the Table 1, Table 2 and Table 3 below. Numbers in bold font indicate the column-wise best results. In each table the first row refers to the STOI and PESQ values of the noisy speech, i.e. the input. The second and third row are the benchmarks PL-LSTM and the autoencoder as described above. The final three rows correspond to the different versions of SEHAE, which were described in section 4.4.

It can be seen that SEHAE outperforms the benchmark models across all noise types and SNR levels for both STOI and PESQ. Although there are large improvement over PL-LSTM, the difference between the autoencoder and SEHAE is relatively small but significant: an improvement of 2.7% STOI and 0.11 PESQ at -5 dB SNR averaged over the three noise types. Therefore, even though the improvements provided by the use of the hierarchical structure are significant, this suggests that a large part of the improvements come from the units themselves.

Out of all the SEHAE versions, the one in which the initial vectors C_1 and C_2 have been set equal to the input X yielded the best results for all STOI tasks and for all but one PESQ tasks for the River and Traffic noise types. This can perhaps be explained by the fact that the input contains more information compared to the smaller and static vector, and is closer to the clean speech target.

For the Cafeteria noise the versions with learnable initial vector take the crown, albeit by a small margin. Comparing the two versions $C_1 \neq C_2$ with $C_1 = C_2$, we see that the former yields superior results on seven out of nine STOI and PESQ tasks. This might be simply due to the fact that having two vectors instead of one gives the model more degrees of freedom.

However, the results of all three were relatively close to each other: The STOI values and PESQ values of the different SEHAE types are within 0.4% and 0.02 of each other, respectively. In the future more experiments with more types of noise could give insight into what causes this difference.

Table 1: River noise; STOI(%) and PESQ comparison

	SNR = -5		SNR = 0		SNR = 5	
	STOI	PESQ	STOI	PESQ	STOI	PESQ
Noisy	63.4	1.59	78.4	1.91	88.4	2.26
PL-LSTM	68.3	1.71	81.9	2.05	90.2	2.41
Autoencoder	73.1	2.05	84.7	2.44	91.9	2.76
$C_1 \neq C_2$	76.6	2.17	87.5	2.54	93.6	2.84
$C_1 = C_2$	76.9	2.10	87.3	2.49	93.3	2.82
$C_1 = C_2 = X$	**77.5**	**2.22**	**87.9**	**2.59**	**93.7**	**2.86**

Table 2: Traffic noise; STOI(%) and PESQ comparison

	SNR = -5		SNR = 0		SNR = 5	
	STOI	PESQ	STOI	PESQ	STOI	PESQ
Noisy	71.6	1.77	84.0	2.15	92.0	2.54
PL-LSTM	78.0	1.97	88.0	2.37	93.9	2.76
Autoencoder	81.4	2.38	90.3	2.71	95.1	2.99
$C_1 \neq C_2$	84.4	2.47	91.9	2.77	**96.0**	**3.07**
$C_1 = C_2$	84.0	2.42	91.7	2.75	95.9	3.05
$C_1 = C_2 = X$	**84.8**	**2.48**	**92.0**	**2.78**	**96.0**	3.05

Table 3: Cafeteria noise; STOI(%) and PESQ comparison

	SNR = -5		SNR = 0		SNR = 5	
	STOI	PESQ	STOI	PESQ	STOI	PESQ
Noisy	50.4	1.54	68.7	1.88	82.9	2.22
PL-LSTM	54.7	1.66	73.1	2.03	85.8	2.39
Autoencoder	57.2	1.69	75.1	2.11	86.8	2.49
$C_1 \neq C_2$	**57.7**	**1.76**	**75.4**	**2.14**	87.3	2.50
$C_1 = C_2$	57.2	1.75	75.1	**2.14**	**87.5**	**2.52**
$C_1 = C_2 = X$	57.5	1.74	75.2	2.13	87.4	2.50

Furthermore, we note that the best performing SEHAE model achieves greater improvements over the noisy speech on the River and Traffic noises than on the Cafeteria noise: an average of 9.3% STOI and 0.64 PESQ, and 8.4% STOI and 0.62 PESQ for River and Traffic noise, respectively, compared to 6.0% STOI and 0.24 PESQ for Cafeteria noise. Based on the examination of several samples, we hypothesize that this is due to the fact that Cafeteria noise contains other speech, making it harder for the model to differentiate between noise and target speech.

Finally, we show in Fig. 5 an utterance from the test set corrupted by Traffic noise at SNR = 5 dB, as well as the intermediate and final predictions from the decoders of the $C_1 \neq C_2$ SEHAE model. At each step the prediction becomes more refined and both STOI and PESQ increase.

6. CONCLUSION AND DISCUSSION

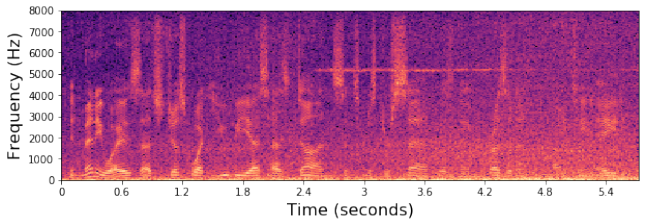
In this paper, we have proposed a novel speech enhancement architecture called SEHAE, consisting of encoder, funnel and decoder units. These were combined to form a hierarchical model, which was characterized by a scale-decomposed latent structure and the additive construction of the output starting from a canvas vector.

The experimental results demonstrated the efficacy of SEHAE as it yielded significant STOI and PESQ improvements over the PL-LSTM benchmark model for all noise types and at all SNRs.

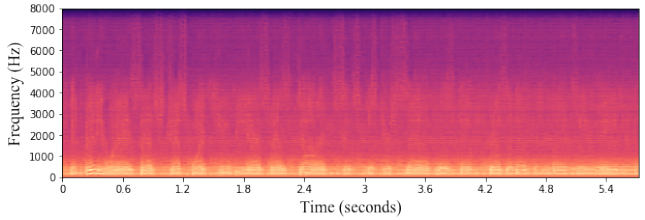
We expect that SEHAE can be improved upon. For instance, the hierarchical structure and intermediate predictions provide an opportunity for the application of progressive learning methods.

7. ACKNOWLEDGMENT

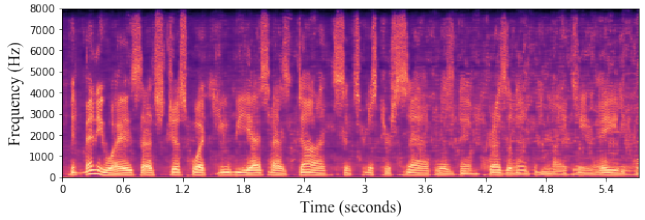
This work was supported in part by the National Key R&D Program of China under contract No. 2017YFB1002202. This work was also funded by Tencent, and will have been partially funded by the China Scholarship Council.



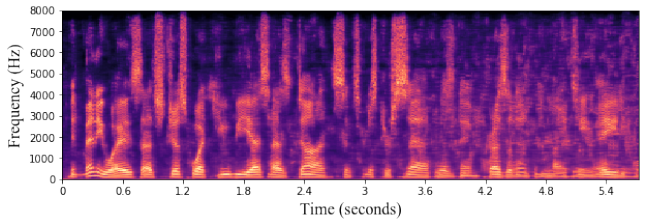
(a) Noisy (STOI = 75.9%, PESQ = 1.90)



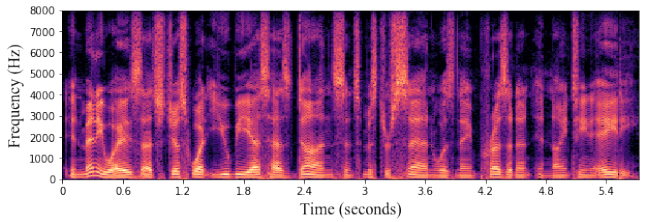
(b) First prediction (STOI = 54.4%, PESQ = 0.80)



(c) Second prediction (STOI = 81.2%, PESQ = 2.15)



(d) Final prediction (STOI = 84.9%, PESQ = 2.48)



(e) Clean

Fig. 5: Spectrograms of an utterance corrupted by Traffic noise at -5 dB SNR: (a) Noisy speech, (b) Decoder 1, (c) Decoder 2, (d) Decoder 3, (e) Clean speech

8. REFERENCES

[1] Randy Phyllis Granovetter, Michael J Sinclair, Zhengyou Zhang, and Zicheng Liu, "Method and apparatus for multi-sensory speech enhancement on a mobile device," Oct. 16 2007, US Patent 7,283,850.

[2] Li-Ping Yang and Qian-Jie Fu, "Spectral subtraction-based speech enhancement for cochlear implant patients in background noise," *The journal of the Acoustical Society of America*, vol. 117, no. 3, pp. 1001–1004, 2005.

[3] Raphael Koning, "Speech enhancement for cochlear implants," 2014.

[4] Tobias Goehring, Federico Bolner, Jessica JM Monaghan, Bas Van Dijk, Andrzej Zarowski, and Stefan Bleeck, "Speech enhancement based on neural networks improves speech intelligibility in noise for cochlear implant users," *Hearing research*, vol. 344, pp. 183–194, 2017.

[5] Pascal Scalart et al., "Speech enhancement based on a priori signal to noise estimation," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. IEEE, 1996, vol. 2, pp. 629–632.

[6] Steven Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[7] Yariv Ephraim and David Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[8] Yi Hu and Philipos C Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 4, pp. 334–341, 2003.

[9] Kevin W Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4029–4032.

[10] Shinichi Tamura and Alex Waibel, "Noise reduction using connectionist models," in *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 553–554.

[11] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.

[12] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

[13] Shahla Parveen and Phil Green, "Speech enhancement with missing data techniques using recurrent neural networks," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2004, vol. 1, pp. I–733.

[14] Martin Wöllmer, Zixing Zhang, Felix Weninger, Björn Schuller, and Gerhard Rigoll, "Feature enhancement by bidirectional lstm networks for conversational speech recognition in highly non-stationary noise," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6822–6826.

[15] Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John R Hershey, and Björn Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 91–99.

[16] Se Rim Park and Jinwon Lee, "A fully convolutional neural network for speech enhancement," *arXiv preprint arXiv:1609.07132*, 2016.

[17] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder.," in *Interspeech*, 2013, vol. 2013, pp. 436–440.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas 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.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] Prashanth Gurunath Shivakumar and Panayiotis G Georgiou, "Perception optimized deep denoising autoencoders for speech enhancement.," in *INTERSPEECH*, 2016, pp. 3743–3747.

[21] Tae Gyoon Kang, Jong Won Shin, and Nam Soo Kim, "Dnn-based monaural speech enhancement with temporal and spectral variations equalization," *Digital Signal Processing*, vol. 74, pp. 102–110, 2018.

[22] Santiago Pascual, Antonio Bonafonte, and Joan Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[23] Chris Donahue, Bo Li, and Rohit Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5024–5028.

[24] Juan Manuel Martín-Doñas, Angel Manuel Gomez, Jose A Gonzalez, and Antonio M Peinado, "A deep learning loss function based on the perceptual evaluation of the speech quality," *IEEE Signal processing letters*, vol. 25, no. 11, pp. 1680–1684, 2018.

[25] Szu-Wei Fu, Chien-Feng Liao, and Yu Tsao, "Learning with learned loss function: Speech enhancement with quality-net to improve perceptual evaluation of speech quality," *IEEE Signal Processing Letters*, vol. 27, pp. 26–30, 2019.

[26] Arash Vahdat and Jan Kautz, "Nvae: A deep hierarchical variational autoencoder," *arXiv preprint arXiv:2007.03898*, 2020.

[27] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[28] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[29] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[30] John Garofalo, David Graff, Doug Paul, and David Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[31] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent, "Demand: a collection of multi-channel recordings of acoustic noise in diverse environments," June 2013, Supported by Inria under the Associate Team Program VERSAMUS.

[32] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[33] Koen Oostermeijer, Qing Wang, and Jun Du, "Frequency gating: Improved convolutional neural networks for speech enhancement in the time-frequency domain," *Annual Summit and Conference 2020, Asia-Pacific Signal and Information Processing Association*, 2020.

[34] Jesper Jensen and Cees H Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2009–2022, 2016.

[35] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[36] ITU-T Recommendation, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[37] Tian Gao, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "Densely connected progressive learning for lstm-based speech enhancement," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5054–5058.