End-to-End Integration of Speech Emotion Recognition and Voice Activity Detection with a Self-Supervised Model for Noise Robustness

Natsuo Yamashita*, Masaaki Yamamoto* and Yohei Kawaguchi*
* Research & Development Group, Hitachi, Ltd., Japan
E-mail: natsuo.yamashita.gh@hitachi.com

***Abstract*—Speech Emotion Recognition (SER) often operates on speech segments detected by a Voice Activity Detection (VAD) model in a cascade manner. However, VAD models may output flawed speech segments including noise-only or non-emotional segments, especially in noisy environments, resulting in performance degradation in subsequent SER models. To address this issue, we propose an end-to-end (E2E) method that integrates VAD and SER using Self-Supervised Learning (SSL) features. By jointly training both VAD and SER models with SSL features for a combined loss function, our approach enables the VAD module to capture emotional speech segments for SER, while making the SER model robust against the flawed VAD output. Experimental results on the IEMOCAP dataset demonstrate that our proposed method improves SER performance without optimizing the VAD threshold. Furthermore, we analyze our method under various conditions, such as different loss weights and noise levels.**

I. Introduction

Speech Emotion Recognition (SER) is the task of identifying and classifying emotional states expressed in spoken language [1]. It is an essential field within the expansive domain of affective computing and human-computer interaction, with a variety of real-world applications, including healthcare [2], customer service [3], and marketing [4].

Traditional SER methods [5]–[8] begin by extracting low-level descriptive features, such as prosodic characteristics and spectral features from speech signals, which are then fed into machine learning models. With recent advances in deep learning, especially the Transformer framework [9], a considerable number of studies [10]–[12] have focused on utilizing pre-trained self-supervised learning (SSL) models such as wav2vec 2.0 [13], HuBERT [14], and WavLM [15] as feature extractors. The SER models leveraging these SSL features (SSL-SER) have shown significantly improved performance in SER and other downstream tasks through fine-tuning [16]–[18].

In practical applications of SER, it is common practice to employ a Voice Activity Detection (VAD) model before feeding the audio into a SER model [19]. This pre-processing step aims to identify the speech segments where the speaker is actively talking, excluding other parts such as silence and noise-only segments. Recently, deep learning-based VAD models [20]–[22] have improved performance compared to traditional schemes based on statistics of speech [23]–[25]. However, these VAD models, while trained to distinguish speech from non-speech, are not optimized for SER performance. They may produce VAD outputs which are fragmented and lack emotional features, especially in noisy environments [26], which results in low SER performance. SER is susceptible to such flawed VAD outputs because it relies on detecting subtle variations and nuances in speech, such as tone, pitch, rhythm, intensity, and speed [27]. Even if speech segments are correctly identified, some parts may contain rich emotional parts while others may not, and the latter can lead to a decline in the accuracy of SER.

A previous study explored the combination of VAD and SER modules in a cascade manner [28]. However, the VAD and SER modules were not jointly trained to interact effectively. The VAD model was trained to distinguish speech from non-speech, but it was not optimized for capturing the emotional content crucial for SER. The SER model was also not trained to handle the fragmented output from the VAD. Additionally, the VAD threshold was arbitrarily determined and not optimized for SER, despite the additional challenges posed by finding the optimal VAD threshold.

In this paper, to address these problems, we propose a method that integrates VAD and SER modules using SSL features in an end-to-end (E2E) manner. SSL features are first input into the VAD module, and then the segmented SSL features are fed into the SER module. Both modules are jointly trained to optimize the combined VAD and SER loss. This approach allows the VAD module to be trained to include more emotional speech segments that are important for SER, while the SER module is trained to be robust against flawed segments from the VAD module. Furthermore, the end-to-end method optimizes the entire system, including both VAD and SER. This allows the system to implicitly and internally establish an optimal VAD output distribution under a fixed VAD threshold, eliminating the need to manually explore the best VAD threshold for SER. As our experimental results demonstrate, the proposed method improves SER performance on the IEMOCAP dataset.

II. Proposed approach

The overall architecture of the proposed approach is shown in Figure 1, which consists of the SSL, VAD, and SER modules. It is worth noting that the choice of network architecture

979-8-3315-7206-8/25/$31.00

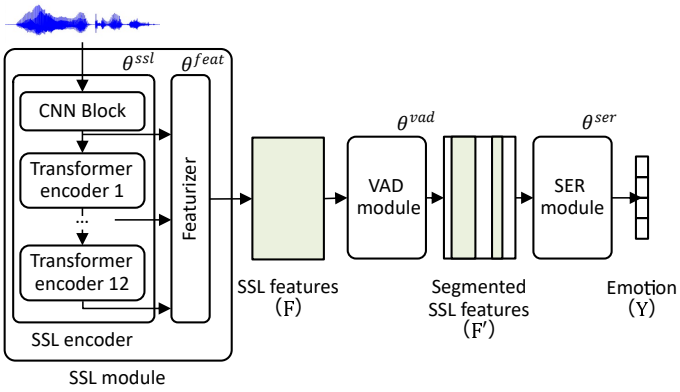

Fig. 1: Overview of the proposed end-to-end approach composed of SSL, VAD, and SER modules.

for each module is not restricted to any specific one.

A. SSL module

In this study, we employ SSL models as common feature extractors for both the following VAD and SER modules, due to their well-known generalizability and accessibility across various speech processing tasks [18]. We denote the feature extraction process of the SSL module as follows:

$$\mathbf{F} = \mathrm{SSL}(\mathbf{X};\, \theta^{\mathrm{ssl}},\, \theta^{\mathrm{feat}}), \tag{1}$$

where $\mathbf{X}$ is an input utterance, $\mathbf{F}$ is the SSL features, and θ^{ssl} and θ^{feat} represent the parameters of the SSL encoder and the Featurizer, respectively. Given an input waveform, the SSL encoder, consisting of a Convolutional Neural Network (CNN) block and 12 Transformer encoder blocks, extracts a frame sequence of 768-dimensional speech features with a frame shift of 20 ms.

Research [15], [18], [29] has shown that intermediate representations of such foundation models contain information useful for different tasks. Therefore, the Featurizer computes the weighted-sum of embeddings from the 13 hidden states of the SSL encoder.

B. VAD module

While conventional VAD methods [20], [21] often use spectrogram features such as log Mel-Filterbanks (Fbank) and Mel-Frequency Cepstral Coefficients (MFCC), very recent work [30] has shown that a VAD architecture based on wav2vec 2.0 outperforms previous works [21], [31]. In this study, we employ the VAD module using SSL features (SSL-VAD) and investigate the use of not only wav2vec 2.0 but also HuBERT and WavLM for our end-to-end approach. If we denote the VAD outputs indicating speech/non-speech as $\mathbf{S}$, the SSL features as $\mathbf{F}$, and the segmented SSL features as $\mathbf{F}'$, we can write the process of the VAD module:

$$\mathbf{S} = \mathrm{VAD}(\mathbf{F};\, \theta^{\mathrm{vad}}), \tag{2}$$

$$\mathbf{F}' = \mathbf{F} \odot \mathbf{S} \tag{3}$$

where θ^{vad} represents the parameters of the VAD module and $\odot$ is the Hadamard product operator.

The VAD module comprises four 1D convolutional layers with a hidden dimension of 256 and leaky ReLU activation, followed by a fully connected (FC) layer with softmax activation. It assigns a label of 1 or 0 to each 20ms frame from the SSL module using a fixed VAD threshold of 0.5. We do not optimize the VAD threshold in our end-to-end method, as the training process implicitly and internally forms an optimal VAD output distribution based on the fixed threshold. The segmented SSL features are obtained by applying a Hadamard product between the SSL features and the binary outputs of the VAD process.

C. SER module

We use the segmented SSL features from the VAD module as input for the SER module. The SER process can be written as:

$$\hat{\mathbf{Y}} = \mathrm{SER}(\mathbf{F}';\, \theta^{\mathrm{ser}}), \tag{4}$$

where $\hat{\mathbf{Y}}$ is a predicted emotion label and θ^{ser} represents the parameters of the SER module. Given the segmented 768-dimensional SSL features, the SER module first applies dimensionality reduction from 768 to 256, followed by average pooling. The representations are then processed through three 1D convolution layers and a FC layer with ReLU activation. A subsequent self-attention pooling layer [32] aggregates features along the time axis, which are then fed into a FC layer with ReLU activation and a FC layer with softmax activation for emotion classification. If the VAD module is not included in the pipeline, the above process is applied to the raw SSL features $\mathbf{F}$ instead of the segmented SSL features $\mathbf{F}'$.

D. End-to-end training

In this study, we propose an end-to-end approach that jointly optimizes the VAD and SER modules for both a combined VAD and SER loss using the SSL features. Our entire end-to-end approach is described as follows:

$$\begin{aligned}\hat{\mathbf{Y}} = \mathrm{SER}(&\mathrm{SSL}(\mathrm{X};\, \theta^{\mathrm{ssl}},\, \theta^{\mathrm{feat}}) \\ &\odot \mathrm{VAD}(\,\mathrm{SSL}(\mathrm{X};\theta^{\mathrm{ssl}},\, \theta^{\mathrm{feat}});\, \theta^{\mathrm{vad}});\, \theta^{\mathrm{ser}}),\end{aligned} \tag{5}$$

To enable effective feedback between the VAD and SER modules from the start of end-to-end training, we initialize each parameter as follows. $\hat{\theta}^{\mathrm{ssl}}$ is initialized using publicly available pre-trained models, while $\hat{\theta}^{\mathrm{feat}}$, $\hat{\theta}^{\mathrm{vad}}$, and $\hat{\theta}^{\mathrm{ser}}$ are initialized through pre-training on their respective tasks, namely SSL-VAD and SSL-SER. In our experiments, $\hat{\theta}^{\mathrm{ssl}}$ is kept frozen to reduce computational costs. Subsequently, θ^{feat}, θ^{vad}, and θ^{ser} are fine-tuned jointly using the combined loss function for VAD and SER to achieve better performance, as described below:

$$\min \mathcal{L} = \alpha\, \mathcal{L}_{\mathrm{VAD}} + (1-\alpha)\, \mathcal{L}_{\mathrm{SER}}, \tag{6}$$

where α is a manually set loss weight.

TABLE I: SER performance (%UA, %WA) for the baselines (Conditions 1 and 2) and proposed approaches (Conditions 3-6) *on extended utterances under noisy environments*.

				wav2vec 2.0		HuBERT		WavLM+	
ID	Condition	Init Param.	Update Param.	UA	WA	UA	WA	UA	WA
1	SSL-SER	$\hat{\theta}^{\text{ssl}}$, $\hat{\theta}^{\text{feat}}$, $\hat{\theta}^{\text{ser}}$		46.0	49.9	42.6	47.4	43.8	48.2
2	MarbleNet-SSL-SER	$\hat{\theta}^{\text{ssl}}$, $\hat{\theta}^{\text{feat}}$, $\hat{\theta}^{\text{ser}}$		40.3	39.6	43.4	48.0	42.8	42.1
3	SSL-VAD-SER	$\hat{\theta}^{\text{ssl}}$, $\hat{\theta}^{\text{feat}}$, $\hat{\theta}^{\text{vad}}$, $\hat{\theta}^{\text{ser}}$		49.6	48.3	46.4	50.8	45.8	43.9
4	E2E SSL-FT.VAD-SER	$\hat{\theta}^{\text{ssl}}$, $\hat{\theta}^{\text{feat}}$, $\hat{\theta}^{\text{vad}}$, $\hat{\theta}^{\text{ser}}$	θ^{feat}, θ^{vad}	49.7	48.0	47.9	51.0	46.7	44.1
5	E2E SSL-VAD-FT.SER	$\hat{\theta}^{\text{ssl}}$, $\hat{\theta}^{\text{feat}}$, $\hat{\theta}^{\text{vad}}$, $\hat{\theta}^{\text{ser}}$	θ^{feat}, θ^{ser}	50.2	53.1	49.4	52.6	49.5	53.5
6	E2E SSL-FT.VAD-FT.SER	$\hat{\theta}^{\text{ssl}}$, $\hat{\theta}^{\text{feat}}$, $\hat{\theta}^{\text{vad}}$, $\hat{\theta}^{\text{ser}}$	θ^{feat}, θ^{vad}, θ^{ser}	**50.6**	**53.7**	**52.2**	**54.9**	**51.6**	**55.2**

III. Experimental setup

A. Model configurations

We used pre-trained SSL models including wav2vec 2.0 BASE[1], HuBERT BASE[2], and WavLM BASE+[3], which are publicly available. For simplicity, the term "BASE" will be omitted hereafter. Each model has approximately 95 million parameters. Wav2vec 2.0 and HuBERT were trained with the concatenation of the *train-clean-100*, *train-clean-360*, and *train-other-500* subsets from the LibriSpeech dataset [33]. WavLM+ was trained with the Libri-Light [34], GigaSpeech [35], and VoxPopuli datasets [36]. Based on these SSL models, the SSL-VAD and SSL-SER were pre-trained and initialized using the Adam optimizer [37] with a fixed learning rate of 1×10^{-4} and a batch size of 8, without data augmentation, for at most 10 epochs. In the proposed approach, the VAD and SER modules were fine-tuned using the combined loss with the same optimizer algorithm and learning rate, for at most 8 epochs. We used $\alpha = 0.2$ for loss weight, which we observed resulted in the best SER performance, as shown in Section IV-B.

For the VAD component of a baseline, we implemented MarbleNet [20], which is one of the commonly used deep learning-based VAD models, as a pre-processing VAD step for SSL-SER. MarbleNet utilized MFCC features with 64 mel-filter bank dimensions, a $25\,\text{ms}$ window size and $10\,\text{ms}$ overlap. MarbleNet and the SSL-SER were individually trained and the segmented speeches detected by MarbleNet were fed into the SSL-SER. In general, the optimal VAD threshold depends on the dataset and noise environment [24]. Therefore, we optimized the thresholds for both the MarbleNet-SSL-SER and the SSL-VAD-SER to compare with the proposed end-to-end method. Each VAD threshold was searched to achieve the best SER performance using a step size of 0.1 from 0 to 1, and the thresholds used in the experiments are summarized in Table II.

[1]https://huggingface.co/facebook/wav2vec2-base-960h
[2]https://huggingface.co/facebook/hubert-base-ls960
[3]https://huggingface.co/microsoft/wavlm-base-plus

TABLE II: VAD threshold parameters.

	wav2vec 2.0	HuBERT	WavLM+
MarbleNet-SSL-SER	0.3	0.4	0.4
SSL-VAD-SER	0.3	0.2	0.2

B. Dataset

We used the IEMOCAP dataset [38], which contains approximately 12 hours of English speech including 5 dyadic conversational sessions between two actors. There are in total 151 dialogues, including 10,039 utterances. To simulate noisy environment, we additively contaminated the utterances with environmental noise, randomly selected from the 37 recordings annotated as background noise in the MUSAN corpus [39] at Signal-to-Noise Ratio (SNR) levels of $\{10, 5, 0, -5, -10\}$ dB for training and evaluation. The IEMOCAP provides the timestamps for each utterance in a dialogue, as well as word-level alignments for each utterance. The utterances contain silence at the beginning, between words, and at the end. The VAD module was trained based on these speech/non-speech alignments, with $40\,\%$ of the segments being non-speech. For SER, we merged emotion class "excited" with "happy" and used audio annotated with one of four labels, which are happy, sad, neutral, and angry.

In the evaluation of each utterance, to simulate a real-world scenario that includes non-speech segments at the beginning and the end, and to evaluate the entire system which includes both VAD and SER, we used timestamps spanning from the end of the previous utterance to the start of the next utterance. These extended utterances averaged $11.7\,\text{s}$, while the original utterances averaged $4.5\,\text{s}$. Speaker exclusive leave-one-session-out five-fold cross validation (CV) is performed and the average performance is reported. Additionally, we made sure that utterances from the same dialogue were entirely in the training set or entirely in the validation set.

IV. Results and Analysis

In this section, we present the evaluation results and analysis on the IEMOCAP dataset. We computed Unweighted Accuracy (UA) and Weighted Accuracy (WA) on the test sets, as

shown in Table I. UA is the average recall across all categories and WA is the total number of correct predictions divided by the total number of samples.

A. Main Results

Conditions 1 and *2* serve as baselines, while *Conditions 3-6* represent our proposed approaches with different fine-tuning strategies. The results of *Condition 1* show much lower accuracy compared to previous studies [10], [14], [15]. Note that this is because our evaluation was conducted on extended utterances including non-speech parts at the beginning and the end under noisy environments, whereas previous studies were conducted on non-extended utterances under clean environments. The results of *Condition 2* show that while SER performance with HuBERT was improved a little, the performance with wav2vec2 and WavLM+ was degraded when MarbleNet was used, compared to *Condition 1*. This indicates that individually combining a VAD model with the SSL-SER cannot always lead to significant improvement in SER performance, especially in noisy environments.

In *Conditions 3-6*, we explored four different fine-tuning strategies for our approach. In *Condition 3*, which combines individually pre-trained SSL-VAD and SSL-SER without fine-tuning, the performance improved across all SSL models compared to *Condition 2*, indicating the effectiveness of employing SSL-VAD for SER. In *Condition 4*, θ^{feat} and θ^{vad} were fine-tuned with the other parameters frozen. The performance of *Condition 4*, on most SSL models, shows improvement compared to *Condition 3*. This suggests that a VAD module optimized solely for speech/non-speech detection may not always be optimal for SER, highlighting the importance of VAD's role in SER. In *Condition 5*, θ^{feat} and θ^{ser} were fine-tuned, with the other parameters frozen. *Condition 5* also improved SER performance compared to *Condition 3* for any SSL model. This indicates the importance of fine-tuning on segmented SSL features $\mathbf{F}'$, rather than fine-tuning on unsegmented SSL features $\mathbf{F}$. In *Condition 6*, θ^{feat}, θ^{vad}, and θ^{ser} were fine-tuned, while $\hat{\theta}^{\text{ssl}}$ remained frozen. We observe that *Condition 6* further improved SER performance on all the SSL models. This indicates that our end-to-end approach successfully fine-tuned the VAD and SER modules jointly for improved SER performance, without requiring any optimization of the VAD threshold. For example, comparing *Condition 2* and *6* on WavLM+, the UA increased from $43.8\,\%$ to $51.6\,\%$ and the WA increased from $48.2\,\%$ to $55.2\,\%$.

B. Impact of different loss weights

Table III presents the WA and UA scores obtained using different loss weights for VAD and SER during end-to-end training in *Condition 6* with WavLM+. We explored loss weights ranging from 0 to 1, with increments of 0.2. The results show that the SER performance was highest when $\alpha = 0.2$, outperforming cases where $\alpha = 0$ or $\alpha = 1$. This suggests that the SER task plays a more significant role, while the VAD task also supports the training process by providing complementary benefits.

TABLE III: SER performance (%UA, %WA) with different loss weights in Confition 6 with WavLM+.

α	0	0.2	0.4	0.6	0.8	1
UA	50.5	**51.6**	51.3	50.3	49.4	39.4
WA	54.0	**55.2**	54.6	54.2	53.1	40.0

C. Performance across different SNR levels

The SER performance under different SNR conditions is shown in Table IV. The results show that SSL-VAD-SER consistently outperforms SSL-SER by incorporating VAD across all SNR levels. Furthermore, our proposed end-to-end approach in *Condition 6* achieved the highest performance across all SNR levels. The improvement was more significant at higher SNR levels, compared to noisy ones. This suggests that the VAD module in our proposed method is more effective at detecting emotional segments at higher SNR levels, while still minimizing the impact of noise-only segments in more noisy environments.

TABLE IV: UA with WavLM+ *using extended utterances at different SNR levels*.

	10 dB	5 dB	0 dB	-5 dB	-10 dB
SSL-SER	47.9	46.0	43.6	41.6	39.5
SSL-VAD-SER	48.8	46.7	44.2	42.1	39.8
E2E SSL-FT.VAD-FT.SER	**57.0**	**54.2**	**49.9**	**46.6**	**43.6**

D. Effect of probabilistic vs. binary VAD outputs

We also compared the use of probabilistic VAD outputs (without a threshold) and binary VAD outputs (with a fixed threshold) in *Condition 6*. The results in Table V show that binary VAD outputs achieve higher accuracy than probabilistic VAD outputs for all SSL models. We infer that binary output clearly segments frames, reduces the impact of noise-only and non-emotional segments, and improves performance compared to probabilistic outputs.

TABLE V: SER performance (%UA, %WA) with probabilistic vs. binary VAD outputs in Condition 6.

	wav2vec 2.0		HuBERT		WavLM+	
	UA	WA	UA	WA	UA	WA
probabilistic	36.3	39.1	44.3	47.7	48.0	52.1
binary	**50.6**	**53.7**	**52.2**	**54.9**	**51.6**	**55.2**

E. Analysis of weights of the Featurizer

We investigated the weights of the Featurizer in SSL-VAD, SSL-SER, and E2E SSL-FT.VAD-FT.SER, which were trained for VAD, SER, and both VAD and SER, respectively. Figure 2 shows the weights of different layers of the Featurizer on HuBERT. The results indicate that Layers 0-4 are effective for

VAD, while Layers 8-10 are more relevant for SER, consistent with previous findings [15], [18], [40]. We observe that the Featurizer in our end-to-end method successfully emphasizes the features of Layers 8 and 9 for SER, as well as of Layers 0-4 for VAD, enabling the VAD module to take emotional features into account.

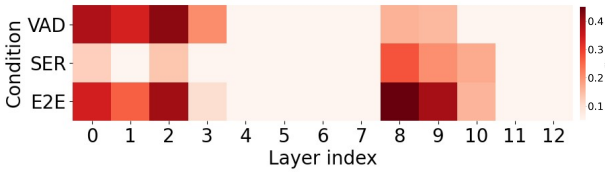

Fig. 2: Visualization of the Featurizer weights on HuBERT. The y-axis represents the weights of each Featurizer for SSL-VAD (top), SSL-SER (middle), and E2E SSL-FT.VAD-FT.SER (bottom), while the x-axis represents different layers. Layer 0 corresponds to the input of the first Transformer layer.

F. Initialization with random vs. pretrained parameters

One limitation of our proposed method is the requirement for pretraining, requiring additional effort to prepare the individual VAD and SER modules. As illustrated in Figure 3, the training curves indicate that starting from random initialization fails to converge to an optimal point. We assume that VAD and SER are interdependent tasks and it takes more time to learn these relationships when starting from random parameters.

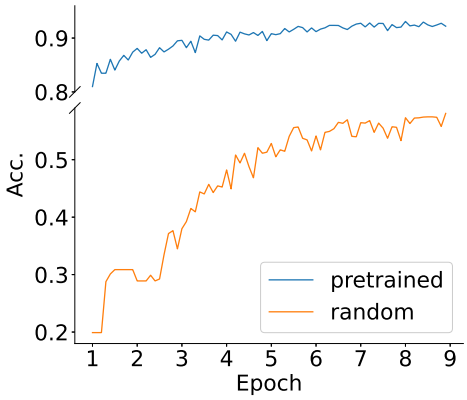

Fig. 3: Accuracies during fine-tuining Condition 6 from random or pretrained parameters on the development set.

V. Conclusions

In this study, we presented a method that integrates VAD and SER modules using SSL features in an end-to-end manner. Our approach allowed the VAD module to capture emotional speech segments for SER, while making the SER module robust against flawed segments from the VAD module. The experimental results on the IEMOCAP dataset showed that our proposed method improved SER performance. Future work includes evaluation of our proposed approach on other datasets.

References

[1] V. Dimitrios and K. Constantine, "Emotional speech recognition: Resources, features, and methods," *Speech communication*, vol. 48, no. 9, pp. 1162–1181, 2006.

[2] N. A. Vaidyam, H. Wisniewski, J. D. Halamka, M. S. Kashavan, and J. B. Torous, "Chatbots and conversational agents in mental health: A review of the psychiatric landscape," *The Canadian Journal of Psychiatry*, vol. 64, no. 7, pp. 456–464, 2019.

[3] V. Petrushin, "Emotion in speech: Recognition and application to call centers," in *Proc. ANNIE*, vol. 710, 1999, p. 22.

[4] R. P. Bagozzi, M. Gopinath, and P. U. Nyer, "The role of emotions in marketing," *Journal of the academy of marketing science*, vol. 27, no. 2, pp. 184–206, 1999.

[5] T. L. Nwe, S. W. Foo, and L. C. De Silva, "Speech emotion recognition using hidden markov models," *Speech communication*, vol. 41, no. 4, pp. 603–623, 2003.

[6] A. Milton, S. S. Roy, and S. T. Selvi, "SVM scheme for speech emotion recognition using MFCC feature," *International Journal of Computer Applications*, vol. 69, no. 9, pp. 34–39, 2013.

[7] W. Lim, D. Jang, and T. Lee, "Speech emotion recognition using convolutional and recurrent neural networks," in *Proc. APSIPA*, 2016, pp. 1–4.

[8] Y. Xie, R. Liang, Z. Liang, C. Huang, C. Zou, and B. Schuller, "Speech emotion classification using attention-based LSTM," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1675–1685, 2019.

[9] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, vol. 30, 2017.

[10] L. Pepino, P. Riera, and L. Ferrer, "Emotion recognition from speech using wav2vec 2.0 embeddings," in *Proc. Interspeech*, 2021, pp. 3400–3404.

[11] M. Macary, M. Tahon, Y. Estève, and A. Rousseau, "On the use of self-supervised pre-trained acoustic and linguistic features for continuous speech emotion recognition," in *Proc. SLT*, 2021, pp. 373–380.

[12] H. Zou, Y. Si, C. Chen, D. Rajan, and E. S. Chng, "Speech emotion recognition with co-attention based multi-level acoustic information," in *Proc. ICASSP*, 2022, pp. 7367–7371.

[13] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for selfsupervised learning of speech representations," in *Proc. NeurIPS*, vol. 33, 2020, pp. 12 449–12 460.

[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 TASLP*, vol. 29, pp. 3451–3460, 2021.

[15] S. Chen, C. Wang, Z. Chen, *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech

processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
[16] Y. Wang, A. Boumadane, and A. Heba, *A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding*, 2021.
[17] L.-W. Chen and A. Rudnicky, "Exploring wav2vec 2.0 fine tuning for improved speech emotion recognition," in *Proc. ICASSP*, 2023, pp. 1–5.
[18] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, *et al.*, "SUPERB: Speech processing universal performance benchmark," in *Proc. NeurIPS*, vol. 33, 2020, pp. 12 449–12 460.
[19] M. Sharma, S. Joshi, T. Chatterjee, and R. Hamid, "A comprehensive empirical review of modern voice activity detection approaches for movies and TV shows," *Neurocomputing*, vol. 494, pp. 116–131, 2022.
[20] F. Jia, S. Majumdar, and B. Ginsburg, "Marblenet: Deep 1d time-channel separable convolutional neural network for voice activity detection," in *Proc. ICASSP*, 2021, pp. 6818–6822.
[21] N. Wilkinson and T. Niesler, "A hybrid CNN-BiLSTM voice activity detector," in *Proc. ICASSP*, 2021, pp. 6803–6807.
[22] Q. Yang, Q. Liu, N. Li, M. Ge, Z. Song, and H. Li, "SVAD: A robust, low-power, and light-weight voice activity detection with spiking neural networks," in *Proc. ICASSP*, 2024, pp. 221–225.
[23] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE signal processing letters*, vol. 6, no. 1, pp. 1–3, 1999.
[24] J.-H. Chang, N. S. Kim, and S. K. Mitra, "Voice activity detection based on multiple statistical models," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 1965–1976, 2006.
[25] *WebRTC VAD*. [Online]. Available: https://webrtc.org/.
[26] S. Braun and I. Tashev, "On training targets for noise-robust voice activity detection," in *Proc. EUSIPCO*, 2021, pp. 421–425.
[27] Y. Huang, J. Xiao, K. Tian, A. Wu, and G. Zhang, "Research on robustness of emotion recognition under environmental noise conditions," *IEEE Access*, vol. 7, pp. 142 009–142 021, 2019.
[28] M. F. Alghifari, T. S. Gunawan, M. A. b. W. Nordin, S. A. A. Qadri, M. Kartiwi, and Z. Janin, "On the use of voice activity detection in speech emotion recognition," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 3607–3611, 2019.
[29] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *Proc. ASRU*, 2021, pp. 914–921.
[30] B. Karan, J. J. van Vüren, F. de Wet, and T. Niesler, "A transformer-based voice activity detector," in *Proc. Interspeech*, 2024, pp. 3819–3823.
[31] *Silero VAD: Pre-trained enterprise-grade voice activity detector (VAD) number detector and language classifier*, 2021. [Online]. Available: https://github.com/snakers4/silero-vad.
[32] P. Safari, M. India, and J. Hernando, "Self-attention encoding and pooling for speaker recognition," in *Proc. Interspeech*, 2020, pp. 941–945.
[33] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.
[34] J. Kahn, M. Riviere, W. Zheng, *et al.*, "Libri-Light: A benchmark for ASR with limited or no supervision," in *Proc. ICASSP*, 2020, pp. 7669–7673.
[35] G. Chen, S. Chai, G. Wang, *et al.*, "GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio," in *Proc. Interspeech*, 2021, pp. 3670–3674.
[36] C. Wang, M. Riviere, A. Lee, *et al.*, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *Proc. ACL-IJCNLP*, 2021, pp. 993–1003.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
[38] C. Busso, M. Bulut, C.-C. Lee, *et al.*, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.
[39] D. Snyder, G. Chen, and D. Povey, *MUSAN: A music, speech, and noise corpus*, arXiv:1510.08484, 2015.
[40] W. u, C. Zhang, and P. C. Woodland, "Integrating emotion recognition with speech recognition and speaker diarisation for conversations," in *Proc. Interspeech*, 2023, pp. 941–945.