MetricAug: A Distortion Metric-Lead Augmentation Strategy for Training Noise-Robust Speech Emotion Recognizer

Ya-Tse Wu, Chi-Chun Lee

Department of Electrical Engineering, National Tsing Hua University, Taiwan

crowpeter@gapp.nthu.edu.tw, cclee@ee.nthu.edu.tw

Abstract

Noise-robust speech emotion recognition (SER) systems are important in real world applications. Conventionally, noise robustness is achieved by training on a noise-augmented dataset. In this work, instead of pre-defining noise SNRs to augment the clean set, we propose an *augment-while-train* strategy while referencing speech distortion metric. This strategy (*MetricAug*) constructs an augmented set per each training epoch by assessing the effect of different distortion levels have on degrading the SER performances. That is, we augment more of those noisy data that degrade the SER performance the most dynamically at each learning epoch. We evaluate our framework on two databases, MSP-Podcast and MELD. Our framework shows consistent robustness against varying levels and even unseen noise types. Further analysis reveals that by choosing STOI as the metric of noise distortion, it leads the construction of augmented sets better than metrics of PESQ and fwSNRseg.

Index Terms: speech emotion recognition, speech distortion metrics, noise robustness

1. Introduction

In recent years, speech emotion recognition (SER) technology is being deployed on real-world systems. A key component in making a real-world SER successful is its ability to maintain high performances when being exposed to varying noisy conditions. Achieving a noise-robust SER can be cast as a problem of building a denoising system as a pre-processor [1, 2], or more elegantly, learning a noise-robust SER model [3, 4, 5]. Data augmentation is an effective mechanism to achieve model robustness. By creating noisy copies of speech to augment the originally clean training set, a noise-robust model can be derived by training on this noise-augmented set. In fact, this strategy has demonstrated its superiority not just for SER but also for automatic speech recognition [6, 7].

Several works have used the augmentation strategy to achieve noise-robust SER. For example, Wilf et al. created noisy condition as another domain, and through the use of domain adaptation, one can derive a noise-robust SER [3]; Leem et al. explored noise robustness of low-level descriptors and applied these descriptors for training [4]; Pappagari et al. expanded the original dataset by mixing noise and emotion data and further evaluated in noisy scenarios [5]. Most of these augmentation methods follow an *augment-then-train* strategy, i.e., by predefining fixed levels (SNRs) of noisy samples to be generated, one would then train a model on this *static* augmented set.

Instead of *augment-then-train*, a better approach would be *augment-while-train*, i.e., on-the-fly constructing an augmented set during the training process. In fact, past studies have explored strategy of *augment-while-train* for learning feature robustness in automatic speech recognition. Park et al. demonstrated improved robustness of spectrogram after random time masking, frequency masking, and time warping during the training [8]. Hu et al. proposed a sample-adaptive policy for augmentation, which augment samples based on the training loss to derive an adaptive augmented hyper-parameter [9].

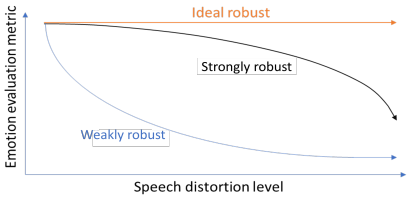

Figure 1: *A noise-robust learning for SER.*

Our concept of achieving SER robustness is shown in Fig. 1. The orange line is the ideal robust line, i.e., as the distortion level increases, the performance stays the same. The augmentation process is to force the "robust curve" to approximate the ideal robust line by adding adequate amount of noisy speech during training. Specifically, our idea is at every learning epoch, we first assess our current model's degradation in performance as a function on the severity of the speech distortion levels, then we dynamically adjust the amount of augmented noisy data accordingly (adding more of those levels of noisy samples that our current model performs the worst at). While a higher distortion level results in a larger degradation of SER performance, the relationship between the types of distortion metrics used and SER performance is not well explored. We use three different speech distortion metrics to lead this epoch-wise augmentation strategy: short-time objective intelligibility (STOI) [10], perceptual evaluation of speech quality (PESQ) [11] and frequency-weighted segmental SNR (fwSNRseg) [12]. In this work, we propose a dynamic augmentation process, *MetricAug*, embedded in the model learning process.

We term this as a distortion metric-lead augmentation strategy. We first make use of the MUSAN noise dataset to create a sufficiently diverse noisy *Superset*. Then, we derive a sampling-weight for each distortion level as measured by the chosen metric at every epoch according to the resulting degradation of performances. This sampling weight dictates the amount and the kind of noisy data from the *Superset* to be added to the epoch-wise augmented set. We evaluate our method on MSP-Podcast and MELD datasets and demonstrate improved SER robustness against varying levels of noisy conditions and even in a setting of unseen noise type. Finally, we reveal that the distortion metric of STOI *leads* the augmentation process better than the other two metrics examined. To the best of our knowledge, we are the first study using the discriminative adaptive augmentation strategy for noise-robust SER.

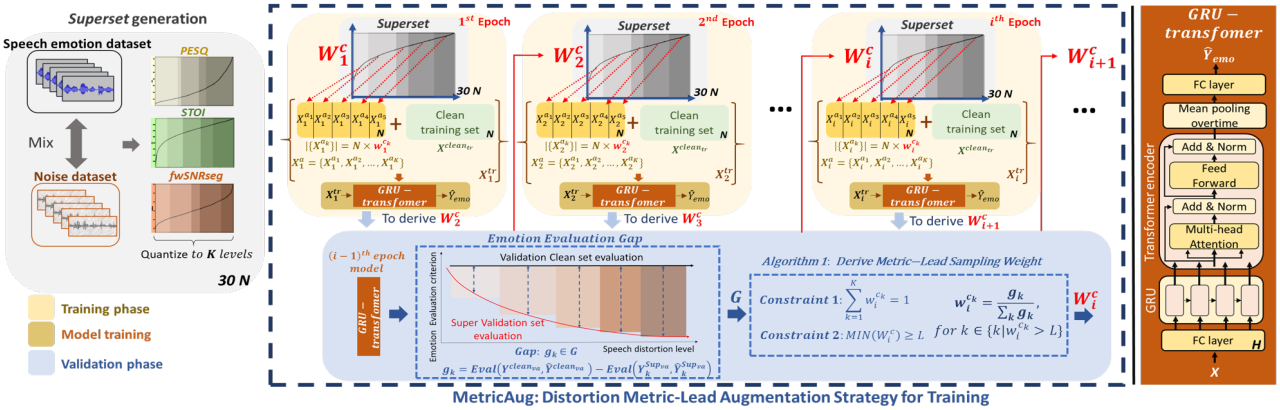

Figure 2: *Illustration of MetricAug: An epoch-wise distortion metric-lead noise augmentation.*

Table 1: *A summary of 4-classes emotion distribution in two databases.*

	MSP			MELD		
	Train	Dev	Test	Train	Dev	Test
Angry	1839	621	758	1109	153	345
Happy/Joy	8564	1320	4401	1743	163	402
Neutral	16518	2529	6962	4710	470	1256
Sad	1681	302	666	683	111	208
Total	28602	4772	12787	8245	897	2211

2. Research Methodology

2.1. Dataset

2.1.1. Speech emotion dataset: MSP-Podcast

We used MSP-Podcast corpus version 1.8 as one of the speech emotion dataset, which contains real podcast recordings (16kHz, 1Ch) segmented in utterances [13]. There are 73,042 utterances for a total of 113 hours in the dataset, including 1,285 speakers. Each utterance is annotated with nine primary categorical emotions by a minimum of five annotators. In this work, we target four classes data for our speech emotion classification: neutral, angry, sad and happy. We follow the split which authors provided, the emotion distributions are shown in Table 1.

2.1.2. Speech emotion dataset: MELD

Multimodal EmotionLines Dataset (MELD) is a multimodal database collected from the TV series "Friends" [14], which contains 13,000 utterances and 407 speakers in total. Each utterance is annotated with seven categorical emotions. We use ffmpeg [15] to extract the speech data from video and further down-sample them into (16kHz, 1Ch) by using Librosa [16]. We gathered from four emotion classes (anger, sadness, joy and neutral) and followed the original split setting which authors provided, the emotion distributions are shown in Table 1.

2.1.3. Noise dataset: MUSAN corpus

A Music, Speech, and Noise (MUSAN) corpus contains about 60 hours of speech, 6 hours of noise and 42 hours of music [17]. This is a common dataset used for studies of noise augmented robust SER learning [18, 5]. In this work, we use the noise and music part to mix with the speech emotion data. There are 929 noise audios and 660 music audios in total.

2.1.4. Noise dataset: ESC-50

Dataset for environmental sound classification (ESC-50) contains 2,000 noise audios in five different categories [19]. This is also used by other SER data augmentation framework [3]. In this work, we mix ESC-50 with the testing data of both speech emotion dataset, and we treat these data as unseen noisy data to evaluate our framework.

2.2. Speech Emotion Classification Model

In this work, we use a similar emotion classification model as in the most recent state of the art [20]. It includes 512 dimensions vq-wav2vec [21] and a GRU-transformer. In our network, the vq-wav2vec is first reduced to H dimensions by a linear fully connected layer, then the hidden embedding is passed to the stack of one GRU layer and one transformer encoder layer with two heads. Average pooling over time is applied before it passes to the final fully connected layer with a softmax function for four emotion classification. Except the last layer, all hidden dimensions of layers are consistent. We decide H by applying the greedy search with a double policy from 8 to 128. The $N_{parameter}$ ranges are from 5k to 265k.

2.3. Distortion Metric-Lead Augmentation Strategy

Our proposed method *MetricAug* is shown in Fig. 2. The strategy centers on generating epoch-wise augmented data, which are sampled from a large *Superset*. In this following, we will describe the construction of our *Superset* and detail our augmentation strategy.

2.3.1. Superset: A Diverse and Large Noisy Set

Superset is constructed by mixing the MUSAN audio samples and the original training set and validation set, denoted as $X^{Sup_{tr}}$ and $X^{Sup_{va}}$, respectively. To ensure that the *Superset* includes adequate diversity, the emotional speech samples are mixed with varying levels of noise and music from the MUSAN. We target the *Superset* to have samples with SNR ranges from 0dB to 30dB at an interval step of 2dB. This results in a $30N$ total number of samples after mixing, where N denotes the total number of samples in the original speech emotion dataset.

2.3.2. Distortion Metrics

There are three distortion metrics used in this work to *lead* the generation of the epoch-wise augmented set.

- **PESQ**: It is used for measuring speech quality after distortion that is designed for modeling the subjective tests for the mean opinion score (MOS) in the range of -0.5 and 4.5.
- **STOI**: It is a correlation coefficient based method, which measures the intelligibility between distorted speech and clean speech. It correlates to the words correct rate in human listening tests.
- **fwSNRseg**: It measures the SNR in several frequency bands in short window frames, which can be seen as an advanced version of the original time domain SNR.

We first obtain a distortion metric distribution on the entire *Superset*. We then further quantize it into K levels using either *Uniform* splitting (every discrete distortion level has the same amount of data samples) or *GMM* clustering (a distributional method for quantization). To preserve at least N data points for each level during clustering of speech distortion metrics, we set $K = 5$ on all of our approaches.

2.3.3. MetricAug: Augmentation Strategy

The core idea of our augmentation strategy is that given N samples to be retrieved from the *Superset* for every epoch i, we generate a dynamic weight vector for i^{th} epoch $W_i^c \in \mathbb{R}^{1\times K}$ to decide the amount of augmented samples retrieved in the k^{th} distortion level as measured by a distortion metric from the *Superset*. This augmented set is denoted as $X_i^{a_k} \subset X^{Sup_{tr}}$, where $a = \{\text{PESQ, STOI, fwSNRseg}\}$.

$$|\{X_i^{a_k}\}| = N \times w_i^{c_k} \tag{1}$$

$$X_i^a = \{X_i^{a_1}, X_i^{a_2}, ..., X_i^{a_K}\} \tag{2}$$

where $|\{X_i^{a_k}\}|$ is the cardinality of set $\{X_i^{a_k}\}$, $k \in [1, K]$, $w_i^{c_k} \in W_i^c$ denote the i^{th} epoch sampling weights of the k^{th} distortion level. The final training set X_i^{tr} of i^{th} epoch is denoted as follow:

$$X_i^{tr} = \{X_i^a, X^{clean_{tr}}\} \tag{3}$$

where $X^{clean_{tr}}$ is the original clean training set. The process of deriving W_i^c for each epoch is detailed in the next section.

2.3.4. Sampling Weight on Distortion Levels

$w_i^{c_k}$ is a weight constant that decides the amount of samples out of N to retrieved from *Superset*'s k^{th} distortion level at i^{th} epoch. We first define two constraints for $w_i^{c_k}$:

$$\begin{aligned} \Sigma_{k=1}^K w_i^{c_k} &= 1 \\ min(W_i^c) &\geq L \end{aligned} \tag{4}$$

where L means the lower bound of sampling weight. By setting $L = 0.05$, this guarantees a minimum of 5% of N will be sampled from the k^{th} level. To reinforce the training efficiency, the total size of the sample augmented is N. To derive W^c, we first define g_k as the "emotion evaluation gap" of the k^{th} level for a specific distortion metric. This gap is computed by the model performance difference between the clean validation set and each of k^{th} levels of noisy validation set $X^{Sup_{va}}$ under specific distortion metric distribution in the $(i-1)^{th}$ epoch:

$$\begin{aligned} g_k = Eval(Y^{clean_{va}}, \hat{Y}^{clean_{va}}) - \\ Eval(Y_k^{Sup_{va}}, \hat{Y}_k^{Sup_{va}}) \end{aligned} \tag{5}$$

where $Eval(.)$ gives model performance measured in weighted-f1. This gap is computed in the validation stage after each epoch, i.e., g_k is also epoch dependent. The process of obtaining W_i^c is shown in the Algorithm 1, it uses the normalized g_k to adjust the sampling weight $w_i^{c_k}$ for the next epoch. If the g_k is greater, the sampling weight of that specific level is larger. To achieve both constraints listed in equation (4), we first assign weights as L for those normalized g_k are less than L, and exclude them in the normalization term by subtracting them from 1. In the final step, we distribute the rest to those normalized $g_k > L$ (line 6 of Algorithm 1). We set all $w_i^{c_k} = 1/K = 0.2$ in the first epoch since there is no g_k initially. We provide all of our source code on a github repository[1].

[1] https://github.com/crowpeter/MetricAug

Algorithm 1 Deriving MetricAug Sampling Weight

Input: $(i-1)^{th}$ epoch emotion evaluation gaps G, $g_k \in G$
Output: i^{th} epoch sampling weights on distortion levels W_i^c, $w_i^{c_k} \in W_i^c$

1: Set $L = 0.05$
2: Initial $w_i^{c_k} = \frac{g_k}{|\Sigma_{k=1}^K g_k|}$, for $k \in [1, K]$
3: **while** $\Sigma_{k=1}^K w_i^{c_k} \neq 1$ **or** $MIN(W_i^c) < L$ **do**
4: Assign $w_i^{c_k} \leftarrow L$ if $w_i^{c_k} < L$, for $k \in [1, K]$
5: Assign $M' \leftarrow 1 - |\{w_i^{c_k} | w_i^{c_k} = L\}| \times L$
6: Update $w_i^{c_k} \leftarrow \frac{g_k}{\Sigma_k g_k} \times M'$, for $k \in \{k | w_i^{c_k} > L\}$
7: **end while**
8: **return** W_i^c

3. Experimental Setup and Results

We ran our experiment on the four class emotion classification task on both MSP-Podcast and MELD. All experiments are implemented by pytorch 1.12.1 [22], the parameters of each model are initialized by the pytorch default setting and takes 12 to 24 hours to train on a Nvidia GeForce RTX 3090 GPU. Due to the limitation of GPU memory, the batch size is set to 32. Early stopping is applied with ten continuous patience. Models are trained using a single cross entropy loss and updated by an Adam optimizer with learning rate 1e-3. The criterion we use to evaluate model performance is weighted-f1 score (WF1).

3.1. Testing Set

- **Clean:** The original testing set without any additive noise.
- **Fixed SNR at 0dB, 5dB, and 10dB:** Adding the noise and music from the MUSAN corpus at the specified SNR levels. The purpose of this testing set is for comparing the performance with the model which augments the fixed SNRs.
- **Unseen Noise:** Adding noise from ESC-50 at random SNRs in the range of [0dB, 30dB]. This set is for testing model robustness when exposed to unseen noise at unknown SNR.

3.2. Augmentation Strategy Comparison

- **None:** Using the original training set to train the model.
- **Fixed SNRs:** Using the fixed SNRs of 0dB, 5dB and 10dB augmented noisy data and the clean set to train the model, which is the most common approach for training noise-robust SER [23, 24, 3].
- **CopyPaste [5]:** A recent SOTA that shows robustness against noise for SER. We re-implement their data augmentation method on the training set. There are two different schemes for augmenting emotion samples, 1) concatenate the neural and emotional sample as a new emotion sample, and 2) concatenate two emotion samples as a new emotion sample. The utterances used to concatenate do not exceed 4 seconds to prevent overfitting. We use both schemes to generate 80 percent more samples from the original data set, and we further take this expanded set to mix with noise (at SNR of 0dB 5dB and 10dB for both noise and music). The final total augmented training data size is (1+0.8)×7×N=12.6N.
- **MetricAug (Proposed Method):** Using distortion metric-lead augmentation strategy to train the model. We test three metric distributions (STOI, PESQ, fwSNRseg) with two distortion levels quantizations method (*Uniform*, *GMM*).

Table 2: *The weighted-f1 score (WF1) for all augmentation methods in each testing set.*

MSP-Podcast	None	Fixed SNR	CopyPaste	MetricAug in Uniform Level			MetricAug in GMM Level		
Metric Distribution				STOI	PESQ	fwSNRseg	STOI	PESQ	fwSNRseg
Training Set Size	N	7N	12.6N	2N	2N	2N	2N	2N	2N
Clean	59.56	60.72	60.08	61.24	61.52	59.85	***61.70**	60.99	60.91
SNR 10dB	54.88	59.61	59.21	58.46	59.52	57.82	***60.16**	59.57	58.68
SNR 5dB	51.58	58.14	57.67	56.98	58.31	56.18	***58.85**	58.12	57.04
SNR 0dB	47.79	55.64	55.33	54.01	55.07	53.47	***56.95**	55.28	54.36
Unseen Noise	55.03	59.78	58.79	58.40	59.70	58.18	***60.09**	59.41	58.99
MELD	None	Fixed SNR	CopyPaste	MetricAug in Uniform Level			MetricAug in GMM Level		
Metric Distribution				STOI	PESQ	fwSNRseg	STOI	PESQ	fwSNRseg
Training Set Size	N	7N	12.6N	2N	2N	2N	2N	2N	2N
Clean	50.48	50.59	51.86	51.91	51.54	51.17	***52.85**	51.23	50.29
SNR 10dB	49.58	50.40	49.89	51.00	50.75	50.34	***51.66**	50.00	50.28
SNR 5dB	47.96	50.05	49.50	49.52	50.01	49.36	***50.53**	49.03	49.12
SNR 0dB	46.20	***49.95**	47.57	47.42	48.07	48.75	49.23	48.27	48.34
Unseen Noise	49.56	50.87	50.21	50.92	51.46	50.66	***51.98**	50.47	50.32

Table 3: *Normalized WF1 ($\frac{WF1_{clean}-WF1_{noisy}}{WF1_{clean}}$) in each GMM clustered leveling result C_k of metrics distribution on MSP-Podcast for non-augmented model.*

clean model	STOI	PESQ	fwSNRseg
GMM C_1	0.023	0.029	0.010
GMM C_2	0.025	0.020	0.017
GMM C_3	0.047	0.039	0.044
GMM C_4	0.112	0.094	0.087
GMM C_5	0.193	0.164	0.151
total	***0.401**	0.346	0.309

3.3. Result and Analysis

Table 2 shows a summary of 4-class emotion classification results. We will discuss various augmentation methods and performance of each testing set.

3.3.1. Performance Comparison on Clean Testing Set

The STOI-GMM augmentation method obtains the best result on both MSP-Podcast and MELD (the hidden dimension is $H = 16$). In comparison to the superior outcomes achieved through CopyPaste and Fixed SNR, the imrprovements obtained for MSP-Podcast and MELD are 0.98 and 2.26, respectively. When compared to the second-best result obtained from the proposed method, the improvements for MSP-Podcast and MELD are 0.18 and 0.94, respectively.

3.3.2. Performance Comparison on Fixed SNRs Condition

Under fixed SNRs scenario, we observe the STOI-GMM perform better than other metric-lead methods on both MSP-Podcast and MELD. In MSP-Podcast, compared to the Fixed SNR augmentation method, STOI-GMM augmentation improves 0.55, 0.71 and 1.31 on the SNR 10dB, 5dB and 0dB, respectively. Compared to the CopyPaste in the same condition, the improvement reached 0.95, 1.18 and 1.62.

In MELD, STOI-GMM augmentation on the SNR 10dB and 5dB condition leads by 1.26 and 0.48 compared to Fixed SNR augmentation, respectively. STOI-GMM also exceeds CopyPaste by 1.77, 1.03 and 1.66 in SNRs of 10dB, 5dB and 0dB, respectively. While both Fixed SNR and CopyPaste have matched noisy condition, i.e., training and testing on the same SNRs (10dB, 5dB and 0dB), STOI-GMM still outperforms almost all of the results of CopyPaste and Fixed SNR (except for SNR 0dB in MELD). Also note that, our training set is only of size $2N$, which is the smallest among all methods.

3.3.3. Performance Comparison on Unseen Noise

While testing on unseen noise from ESC-50, the STOI-GMM still obtained the best results. Compared to the next best result from the other methods, it improves by 0.31 and 0.52 on MSP-Podcast and MELD, respectively. This result shows that our augmentation strategy is even robust against unseen noise types.

3.3.4. Distortion Metric Analysis

We further carry out an analysis to understand how an emotion classification performance changes as a function on different kinds of distortion metric used to lead the augmentation process. We first evaluate the non-augmented (clean-trained) model performance on clean set and noisy set. Then, we compute the gap of WF1 between each GMM-quantized distortion level on the set and the clean testing set result $\frac{WF1_{clean}-WF1_{noisy}}{WF1_{clean}}$. Table 3 shows the sum of gaps for three different metrics STOI, PESQ and fwSNRseg (0.401, 0.346 and 0.309). By referencing the performance of MSP-Podcast in Table 2 using STOI-GMM, PESQ-GMM and fwSNRseg-GMM (61.70, 60.99 and 60.91), we see that if a metric has gap values that is larger (indicating that a more severe model performance degradation due to increased noisy conditions), using that metric to lead the augmentation process is better. According to these results, the degradation in SER performances seems to correlate more with the distortion severity level as measured by STOI (measure of intelligibility) than by PESQ (measure of quality).

4. Conclusions and Future Work

In this work, we carry out a novel strategy of epoch-wise automation for training a noise robust SER model. We compare the use of several speech distortion metrics to assess distortion levels and further demonstrate the use of STOI is better than two others (PESQ, fwSNRweg) metrics when leading the noise augmentation. Our method consistently outperforms other augmentation methods with smaller training sets and even can handle unseen noise conditions. A limitation of our work is that it focuses on the additive noise condition currently. An immediate future work is to extend the distortion types (reverberation, background speech), and to improve the current method to work without the need of iterative model performance evaluation.

5. References

[1] H. Zhou, J. Du, Y.-H. Tu, and C.-H. Lee, "Using speech enhancement preprocessing for speech emotion recognition in realistic noisy conditions," *Proc. Interspeech 2020*, pp. 4098–4102, 2020.

[2] Y. Nam and C. Lee, "Cascaded convolutional neural network architecture for speech emotion recognition in noisy conditions," *Sensors*, vol. 21, no. 13, p. 4399, 2021.

[3] A. Wilf and E. M. Provost, "Towards noise robust speech emotion recognition using dynamic layer customization," in *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2021, pp. 1–8.

[4] S.-G. Leem, D. Fulford, J.-P. Onnela, D. Gard, and C. Busso, "Not all features are equal: Selection of robust features for speech emotion recognition in noisy environments," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6447–6451.

[5] R. Pappagari, J. Villalba, P. Żelasko, L. Moro-Velazquez, and N. Dehak, "Copypaste: An augmentation method for speech emotion recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6324–6328.

[6] Z. Tu, J. Deadman, N. Ma, and J. Barker, "Auditory-based data augmentation for end-to-end automatic speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7447–7451.

[7] L. Meng, J. Xu, X. Tan, J. Wang, T. Qin, and B. Xu, "Mixspeech: Data augmentation for low-resource automatic speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7008–7012.

[8] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[9] T.-Y. Hu, A. Shrivastava, J.-H. R. Chang, H. Koppula, S. Braun, K. Hwang, O. Kalinli, and O. Tuzel, "Sapaugment: Learning a sample adaptive policy for data augmentation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4040–4044.

[10] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. 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.

[11] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. 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. IEEE, 2001, pp. 749–752.

[12] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on audio, speech, and language processing*, vol. 16, no. 1, pp. 229–238, 2007.

[13] R. Lotfian and C. Busso, "Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings," *IEEE Transactions on Affective Computing*, vol. 10, no. 4, pp. 471–483, 2017.

[14] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "Meld: A multimodal multi-party dataset for emotion recognition in conversations," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 527–536.

[15] S. Tomar, "Converting video formats with ffmpeg," *Linux journal*, vol. 2006, no. 146, p. 10, 2006.

[16] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015, pp. 18–25.

[17] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[18] C.-W. Huang and S. S. Narayanan, "Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition," in *2017 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2017, pp. 583–588.

[19] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1015–1018.

[20] J.-L. Li and C.-C. Lee, "An enroll-to-verify approach for cross-task unseen emotion class recognition," *IEEE Transactions on Affective Computing*, no. 01, pp. 1–13, 2022.

[21] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *International Conference on Learning Representations*, 2019.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[23] A. Triantafyllopoulos, G. Keren, J. Wagner, I. Steiner, and B. W. Schuller, "Towards robust speech emotion recognition using deep residual networks for speech enhancement," *Proc. Interspeech 2019*, pp. 1691–1695, 2019.

[24] U. Tiwari, M. Soni, R. Chakraborty, A. Panda, and S. K. Kopparapu, "Multi-conditioning and data augmentation using generative noise model for speech emotion recognition in noisy conditions," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7194–7198.