INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Supervised Contrastive Learning with Nearest Neighbor Search for Speech Emotion Recognition

Xuechen Wang[1], *Shiwan Zhao*[*], *Yong Qin*[1,†]

[1]Nankai University, Tianjin, China

shirleywxc0103@mail.nankai.edu.cn, zhaosw@gmail.com, qinyong@nankai.edu.cn

Abstract

Speech Emotion Recognition (SER) is a challenging task due to limited data and blurred boundaries of certain emotions. In this paper, we present a comprehensive approach to improve the SER performance throughout the model lifecycle, including pre-training, fine-tuning, and inference stages. To address the data scarcity issue, we utilize a pre-trained model, wav2vec2.0. During fine-tuning, we propose a novel loss function that combines cross-entropy loss with supervised contrastive learning loss to improve the model's discriminative ability. This approach increases the inter-class distances and decreases the intra-class distances, mitigating the issue of blurred boundaries. Finally, to leverage the improved distances, we propose an interpolation method at the inference stage that combines the model prediction with the output from a k-nearest neighbors model. Our experiments on IEMOCAP demonstrate that our proposed methods outperform current state-of-the-art results.

Index Terms: speech emotion recognition, supervised contrastive learning, k-nearest neighbors

1. Introduction

The recognition of emotions has emerged as a significant aspect in the field of human-computer interaction. Speech, being a rich source of emotional cues conveyed through various attributes such as pitch, frequency, speed, and accent, has received considerable attention in this regard. With the development of artificial intelligence technologies, Speech Emotion Recognition (SER) has been applied in a broad range of domains such as online education, human customer service, psychological health, and entertainment. Nevertheless, the recognition of emotional categories from speech utterances presents a daunting challenge due to their abstract and complex nature, coupled with limited data availability.

Recent advances in deep learning have made it the primary method for SER. Specifically, Badshah et al. [1] proposed a framework based on Convolutional Neural Networks (CNNs) to predict emotions. Satt et al. [2] used spectrograms extracted from speech as the model input directly, obtaining better performance and limiting the latency. Mirsamadi et al. [3] used Recurrent Neural Networks (RNNs) to learn frame-level features. Kumawat et al. [4] applied Time Delay Neural Network (TDNN) to capture the temporal information and provide an utterance level prediction. Liu et al. [5] proposed a method with attention mechanism to capture both temporal and frequency domain context information from input spectrograms. Despite these developments, challenges remain in this field that need to be addressed.

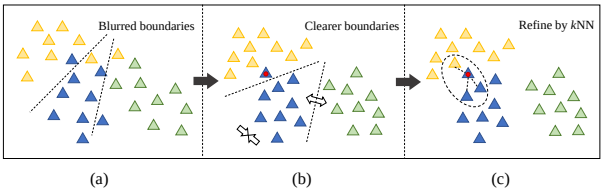


Figure 1: *The overall approach throughout the entire lifecycle of SER. (a): Feature embeddings after pre-training. (b): Feature embeddings after fine-tuning with the help of supervised contrastive learning. (c): Refine predictions by interpolating model outputs with a kNN model in the inference stage.*

Compared to other related fields, the datasets in SER are often limited in size, making it challenging to train large-scale and robust models exclusively using SER datasets. Moreover, there is no generic pre-trained model available that can be directly applied to SER. Meanwhile, self-supervised pre-trained models such as Wav2vec2.0 [6] and Hubert [7] have achieved good performances in Automatic Speech Recognition (ASR). They have been trained with a large amount of speech data and could construct better feature embeddings for utterances. However, their application in other related fields, such as SER, is still in its initial stages. In this paper, we adopt a transfer learning method [8] to address these challenges. Specifically, we leverage the self-supervised model wav2vec2.0 as a pre-trained model to obtain more accurate speech representations. We fine-tune the pre-trained model on a specific SER task to make the representations more suitable for the downstream task.

In the fine-tuning stage, previous works [9, 10] used cross-entropy loss to guide the model to solve multi-class classification problems. However, some researches [11, 12] found that cross-entropy loss may result in poor generalization performance and lack robustness to noisy labels, especially when the training data is limited. In addition, due to the blurred boundaries of certain emotions, general models may struggle to distinguish these emotions, for example, resulting in the misclassification of excited emotions as neutral emotions [13]. To solve the above problems, Li et al. [14] and Lian et al. [15] introduced contrastive learning to pull samples from the same classes closer and push samples from different classes farther apart. However, such naive contrastive learning approaches could not make full use of data labels for limited data. Alaparth et al. [16] applied Supervised Contrastive Learning (SCL) [17] loss to fine-tune the original model, which required two training stages to obtain the final predictions. In this paper, we propose a new loss function that combines cross-entropy loss and SCL loss in a weighted manner. The new objective function allows the model to learn better feature representations with increased inter-class

* Independent researcher.
†Corresponding author.

10.21437/Interspeech.2023-842

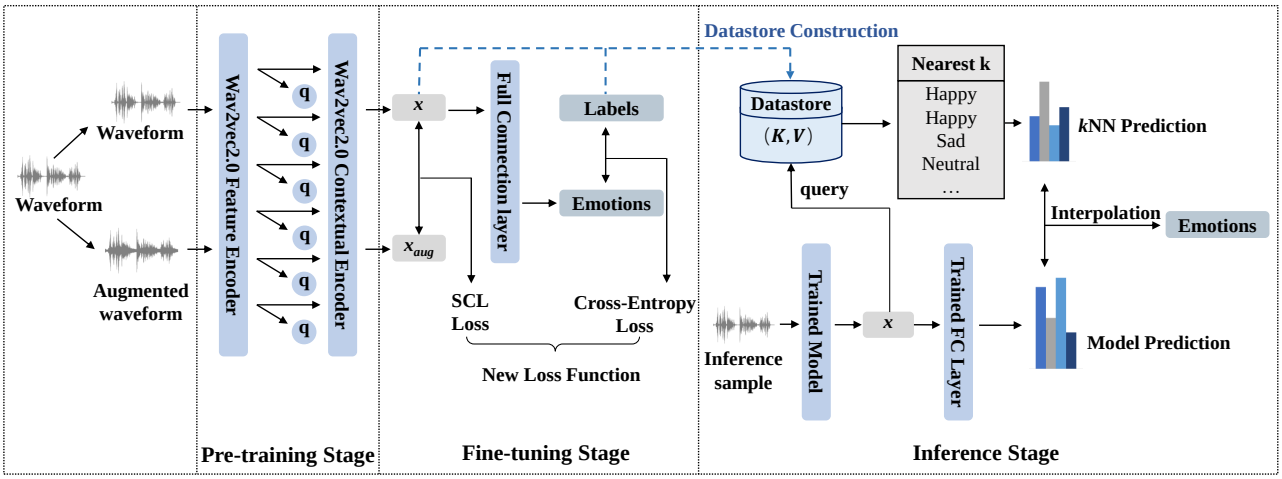


Figure 2: *The overall framework of our proposed method to improve the performance of SER throughout all stages, including the stages of pre-training, fine-tuning, and inference.*

distances and decreased intra-class distances more efficiently. This helps to address the problem of blurred boundaries and, as a result, improve the classification performance of the model.

Following the training with supervised contrastive learning, the intra-class feature representations become more compact while the inter-class feature representations become more separated. However, previous works did not fully utilize these enhanced feature representations that contain better distance information. In this study, we propose to apply the k-nearest neighbors (kNN) algorithm in the inference stage to fully leverage the feature distances. As illustrated in figure 1, kNN can refine the model's predictions by interpolating between the model's output and the output of kNN. In the embedding space generated by the fine-tuned model, kNN results can be obtained by retrieving the nearest training samples. Retrieving augmented methods have been demonstrated to be effective in various Natural Language Processing (NLP) works [18, 19]. By explicitly utilizing this information during inference, we can improve the model's classification ability without additional training. To the best of our knowledge, our work is the first to leverage the retrieval mechanism to enhance the SER performance.

In summary, our main contributions are as follows:

- We propose a comprehensive framework to improve SER performance throughout the model lifecycle, including pre-training, fine-tuning, and inference stages.
- We combine cross-entropy loss and SCL loss in the fine-tuning stage, improving the performance with better feature representations.
- We employ a kNN model to further enhance the model performance in the inference stage by leveraging the improved sample distance from SCL.
- Our experiments on the IEMOCAP dataset demonstrate that our proposed methods outperform current state-of-the-art results, achieving 74.13% and 75.13% on WA and UA, respectively.

2. Proposed Methods

In this section, we discuss the overall architecture of our proposed method. As shown in figure 2, our approach covers the entire lifecycle of a SER pipeline, including pre-training, fine-tuning, and inference stages. We first review the pre-trained model wav2vec2.0. Then, we introduce our new learning objective combining cross-entropy loss and SCL loss in the fine-tuning stage. Finally, we present our proposed method for interpolating model outputs using the kNN algorithm during the inference stage.

2.1. Pre-training Stage

Wav2vec2.0 is a transformer-based model. It contains three modules, feature encoder module, contextual encoder module, and quantization module. The feature encoder module contains several convolution layers. The contextual encoder module is based on transformer architecture. The quantization module is used to discretize the output of the feature encoder to a finite set of speech representations via product quantization. During training, wav2vec2.0 relies on the method of self-supervised learning, using a large amount of unlabeled speech data.

In this paper, to overcome the data scarcity issue in SER and obtain more accurate representations of utterances, we utilize wav2vec2.0 as the pre-trained model to extract features in the pre-training stage.

2.2. Fine-tuning Stage

During fine-tuning, we propose a new loss function which combines cross-entropy loss and SCL loss in a weighted manner. In SCL, multiple samples belonging to the same class can be treated as positive samples to each other. To compute the SCL loss, we first define the concepts of *positives* and *negatives* for supervised contrastive learning in our task.

For a set of N instances $\{x_k, y_k\}$, $k = 1, \ldots, N$, x_k denotes the feature embedding of a waveform and y_k denotes its label, represented by one-hot code. A training batch consists of $2N$ instances, $\{x_l, y_l\}$, $l = 1, \ldots, 2N$, where x_{2t} $(t = 1, \ldots, N)$ denotes the original waveform x_k and x_{2t-1} denotes the augmented version of x_k $(k = 1, \ldots, N)$. The label of an augmented waveform is the same as the original waveform, which could be expressed as $y_{2t} = y_{2t-1} = y_k$. Instances that have the same label y are called *positives*, and instances that have different labels are called *negatives*. The loss function of supervised contrastive learning is as follows [20]:

$$\mathcal{L}_{scl} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\left(\boldsymbol{x}_i \cdot \boldsymbol{x}_p\right)/\tau\right)}{\sum_{a \in A(i)} \exp\left(\left(\boldsymbol{x}_i \cdot \boldsymbol{x}_a\right)/\tau\right)}, \quad (1)$$

where $i \in I = \{1, \ldots, 2N\}$ denotes the index of an instance, $A(i)$ denotes all indices except i , and $\boldsymbol{x}$ denotes the feature embeddings of waveforms. $P(i)$ denotes all indices of the positive instances of sample i. τ is a hyperparameter.

After calculating the SCL loss $\mathcal{L}_{scl}$ and cross-entropy loss $\mathcal{L}_{ce}$, we could calculate the new learning objective as follows:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{ce} + \lambda \mathcal{L}_{scl}, \quad (2)$$

where λ denotes a scalar weighting hyperparameter.

2.3. Inference Stage

To implement the kNN algorithm in the inference stage, we first create the datastore with all samples in the training and validating datasets. The storage format of the datastore is as follows:

$$(\mathcal{K}, \mathcal{V}) = \{(x_i, y_i), i \in \mathcal{D}\}, \quad (3)$$

where $\mathcal{D}$ is a set of all indices of the samples from training and validating data, x denotes the fixed-length representation of an input waveform computed by the trained model, and y denotes its label.

Afterward, when given an inference sample, we can retrieve its k nearest neighbors from the datastore and make predictions for its label accordingly. These neighbors are calculated based on the distance in the embedding space produced by the fine-tuned model, and we adopt the *Euclidean Distance* metric for this purpose.

Based on the predictions from both the kNN model and the trained model, we could obtain a final probability distribution of a given inference sample by a linear interpolation:

$$p(y|x) = \alpha p_{knn}(y|x) + (1-\alpha) p_{model}(y|x), \quad (4)$$

where α is a scalar weighting hyperparameter, $p_{model}(y|x)$ and $p_{knn}(y|x)$ denotes the probability distributions output from the trained model and the kNN model, respectively.

3. Experiments

3.1. Dataset

We evaluate the effectiveness of our proposed method on the IEMOCAP [21] dataset, which is widely used in SER. The dataset includes various emotions such as happy, angry, neutral, sad, surprised, excited, fearful, frustrated, disgusted, and so on. To be consistent with prior works, we conduct experiments on a subset of four emotions, namely, angry, happy, sad, and neutral, where the original happy category and excited category are merged as the happy category. In total, there are 5,531 utterances, comprising 1,103 angry, 1,636 happy, 1,084 sad, and 1,708 neutral utterances.

3.2. Experimental Settings

In this paper, we use the publicly available pre-trained model, *Wav2Vec 2.0 Base*, developed by Facebook AI. It is trained on the LibriSpeech dataset [22] through self-supervised learning without any fine-tuning. We apply 10-fold cross-validation to evaluate our results, leaving two speakers out for each fold, one for validating and the other for testing. To retain as much information as possible and save computing resources, we set the max-length of all utterances to 7.5 seconds. The batch size is set to 12 and the max training epoch is set to 150. We choose SGD optimizer with the initial learning rate of 10^{-4} and reduce it if the validation loss has not decreased for 20 consecutive epochs. The hyperparameter τ is set to 0.07. To leverage the auxiliary role of the contrastive learning loss, we test various λ values and achieve the best results with λ set to 0.1 on the IEMOCAP dataset. We search the best k for the kNN model in the range $[1, \ldots, 32]$, as well as the best $\alpha \in (0, 1)$ for each fold. We use weighted accuracy (WA) and unweighted accuracy (UA) as metrics to evaluate the performance of our proposed methods.

https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec

3.3. Results and Discussions

Table 1 shows the main results of our proposed methods at different stages, including pre-training, fine-tuning, and inference stages. Note that we utilize wav2vec2.0 as the pre-trained model for feature extraction during the pre-training stage. To evaluate the pre-training stage, the cross-entropy loss is used as the baseline (the S1 row in table 1) for fine-tuning the model for emotion prediction. We can observe a gradual improvement in performance throughout the SER pipeline as each method is applied, culminating in the best results achieved with the comprehensive approach. Particularly, the comprehensive approach outperforms the baseline system relatively by 3.90% and 4.59% on WA and UA, respectively. Figure 3 also demonstrates the same results with confusion matrices, which show more detailed performance for each emotion category. Further details of the experiments are discussed below.

Table 1: *Evaluation results on IEMOCAP at different stages of the whole SER pipeline.*

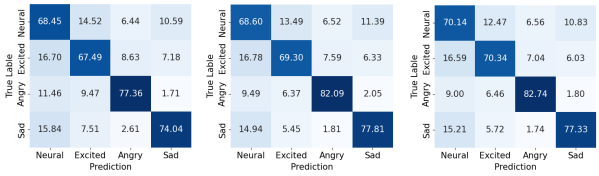
Stages	**Main Method**	**WA(%)**	**UA(%)**
S1 (Pre-training)	Wav2vec2.0	71.35	71.84
S2 (Fine-tuning)	S1 + SCL	73.32	74.45
S3 (Inference)	S2 + kNN	**74.13**	**75.14**

3.3.1. Effectiveness of SCL loss during fine-tuning

To demonstrate the effectiveness of the new loss function, we visualize the outputs of the intermediate layer using the t-SNE technique [23]. Figure 4(a) and figure 4(b) visualize the distributions of feature representations obtained by fine-tuning with only cross-entropy loss and our new loss function respectively. It could be found that after fine-tuning with a normal strategy, the model gets a basic ability to distinguish emotions. However, there still exist overlaps of certain emotions. After fine-tuning with the help of SCL loss, the boundaries between different emotions become clearer, especially for happy and neutral emotions, which are recognized as more difficult emotions to distinguish in previous researches [24]. The above findings suggest that our new learning objective combining SCL loss improves the quality of emotional representation effectively.

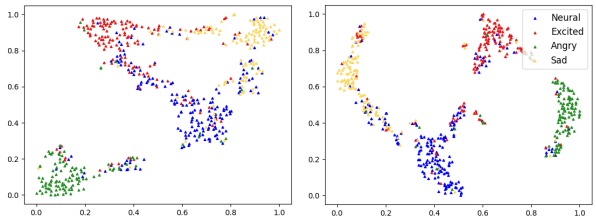
3.3.2. Performances with different data augmentations

In this paper, we also explore the model performances with different data augmentations in the fine-tuning stage, including adding noise, changing volume, adding reverberation, changing



(a) (b) (c)

Figure 3: *Confusion matrices of the results during different stages. (a): Results after pre-training with wav2vec2.0. (b): Results after fine-tuning with our new loss function. (c): Results after interpolating model outputs with a kNN model in the inference stage. Numbers represent UA for each emotion.*



(a) (b)

Figure 4: *Visualization of the distributions about feature representations when fine-tuning with different strategies. (a): Outputs after fine-tuning with only cross-entropy loss. (b): Outputs after fine-tuning with the help of SCL.*

pitch, and mixing the above methods. The experimental results of various augmentations are shown in table 2.

It could be found that mixing several methods is more effective for improving the classification ability of the model. A complex method leads to more differences between *positives*, which could make the model learn more information and have a stronger ability to distinguish emotions in a complicated situation. Mixed data augmentation helps contrastive learning play a better role in the fine-tuning stage.

Table 2: *Performances with different methods of data augmentations in the fine-tuning stage.*

Augmentation Methods	**WA(%)**	**UA(%)**
Noise	72.02	73.23
Volume	72.72	73.97
Reverberation	72.64	73.81
Pitch	72.75	73.88
Mixed augmentations	**73.32**	**74.45**

3.3.3. Effectiveness of interpolation in the inference stage

In the inference stage, we employ a kNN model to interpolate the fine-tuned model. To investigate whether SCL improves the sample distances and thus benefits kNN, we apply the kNN model to two fine-tuned models with different strategies: one fine-tuned with cross-entropy loss only (Loss w/o SCL), and the other with the incorporation of SCL (Loss w/ SCL). The results are reported in table 3.

Our experiments reveal that applying a kNN model to interpolate the original outputs leads to improved performance, regardless of the fine-tuning strategy used. By leveraging the feature information obtained during training, the kNN model is able to correct predictions made by the original model.

Table 3: *Performances before and after introducing kNN algorithm in the inference stage with different fine-tuning strategies*

Fine-tuning	**Inference**	**WA(%)**	**UA(%)**	
Loss w/o SCL	w/o kNN	71.35	71.84	
Loss w/o SCL	w/ kNN	71.85	72.61	+
Loss w/ SCL	w/o kNN	73.32	74.45	
Loss w/ SCL	w/ kNN	**74.13**	**75.14**	+

Furthermore, the improvements in WA provide an intuitive reflection of the total number of corrected samples, highlighting the promising correction capability of the kNN model in our work. As shown in table 3, after fine-tuning with SCL, the improvement of WA is more obvious, suggesting that kNN could play a better role in correcting the predictions under this condition. Additionally, in the embedding space generated by SCL, the predictions of some samples have already been corrected, indicating the presence of hard samples with incorrect predictions. In such cases, kNN can still leverages the improved sample distances to enhance the model's classification ability without any additional training.

3.4. Comparison to SOTA Approaches

Table 4 presents a comparison of our proposed method with recent state-of-the-art (SOTA) approaches. The results show that our proposed method, which applies improvements throughout the whole SER pipeline, achieves a relative improvement of 2.28% and 1.13% on WA and UA, respectively, compared to the best results of the SOTA approaches. These findings demonstrate the effectiveness of our approach.

Table 4: *Performance comparison of our proposed methods with SOTA approaches on IEMOCAP.*

Model	**WA(%)**	**UA(%)**
Zou et al. [25]	71.64	72.70
Lu et al. [26]	71.72	72.56
Hu et al. [27]	69.31	70.11
Hu et al. [28]	72.48	67.72
Wav2vec2.0-PT [29]	–	66.30
Wav2vec2.0 P-TAPT [30]	–	74.30
Ours	**74.13**	**75.14**

4. Conclusions

In this paper, we proposed a comprehensive framework to improve the performance of SER throughout its lifecycle, including pre-training, fine-tuning, and inference stages. We conducted a series of experiments which proved the effectiveness and necessity of our proposed methods in each stage. In comparison to state-of-the-art results, our proposed methods showed significant improvements on both WA and UA.

5. Acknowledgments

This work was supported by the National Key R&D Program of China (Grant No.2022ZD0116307) and NSF China (Grant No.62271270).

6. References

[1] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, "Speech emotion recognition from spectrograms with deep convolutional neural network," in *2017 International Conference on Platform Technology and Service (PlatCon)*, 2017, pp. 1–5.

[2] A. Satt, S. Rozenberg, and R. Hoory, "Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms," in *Proc. Interspeech 2017*, 2017, pp. 1089–1093.

[3] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2227–2231.

[4] P. Kumawat and A. Routray, "Applying TDNN Architectures for Analyzing Duration Dependencies on Speech Emotion Recognition," in *Proc. Interspeech 2021*, 2021, pp. 3410–3414.

[5] Y. Liu, H. Sun, W. Guan, Y. Xia, Y. Li, M. Unoki, and Z. Zhao, "A discriminative feature representation method based on cascaded attention network with adversarial strategy for speech emotion recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1063–1074, 2023.

[6] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 449–12 460.

[7] 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.

[8] S. Latif, R. Rana, S. Younis, J. Qadir, and J. Epps, "Transfer learning for improving speech emotion classification accuracy," *Proc. Interspeech 2018*, pp. 257–261, 2018.

[9] N. K. Kim, J. Lee, H. K. Ha, G. W. Lee, J. H. Lee, and H. K. Kim, "Speech emotion recognition based on multi-task learning using a convolutional neural network," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 704–707.

[10] S. Zhou and H. Beigi, "A transfer learning method for speech emotion recognition from automatic speech recognition," *arXiv preprint arXiv:2008.02863*, 2020.

[11] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[12] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," in *International Conference on Learning Representations*, 2015.

[13] X. Ma, Z. Wu, J. Jia, M. Xu, H. Meng, and L. Cai, "Emotion Recognition from Variable-Length Speech Segments Using Deep Learning on Spectrograms," in *Proc. Interspeech 2018*, 2018, pp. 3683–3687.

[14] M. Li, B. Yang, J. Levy, A. Stolcke, V. Rozgic, S. Matsoukas, C. Papayiannis, D. Bone, and C. Wang, "Contrastive unsupervised learning for speech emotion recognition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6329–6333.

[15] Z. Lian, Y. Li, J. Tao, and J. Huang, "Speech emotion recognition via contrastive loss under siamese networks," in *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and First Multi-Modal Affective Computing of Large-Scale Multimedia Data*, 2018, pp. 21–26.

[16] V. S. Alaparthi, T. R. Pasam, D. A. Inagandla, J. Prakash, and P. K. Singh, "Scser: Supervised contrastive learning for speech emotion recognition using transformers," in *2022 15th International Conference on Human System Interaction (HSI)*. IEEE, 2022, pp. 1–7.

[17] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," in *International Conference on Learning Representations*, 2021.

[18] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," in *International Conference on Learning Representations*, 2020.

[19] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 3929–3938.

[20] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 18 661–18 673.

[21] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[23] V. D. M. Laurens and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, 2008.

[24] M. Chen, X. He, J. Yang, and H. Zhang, "3-d convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1440–1444, 2018.

[25] H. Zou, Y. Si, C. Chen, D. Rajan, and E. S. Chng, "Speech emotion recognition with co-attention based multi-level acoustic information," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7367–7371.

[26] Z. Lu, L. Cao, Y. Zhang, C.-C. Chiu, and J. Fan, "Speech sentiment analysis via pre-trained features from end-to-end asr models," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7149–7153.

[27] D. Hu, X. Hu, and X. Xu, "Multiple Enhancements to LSTM for Learning Emotion-Salient Features in Speech Emotion Recognition," in *Proc. Interspeech 2022*, 2022, pp. 4720–4724.

[28] Y. Hu, Y. Tang, H. Huang, and L. He, "A Graph Isomorphism Network with Weighted Multiple Aggregators for Speech Emotion Recognition," in *Proc. Interspeech 2022*, 2022, pp. 4705–4709.

[29] L. Pepino, P. Riera, and L. Ferrer, "Emotion Recognition from Speech Using wav2vec 2.0 Embeddings," in *Proc. Interspeech 2021*, 2021, pp. 3400–3404.

[30] L.-W. Chen and A. Rudnicky, "Exploring wav2vec 2.0 fine-tuning for improved speech emotion recognition," *arXiv preprint arXiv:2110.06309*, 2021.