INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

MOSLight: A Lightweight Data-Efficient System for Non-Intrusive Speech Quality Assessment

Zitong Li[1], *Wei Li*[1,2*]

[1]School of Computer Science and Technology, Fudan University, Shanghai, China
[2]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
21210240239@m.fudan.edu.cn,weili-fudan@fudan.edu.cn

Abstract

Automatically predicting the mean opinion score (MOS) of a synthesized speech without the reference signal with deep learning systems has been studied extensively recently and shown great results. However, previous best systems are mostly based on self-supervised learned (SSL) models consisting of up to hundreds of millions of parameters making them unsuitable for mobile or embedded applications. In this paper, we propose MOSLight, a non-SSL-based lightweight yet powerful system for MOS prediction. We argue that 2D convolutions are inefficient for audio feature processing and not ideal for tasks where training data are scarce. To build MOSLight, we utilized depthwise separable dilated 1D convolutions and incorporated multi-task learning and non-strict frame-level score clipping. We conducted experiments on the Voice Conversion Challenge 2018 (VCC2018) and BVCC. Results show MOSLight achieves great effectiveness despite being a lightweight model trained with limited training data.

Index Terms: speech quality assessment, MOS prediction, speech synthesis

1. Introduction

The automatic quality assessment for synthesized speech signals has been a challenging problem. Commonly, the evaluation of speech synthesis systems is done via costly subjective listening tests. Full-reference objective assessment algorithms such as the perceptual evaluation of speech quality (PESQ) [1] and the short-time objective intelligibility (STOI) [2] are used as metrics in tasks where the reference signals are available (e.g. speech enhancement) [3, 4, 5]. However, reference signals are not always available for tasks such as text-to-speech (TTS) or voice conversion (VC). This problem is especially prevalent for deep neural speech codecs (e.g. Tacotron2 [6]), since their output signals do not usually align with the reference signals.

Recently, deep learning models for non-intrusive speech quality assessment have been extensively researched and shown great potential. MOSNet [7] is the first deep learning-based mean opinion score (MOS) prediction model for voice conversion to our knowledge. One main obstacle to building a DNN-based MOS prediction system is the scarcity of available data since most public-available datasets that can be used for MOS prediction are relatively small for DNN models. Researchers have found ways to use available data more efficiently. The mean-bias net (MBNet) [8] uses a separate bias net to predict the bias of a certain listener towards the mean score to utilize all individual ratings by each listener. Choi et al.[9] proposed multi-task learning (MTL) for MOS prediction. LDNet [10] utilizes the encoder-decoder structure to introduce shared parameters between the mean net and the bias net and adopts MobileNet [11, 12] architecture for its encoder design.

Another way to tackle this data scarcity is to incorporate self-supervised learned (SSL) models such as Wav2vec2.0 [13] or Hubert [14] in MOS prediction systems [15, 16]. Due to the generalization ability of SSL models, SSL-based systems have achieved great results in MOS prediction. In the VoiceMOS Challenge [17], top entries are mostly SSL-based systems. However, despite being powerful, the parameter count of an SSL model is enormous and a considerable amount of computing power and energy is needed to train or reference an SSL model. This makes SSL models unsuitable for situations where computing power is limited or low latency is required (e.g. mobile applications and embedded systems).

Convolution neural networks (CNN) are widely used in deep learning systems. One key factor for the effectiveness of CNNs is the translation-equivariance property of convolutions. CNNs are ideal for image processing, as images are translation-equivariant in nature, that is, when two similar patterns occur on different images or different locations in the same image, they often share similar semantics. CNNs are also widely used in audio deep learning applications including MOS prediction [7, 8, 9, 10]. Conventionally, the input spectrum is processed by 2D CNNs similarly to an image. However, in this paper, we argue that 2D CNNs are not efficient for spectrum processing. This is because 2D convolutions assume the input to be translation-equivariant along both axes, while spectrums are only translation-equivariant along the time axis, that is, a pattern shift on the frequency axis completely changes the sound. Due to the inefficiency caused by this incorrect inductive bias, models may be slow to converge or stuck in bad local optima. This may not be apparent when the training data are sufficient because deep learning models are good at generalizing when provided with enough training data. However, for circumstances where data are scarce, as is the case for most public MOS datasets including VCC2018[18] and BVCC [17], models based on 2D CNNs may underperform.

In this paper, we propose MOSLight, a non-SSL-based lightweight yet powerful model for non-intrusive speech assessment. MOSLight is built on depthwise separable dilated 1D convolution blocks and incorporated MTL and non-strict frame-level score clipping. We conducted experiments on VCC2018 and BVCC datasets and despite being a lightweight model trained with limited training data, MOSLight has achieved great results on both datasets.

*corresponding author
This work was supported by NSFC (62171138).

10.21437/Interspeech.2023-263

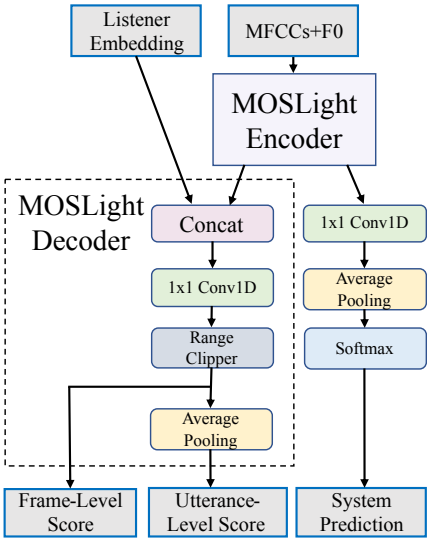

Figure 1: *Overview of the MOSLight framework*

2. Proposed Method

In this section, we describe the overall framework of MOSLight and explain some of the design choices we made. 80-dimensional Mel frequency cepstrum coefficients (MFCCs) and 1-dimensional fundamental frequencies (F0s) are used as the model input.

2.1. Listener-dependent modeling

Previous works have shown that instead of directly using MOS scores as the training objective, it is beneficial to leverage all individual scores by each listener (i.e. listener-dependent modeling or LD modeling) [8, 10]. In MOSLight specifically, LD modeling is done by utilizing an encoder-decoder structure. The encoder learns the listener-independent representations from the input audio features, while the decoder is injected with additional listener information and combines that with the learned representations to predict MOS scores, as shown in Figure 1.

The encoder consists of a 1×1 convolution [1] layer, several repeated convolution blocks, and another 1×1 convolution layer as shown in Table 1. A size factor m is used to scale the number of channels. The 1D convolution block is the core of MOSLight and will be discussed in detail in the latter section. The decoder consists of a 1×1 convolution layer, a range clipper to clip frame-level scores to a reasonable range and an average pooling layer to average frame-level scores to utterance-level scores. The decoder is simple and has very few trainable parameters (about 0.3% of total trainable parameters when $m = 3$). The reason for this design is that most of the time the listener preference only adds a bias to the MOS score[10], so the decoder should be made simple and the encoder more powerful to learn better listen-independent representations.

2.2. Non-strict range clipper

Frame-level scores can be highly unstable during training, which makes models slow to converge and performance degraded [7]. One way to stabilize frame-level scores is to apply a hyperbolic tangent function to hard clip all frame-level scores between 1 to 5 [19]. However, although a strict range clipper can stabilize training greatly, it also makes the model too conservative when making predictions, as the predicted MOS score is the average of all clipped frame-level scores. Therefore we utilized a non-strict range clipper in MOSLight:

$$s_t = (2 + \alpha) \tanh h_t + 3 \tag{1}$$

where s_t and h_t denote the predicted frame-level score and the input sequence at the timestep t, and α denotes the loose factor. The score is clipped between $1-\alpha$ and $5+\alpha$. When $\alpha = 0$, this is equivalent to a strict clipping. When α is higher, the clipping is looser, and vice versa. Non-strict clipping allows the model to be more expressive while still limiting the frame-level scores to a reasonable range.

Table 1: *MOSLight encoder framework with the model size factor m.*

Input	Layer	Output	Repeat
$81 \times t$	1×1 conv1d	$64m \times t$	1
$64m \times t$	conv1d block, dilation=1 conv1d block, dilation=2	$64m \times t$	3
$64m \times t$	conv1d block, dilation=1 conv1d block, dilation=2 conv1d block, dilation=4	$64m \times t$	4
$64m \times t$	1×1 conv1d instancenorm1d, GELU	$64m \times t$	1

2.3. 1D convolution block

To replace inefficient 2D CNNs, instead of introducing complicated mechanics such as adaptive kernels or attention mechanisms that can be difficult to design, implement and train, we simply used 1D CNNs as the basic building blocks for MOSLight. 1D convolutions assume translation-equivariance along only one axis, so it is natural for 1D CNNs to process spectrums and cepstrums. The input audio feature map is treated like a multi-channel 1D signal. For instance, an 81×100 dimension audio feature map is treated like an 81-channel signal with a length of 100 frames.

Two important features are incorporated in the MOSLight 1D convolution block design to further improve efficiency.

Dilated convolution [20] Dilated convolutions are convolutions whose kernels are widened by skipping a certain amount of steps when being applied to the input. Popularized by WaveNet [21], dilated convolutions are widely used in audio deep learning applications to enlarge the receptive field without additional parameters. It should be noted that prior works usually choose large dilation factor sequences because the input of their models is usually raw audio signals which require a large receptive field. The input for MOSLight is condensed audio features (MFCCs and F0) which is far shorter than raw signals depending on the hop length, so we chose smaller dilation factor sequences, as shown in Table 1.

Depthwise separable convolution [22] Depthwise separable convolutions factorize a standard convolution into a depthwise convolution, which applies a single filter to each channel, and a point-wise convolution, which is a 1×1 convolution used for combining the outputs. Depthwise separable convolutions

[1] Although a 1×1 convolution technically refers to a 2D convolution, it also refers to a 1D convolution with kernel size 1 by convention.

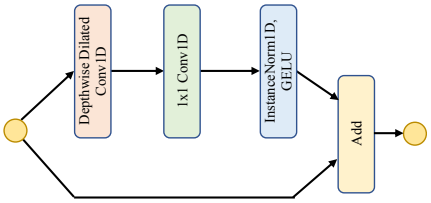

Figure 2: *Illustration of the structure for conv1d block used in the MOSLight encoder. All CNN layers have the same number of input and output channels. Kernel size 3 is set for all non-* 1×1 *convolutions.*

can greatly reduce the parameter count and are widely used in lightweight models such as MobileNet [11, 12, 23].

A 1D instance normalization [24] layer and a GELU nonlinearity [25] are used following the point-wise convolution. Finally, the input is added to the output to form a skip connection, as shown in Figure 2.

2.4. Multi-task learning

There is a great correlation between the quality of a speech and the system by which it was synthesized. Systems with high system-level MOS scores generally output high-score speeches and vice versa. Therefore, by incorporating system prediction (or called "spoofing type classification" in [9]) as an auxiliary task, MOS prediction models can more easily distinguish good systems from bad systems, thus improving MOS prediction results. Cross-entropy loss is used for this auxiliary task:

$$L_{cls} = -\frac{1}{U}\sum_{i=1}^{U}\sum_{j=1}^{N} \hat{p_{i,j}} * \log p_{i,j} \tag{2}$$

where U denotes the number of utterances in a mini-batch, N denotes the total number of systems, and $\hat{p_{i,j}}$ and $p_{i,j}$ denote the ground truth and the predicted probability of the jth system of the ith utterance.

2.5. Loss function

Both the utterance-level loss and the frame-level loss [7] is used for the training of MOSLight. Mean squared error (MSE) is used for the utterance-level loss. Formally, we can write the utterance-level loss function as:

$$L_{utt} = \frac{1}{U}\sum_{i=1}^{U}(\hat{Q}_i - Q_i)^2 \tag{3}$$

where $\hat{Q}_i$ and Q_i denote the ground truth and the predicted score of the ith utterance. Clipped MSE is used for the frame-level loss to prevent the model from overfitting the frame-level scores. We can write the frame-level loss function as:

$$L_{frame} = \frac{1}{U}\sum_{i=1}^{U}(\frac{1}{T_i}\sum_{t=1}^{T_i}\max\left[(\hat{Q}_i - q_{i,t})^2, \beta\right]) \tag{4}$$

where T_i denotes the length of the ith utterance, $q_{i,t}$ denotes the predicted frame-level score of the ith utterance at the timestep t and β denotes the threshold below which the gradient is set to zero.

Combining all three losses, the final loss function is:

$$L = L_{utt} + \lambda L_{frame} + \mu L_{cls} \tag{5}$$

where λ and μ are hyperparameters to balance the losses.

3. Experiments

3.1. Datasets

VCC2018 [18] The Voice Conversion Challenge 2018 is a large-scale challenge for voice conversion and the dataset contains 20580 samples generated by 38 systems (including real speech samples). All samples are in English. Each sample is rated by 4 listeners and there are 267 listeners in total. A random 13580/3000/4000 for train/val/test split is used. For VCC2018, different splits can cause the results to be vastly different, so we made sure all compared models are trained with the exact same split[2]. There are no unseen listeners or systems during validation and testing.

BVCC [17] This dataset is collected from past speech synthesis challenges and contains 7106 speech samples[3] generated by 187 systems (including real speech samples). All samples are in English. Each sample is rated by 8 listeners and there are 304 listeners in total. This dataset was pre-split 4974/1066/1066 for train/val/test under carefully-chosen rules by the VoiceMOS Challenge organizer. There are several unseen listeners and systems during validation and testing.

3.2. Data processing

All speech samples are downsampled to 16 kHz. A short-time Fourier transform (STFT) with a window length of 1024 samples and a hop length of 256 samples is conducted. A Mel-filterbank with 128 Mel filters is used to convert spectrums to Mel spectrums. MFCCs are extracted from Mel spectrums with a type-III discrete cosine transform (DCT). The first 80 dimensions of MFCCs are utilized while the rest are discarded. The pYIN [26] algorithm is used to extract F0s.

We utilized several data augmentation strategies for the BVCC training set: increasing the audio speed slightly, decreasing the audio speed slightly, and cropping the audio. Each strategy is utilized twice for each sample resulting in 6 times more data than the original. For audio speed altering, we have compared stretching and resampling and found that the pitch shift caused by resampling is less noticeable than the artifacts caused by stretching algorithms, so the resampling method is used. Data augmentations are not used for VCC2018.

3.3. Training setup

On VCC2018, a mean listener is utilized during training and inference like in [10]. On BVCC, the model is trained with 38 listener groups instead of individual listeners like in [27] and during training, 5 percent of the known listener groups are replaced with a "unknown" listener group, which is used for inference. For a classification task, the number of classes in the BVCC training set (175 systems) is too large for the number of training samples, so we grouped the systems in BVCC evenly into 6 groups according to their system-level scores and used system groups as the training target for MTL.

All models are implemented with the Pytorch framework and trained on an RTX3070 GPU. All hyperparameters are set the same for both datasets. We utilized the Adam optimizer with a learning rate of 10^{-4}. We did not utilize any schedulers. The batch size is set to 40. We used zero padding instead of repetitive padding [8] during training and implemented careful masking between individual Pytorch modules to prevent the padded values from affecting the normalization layers. The loose factor

[2]The split is from `https://github.com/unilight/LDNet`

[3]The "out-of-domain" (OOD) track of BVCC was not used.

Table 2: *Experimental results on the VCC2018 test set and the BVCC test set. Numbers from the first 4 rows are taken from [10]. "ML" and "All" stand for mean listener inference and all listener inference [10].*

Model	VCC2018						BVCC					
	Utterance level			System level			Utterance level			System level		
	MSE	LCC	SRCC	MSE	LCC	SRCC	MSE	LCC	SRCC	MSE	LCC	SRCC
MBNet [8]	0.955	0.658	0.630	0.665	0.978	0.957	0.669	0.757	0.765	0.522	0.854	0.860
MBNet-All [10]	0.615	0.656	0.627	0.154	0.980	0.966	0.492	0.758	0.765	0.271	0.856	0.860
LDNet-ML [10]	0.479	0.648	0.613	0.021	0.983	**0.979**	0.333	0.795	0.794	0.169	0.885	0.886
LDNet-All [10]	0.463	0.653	0.617	0.024	0.983	0.975	0.316	0.795	0.794	0.175	0.881	0.881
MOSLight-m=1	0.478	0.674	0.635	0.094	0.973	0.953	0.297	0.818	0.819	0.172	0.881	0.884
MOSLight-m=2	0.410	0.695	0.664	0.026	0.982	0.965	0.270	0.831	0.827	0.135	0.893	0.892
MOSLight-m=3	0.427	0.700	0.667	0.050	**0.986**	0.974	0.262	**0.844**	**0.842**	0.155	0.903	0.904
MOSLight-m=4	**0.400**	**0.703**	**0.672**	**0.014**	0.982	0.972	**0.247**	0.842	0.836	**0.122**	**0.909**	**0.905**

α in Equation 2 is set to 6. The threshold β in Equation 4 is set to 0.4. The loss balancing factor λ and μ in Equation 5 are set to 0.2 and 0.5 respectively. It should be noted that all hyperparameters are set empirically and we did not conduct a formal hyperparameter search. Three metrics, i.e. MSE, linear correlation coefficients (LCC), and Spearsman's rank correlation coefficients (SRCC) for both utterance-level and system-level are used. We chose the checkpoint with the best utterance-level SRCC on the validation set among 50 epochs.

3.4. Experimental results

Table 3: *Resource usage of LDNet and MOSLight. Runtime and million mult-adds are calculated by inputting 6 seconds of audio (for the hop length of 256 samples set by both systems, the input audio feature map is 375 frames in length). Runtime is averaged from 100 runs. All runs are done on a Ryzen CPU.*

Model	Million Params	Million Mult-Adds	Runtime (s)
LDNet-ML [10]	0.920	486	0.362
MOSLight-m=1	0.089	32.4	0.082
MOSLight-m=2	0.334	123	0.091
MOSLight-m=3	0.734	272	0.104
MOSLight-m=4	1.29	480	0.127

We have tested 4 configurations of the MOSLight, with the model size factor m ranging from 1 to 4, on VCC2018 and BVCC. We chose LDNet [10], a state-of-the-art lightweight MOS prediction system, as our baseline system. The experimental results are shown in Table 2. We have also conducted a system complexity comparison, as shown in Table 3. "All listener inference" mode LDNet requires running the decoder multiple times depending on the number of listeners, making it far slower, so it is not shown on the complexity comparison.

Table 2 shows the smallest "m=1" MOSLight model, with a tenth of the parameter and mult-add count of LDNet, can achieve comparable results to LDNet on VCC2018 and outperforms LDNet on BVCC. The "m=3" and "m=4" MOSLight models outperform LDNet significantly on VCC2018 utterance-level metrics and on all BVCC metrics despite similar or less resource usage.

It should be noted that on BVCC, the system-level SRCC between the training set and the test set is 0.905, so in theory, this should be near the top limit of systems that do not utilize any external data [28]. Table 2 shows both the "m=3" and "m=4" MOSLight models have already reached near this theoretical system-level SRCC limit, showing the effectiveness of the MOSLight framework.

3.5. Ablation study

Table 4: *Ablation study of MOSLight. "-X" and "+X" denote component X is removed from or added to the framework. Only the utterance-level SRCC and system-level SRCC on the BVCC test set are shown due to the space limitation.*

Model	Utterance SRCC	System SRCC
MOSLight-m=3	0.842	0.904
- clipper	0.829	0.895
- clipper + strict clipper	0.827	0.885
- MTL	0.827	0.882
- frame-level loss	0.800	0.865
- dilation	0.839	0.897
- dw separable conv + normal conv	0.819	0.885
- data augmentation	0.815	0.892

We have conducted several experiments on BVCC to show the effectiveness of each component of MOSLight, as shown in Table 4. It should be highlighted that by removing the frame-level loss, the results suffer the most in the listed experiments, indicating although score clipping is used, the frame-level loss is still necessary for stable training. It also should be pointed out that by replacing the depthwise separable convolutions, the parameter count of MOSLight increased by about 3 times but the model suffered from overfitting.

4. Conclusions

In this paper, we proposed an efficient MOS prediction model MOSLight, which is built on depthwise separable dilated 1D CNNs and incorporated MTL and non-strict range clipping. Experimental results show small configuration MOSLight achieved comparable results to LDNet baselines with far less resource usage while big configuration MOSLight outperformed LDNet. We also conducted an ablation study to prove the effectiveness of each module. For future work, we plan to explore more complicated structures than 1D CNNs and try to incorporate more domain knowledge into MOS prediction models.

5. References

[1] A. Rix, J. Beerends, M. Hollier, and A. 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.

[2] 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.

[3] Z. Huang, S. Watanabe, S.-w. Yang, P. García, and S. Khudanpur, "Investigating self-supervised learning for speech enhancement and separation," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6837–6841.

[4] C. Yu, S. wei Fu, T.-A. Hsieh, Y. Tsao, and M. Ravanelli, "OSSEM: one-shot speaker adaptive speech enhancement using meta learning," in *Proc. Interspeech 2022*, 2022, pp. 981–985.

[5] J.-H. Huang and C.-H. Wu, "Memory-Efficient Multi-Step Speech Enhancement with Neural ODE," in *Proc. Interspeech 2022*, 2022, pp. 961–965.

[6] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[7] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, "MOSNet: Deep Learning-Based Objective Assessment for Voice Conversion," in *Proc. Interspeech 2019*, 2019, pp. 1541–1545.

[8] Y. Leng, X. Tan, S. Zhao, F. Soong, X.-Y. Li, and T. Qin, "Mbnet: Mos prediction for synthesized speech with mean-bias network," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 391–395.

[9] Y. Choi, Y. Jung, and H. Kim, "Neural mos prediction for synthesized speech using multi-task learning with spoofing detection and spoofing type classification," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 462–469.

[10] W.-C. Huang, E. Cooper, J. Yamagishi, and T. Toda, "Ldnet: Unified listener dependent modeling in mos prediction for synthetic speech," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 896–900.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[12] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[14] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[15] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of mos prediction networks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8442–8446.

[16] R. E. Zezario, S.-W. Fu, F. Chen, C.-S. Fuh, H.-M. Wang, and Y. Tsao, "Deep learning-based non-intrusive multi-objective speech assessment model with cross-domain features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 54–70, 2022.

[17] W. C. Huang, E. Cooper, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, "The VoiceMOS Challenge 2022," in *Proc. Interspeech 2022*, 2022, pp. 4536–4540.

[18] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 195–202.

[19] W.-C. Tseng, C. yu Huang, W.-T. Kao, Y. Y. Lin, and H. yi Lee, "Utilizing Self-Supervised Representations for MOS Prediction," in *Proc. Interspeech 2021*, 2021, pp. 2781–2785.

[20] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[21] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[24] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[25] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[26] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2014, pp. 659–663.

[27] M. Chinen, J. Skoglund, C. K. A. Reddy, A. Ragano, and A. Hines, "Using Rater and System Metadata to Explain Variance in the VoiceMOS Challenge 2022 Dataset," in *Proc. Interspeech 2022*, 2022, pp. 4531–4535.

[28] A. Stan, "The ZevoMOS entry to VoiceMOS Challenge 2022," in *Proc. Interspeech 2022*, 2022, pp. 4516–4520.