Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041332

ARTICLE

Using Speaker-Specific Emotion Representations in Wav2vec 2.0-Based Modules for Speech Emotion Recognition

Somin Park[1], **Mpabulungi Mark**[1], **Bogyung Park**[2] **and Hyunki Hong**[1,*]

[1]College of Software, Chung-Ang University, Seoul, 06973, Korea

[2]Department of AI, Chung-Ang University, Seoul, 06973, Korea

*Corresponding Author: Hyunki Hong. Email: honghk@cau.ac.kr

Received: 18 April 2023 Accepted: 09 August 2023 Published: 31 October 2023

ABSTRACT

Speech emotion recognition is essential for frictionless human-machine interaction, where machines respond to human instructions with context-aware actions. The properties of individuals' voices vary with culture, language, gender, and personality. These variations in speaker-specific properties may hamper the performance of standard representations in downstream tasks such as speech emotion recognition (SER). This study demonstrates the significance of speaker-specific speech characteristics and how considering them can be leveraged to improve the performance of SER models. In the proposed approach, two wav2vec-based modules (a speaker-identification network and an emotion classification network) are trained with the Arcface loss. The speaker-identification network has a single attention block to encode an input audio waveform into a speaker-specific representation. The emotion classification network uses a wav2vec 2.0-backbone as well as four attention blocks to encode the same input audio waveform into an emotion representation. These two representations are then fused into a single vector representation containing emotion and speaker-specific information. Experimental results showed that the use of speaker-specific characteristics improves SER performance. Additionally, combining these with an angular marginal loss such as the Arcface loss improves intra-class compactness while increasing inter-class separability, as demonstrated by the plots of t-distributed stochastic neighbor embeddings (t-SNE). The proposed approach outperforms previous methods using similar training strategies, with a weighted accuracy (WA) of 72.14% and unweighted accuracy (UA) of 72.97% on the Interactive Emotional Dynamic Motion Capture (IEMOCAP) dataset. This demonstrates its effectiveness and potential to enhance human-machine interaction through more accurate emotion recognition in speech.

KEYWORDS

Attention block; IEMOCAP dataset; speaker-specific representation; speech emotion recognition; wav2vec 2.0

1 Introduction

The recent rapid growth of computer technology has made human-computer interaction an integral part of the human experience. Advances in automatic speech recognition (ASR) [1] and text-to-speech (TTS) synthesis [2] have made smart devices capable of searching and responding to verbal requests. However, this only supports limited interactions and is not sufficient for interactive conversations. Most ASR methods generally focus on the content of speech (words) without regard for the intonation, nuance, and emotion conveyed through audio speech. Speech emotion recognition (SER) is one of the most active research areas in the computer science field because the friction in every human-computer interaction could be significantly reduced if machines could perceive and understand the emotions of their users and perform context-aware actions.

Previous studies used low-level descriptors (LLDs) generated from frequency, amplitude, and spectral properties (spectrogram, Mel-spectrogram, etc.) to recognize emotions in audio speech. Although the potential of hand-crafted features has been demonstrated in previous works, features and their representations should be tailored and optimized for specific tasks. Deep learning-based representations generated from actual waveforms or LLDs have shown better performance in SER.

Studies in psychology have shown that individuals have different vocal attributes depending on their culture, language, gender, and personality [3]. This implies that two speakers saying the same thing with the same emotion are likely to express different acoustic properties in their voices. The merits of considering speaker-specific properties in audio speech-related tasks have been demonstrated in several studies [4,5].

In this paper, a novel approach in which a speaker-specific emotion representation is leveraged to improve emotional speech recognition performance is introduced. The proposed model consists of a speaker-identification network and an emotion classifier. The wav2vec 2.0 [6] (base model) is used as a backbone for both of the proposed networks, where it is used to extract emotion-related and speaker-specific features from input audio waveforms. A novel tensor fusion approach is used to combine these representations into a speaker-specific emotion representation. In this tensor fusion operation, the representation vectors are element-wise multiplied by a trainable fusion matrix, and then the resultant vectors are summed up. The main contributions of this paper are summarized as follows:

- Two wav2vec 2.0-based modules (the speaker-identification network and emotion classification) that generate a speaker-specific emotion representation from an input audio segment are proposed. The two modules are trained and evaluated on the Interactive Emotional Dynamic Motion Capture (IEMOCAP) dataset [7]. Training networks on the IEMOCAP dataset is prone to over-fitting because it has only ten speakers. The representations generated by the speaker-identification network pre-trained on the VoxCeleb1 dataset [8] facilitate better generalization to unseen speakers.
- A novel tensor fusion approach is used to combine generated emotion and speaker-specific representations into a single vector representation suitable for SER. The use of the Arcface [9] and cross-entropy loss terms in the speaker-identification network was also explored, and detailed evaluations have been provided.

2 Related Work

2.1 Hand-Crafted Audio Representations

A vast array of representations and models have been explored to improve audio speech-based emotion recognition. LLDs such as pitch and energy contours have been employed in conjunction with hidden Markov models [10] to recognize a speaker's emotion from audio speech. Reference [11] used the delta and delta-delta of a log Mel-spectrogram to reduce the impact of emotionally irrelevant factors on speech emotion recognition. In this approach, an attention layer automatically drove focus to emotionally relevant frames and generated discriminative utterance-level features. Global-Aware

Multi-Scale (GLAM) [12] used Mel-frequency cepstral coefficient (MFCC) inputs and a global-aware fusion module to learn a multi-scale feature representation, which is rich in emotional information.

Time-frequency representations such as the Mel-spectrogram and MFCCs merge frequency and time domains into a single representation using the Fast Fourier Transform (FFT). Reference [13] addressed the challenges associated with the tradeoff between accuracy in frequency and time domains by employing a wavelet transform-based representation. Here, Morlet wavelets generated from an input audio sample are decomposed into child wavelets by applying a continuous wavelet transform (CWT) to the input signal with varying scale and translation parameters. These CWT features are considered as a representation that can be employed in downstream tasks.

2.2 Learning Audio Representation Using Supervised Learning

In more recent approaches, models learn a representation directly from raw waveforms instead of hand-crafted representations like the human perception emulating Mel-filter banks used to generate the Mel-spectrogram. Time-Domain (TD) filter banks [14] use complex convolutional weights initialized with Gabor wavelets to learn filter banks from raw speech for end-to-end phone recognition. The proposed architecture has a convolutional layer followed by an l_2 feature pooling-based modulus operation and a low-pass filter. It can be used as a learnable replacement to Mel-filter banks in existing deep learning models. In order to approximate the Mel-filter banks, the square of the Hanning window was used, and the biases of the convolutional layers were set to zero. Due to the absence of positivity constraints, a 1 was added to the output before applying log compression. A key limitation of this approach is that the log-scale compression and normalization that were used reduce the scale of spectrograms, regardless of their contents.

Wang et al. [15] also proposed a learned drop-in alternative to the Mel-filter banks but replaced static log compression with dynamic compression and addressed the channel distortion problems in the Mel-spectrogram log transformation using Per-Channel Energy Normalization (PCEN). This was calculated using a smoothed version of the filter bank energy function, which was computed from a first-order infinite impulse response (IIR) filter. A smoothing coefficient was used in combining the smoothed version of the filter bank energy function and the current spectrogram energy function. In order to address the compression function's fixed non-linearity, PCEN was modified to learn channel-dependent smoothing coefficients alongside the other hyper-parameters [16] in a version of the model referred to as sPer-Channel Energy Normalization (sPCEN).

2.3 Learned Audio Representation Using Self-Supervised Learning

In supervised learning, class labels are used to design convolution filters and generate task-specific representations. Due to the vast amounts of unlabeled audio data available, self-supervised learning (SSL) methods have been proposed for obtaining generalized representations of input audio waveforms for downstream tasks. These audio SSL methods can be categorized into auto-encoding, siamese, clustering, and contrastive techniques [17].

Audio2vec [18] was inspired by word2vec [19] and learned general-purpose audio representations using an auto-encoder-like architecture to reconstruct a Mel-spectrogram slice from past and future slices. Continuous Bags of Words (CBoW) and skip-gram variants were also implemented and evaluated. In the Mockingjay [20] network, bidirectional Transformer encoders trained to predict the current frame from past and future contexts were used to generate general-purpose audio representations. Bootstrap your own latent for audio (BYOL-A) [21] is a Siamese model-based architecture that assumes no relationships exist between time segments of audio samples. In this architecture, two neural networks were trained by maximizing the agreement in their outputs given the same input. Normalization and augmentation techniques were also used to differentiate between augmented versions of the same audio segment, thereby learning a general-purpose audio representation. Hidden unit bidirectional encoder representations from Transformers (HuBERT) [22] addressed the challenges associated with multiple sound units in utterance, the absence of a lexicon of input sounds, and the variable length of sound units by using an offline clustering step to provide aligned target labels for a prediction loss similar to that in BERT [23]. This prediction loss was only applied over masked regions, forcing the model to learn a combined acoustic and language model over continuous inputs. The model was based on the wav2vec 2.0 architecture that consists of a convolutional waveform encoder, projection layer, and code embedding layer but has no quantization layer. The HuBERT and wav2vec 2.0 models have similar architectures but differ in the self-supervised training techniques that they employ. More specifically, the wav2vec 2.0 masks a speech sequence in the latent space and solves a contrastive task defined over a quantization of the latent representation. On the other hand, the HuBERT model learns combined acoustic and language properties over continuous input by using an offline clustering step to provide aligned target labels for a BERT-like prediction loss applied over only the masked regions. Pseudo labels for encoded vectors were generated by applying K-means clustering on the MFCCs of the input waveforms.

Contrastive methods generate an output representation using a loss function that encourages the separation of positive from negative samples. For instance, Contrastive Learning of Auditory Representations (CLAR) [24] encoded both the waveform and spectrogram into audio representations. Here, the encoded representations of the positive and negative pairs are used contrastively.

2.4 Using Speaker Attributes in SER

The Individual Standardization Network (ISNet) [4] showed that considering speaker-specific attributes can improve emotion classification accuracy. Reference [4] used an aggregation of individuals' neutral speech to standardize emotional speech and improve the robustness of individual-agnostic emotion representations. A key limitation of this approach is that it only applies to cases where labeled neutral training data for each speaker is available. Self-Speaker Attentive Convolutional Recurrent Neural Net (SSA-CRNN) [5] uses two classifiers that interact through a self-attention mechanism to focus on emotional information and ignore speaker-specific information. This approach is limited by its inability to generalize to unseen speakers.

2.5 Wav2vec 2.0

Wav2vec 2.0 converted an input speech waveform into spectrogram-like features by predicting the masked quantization representation over an entire speech sequence [6]. The first wav2vec [25] architecture attempted to predict future samples from a given signal context. It consists of an encoder network that embeds the audio signal into a latent space and a context network that combines multiple time steps of the encoder to obtain contextualized representations. VQ-wav2vec [26], a vector quantized (VQ) version of the wav2vec model, learned discrete representations of audio segments using a future time step prediction task in line with previous methods but replaced the original representation with a Gumbel-Softmax-based quantization module. Wav2vec 2.0 adopted both the contrastive and diversity loss in the VQ-wav2vec framework. In other words, wav2vec 2.0 compared positive and negative samples without predicting future samples.

Wav2vec 2.0 comprises a feature encoder, contextual encoder, and quantization module. First, the feature encoder converts the normalized waveform into a two-dimensional (2-d) latent representation.

The feature encoder was implemented using seven one-dimensional (1-d) convolution layers with different kernel sizes and strides. A Hanning window of the same size as the kernel and a short-time Fourier transform (STFT) with a hop length equal to the stride were used. The encoding that the convolutional layers generate from an input waveform is normalized and passed as inputs to two separate branches (the contextual encoder and quantization module). The contextual encoder consists of a linear projection layer, a relative positional encoding 1-d convolution layer followed by a Gaussian error linear unit (GeLU), and a transformer model. More specifically, each input is projected to a higher dimensional feature space and then encoded based on its relative position in the speech sequence. Here, the projected and encoded input, along with its relative position, are summed and normalized. The resultant speech features are randomly masked and fed into the Transformer, aggregating the local features into a context representation (C). The quantization module discretizes the feature encoder's output into a finite set of speech representations. This is achieved by choosing V quantized representations (codebook entries) from multiple codebooks using a Gumbel softmax operation, concatenating them, and applying a linear transformation to the final output. A diversity loss encourages the model to use code book entries equally often.

The contextual representation c_t of the masked time step (t) is compared with the quantized latent representation q_t at the same time step (t). The contrastive loss makes c_t similar to q_t and c_t dissimilar to K sampled quantized representations in every masked time step ($Q \sim q_t$). The contrastive task's loss term is defined as

$$L_m = -log \frac{exp\left(\frac{c_t^T q_t}{\| c_t \| \| q_t \| \kappa}\right)}{\sum_{q^{\sim} \sim Q_t} exp\left(\frac{c_t^T q^{\sim}}{\| c_t \| \| q^{\sim} \| \kappa}\right)}, \quad (1)$$

where κ is the temperature of the contrastive loss. The diversity loss and the contrastive loss are balanced using a hyper-parameter. A more detailed description is available in the wav2vec 2.0 paper [6].

Several variations of the wav2vec 2.0 model have been proposed in recent studies [27–29]. The wav2vec 2.0-robust model [27] was trained on more general setups where the domain of the unlabeled data for pre-training data differs from that of the labeled data for fine-tuning. This study demonstrated that pre-training on various domains improves the performance of fine-tuned models on downstream tasks. In order to make speech technology accessible for other languages, several studies pre-trained the wav2vec 2.0 model on a wide range of tasks, domains, data regimes, and languages to achieve cross-lingual representations [28,29]. More specifically, in the wav2vec 2.0-xlsr and wav2vec 2.0-xls-r variations of the wav2vec 2.0 model such as wav2vec 2.0-large-xlsr-53, wav2vec 2.0-large-xlsr-53-extended, wav2vec 2.0-xls-r-300m, and wav2vec 2.0-xls-r-1b, "xlsr" indicates that a single wav2vec 2.0 model was pre-trained to generate cross-lingual speech representations for multiple languages. Here, the "xlsr-53" model is large and was pre-trained on datasets containing 53 languages. Unlike the "xlsr" variations, the "xls-r" model variations are large-scale and were pre-trained on several large datasets with up to 128 languages. Here, the "300m" and "1b" refer to the number of model parameters used. The difference between the "300m" and "1b" variations is mainly in the number of Transformer model parameters.

The wav2vec 2.0 representation has been employed in various SER studies because of its outstanding ability to create generalized representations that can be used to improve acoustic model training. SUPERB [30] evaluated how well pre-trained audio SSL approaches performed on ten speech tasks. The pre-trained SSL networks with high performance can be frozen and employed on downstream tasks. SUPERB's wav2vec 2.0 models are variations of the wav2vec 2.0 with the original weights frozen and an extra fully connected layer added. For the SER task, the IEMOCAP dataset was used. Since the outputs of SSL networks effectively represent the frequency features in the speech sequence, the length of representations varies with the length of utterances. In order to obtain a fixed-size representation for utterances, average time pooling is performed before the fully connected layer. In [31], the feasibility of partly or entirely fine-tuning these weights was examined. Reference [32] proposed a transfer learning approach in which the outputs of several layers of the pre-trained wav2vec 2.0 model were combined using trainable weights that were learned jointly with a downstream model. In order to improve SER performance, reference [33] employed various fine-tuning strategies on the wav2vec 2.0 model, including task adaptive pre-training (TAPT) and pseudo-label task adaptive pre-training (P-TAPT). TAPT addressed the mismatch between the pre-training and target domain by continuing to pre-train on the target dataset. P-TAPT achieves better performance than the TAPT approach by altering its training objective of predicting the cluster assignment of emotion-specific features in masked frames. The emotion-specific features act as pseudo labels and are generated by applying k-means clustering on representations generated using the wav2vec model.

2.6 Additive Angular Margin Loss

Despite their popularity, earlier losses like the cross-entropy did not encourage intra-class compactness and inter-class separability [34] for classification tasks. In order to address this limitation, contrastive, triplet [35], center [36], and Sphereface [37] losses encouraged the separability between learned representations. Additive Angular Margin Loss (Arcface) [9] and Cosface [38] achieved better separability by encouraging stronger boundaries between representations. In Arcface, the representations were distributed around feature centers in a hypersphere with a fixed radius. An additive angular penalty was employed to simultaneously enhance the intra-class compactness and inter-class discrepancy. Here the angular difference between an input feature vector ($x \in R^d$) and the center representation vectors of classes ($W \in R^{N\times d}$) are calculated. A margin is added to the angular difference between features in the same class to make learned features separable with a larger angular distance. Reference [39] used the Arcface loss to train a bimodal audio text network for SER and reported improved performance. A similar loss term is used in the proposed method.

Eq. (2) is the equivalent of calculating the softmax with a bias of 0. After applying a logit transformation, Eq. (2) can be rewritten as Eq. (3).

$$L = -\frac{1}{N}\sum_{i=1}^{N} \quad log\frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{n} e^{W_j^T x_i}}, \tag{2}$$

$$L = -\frac{1}{N}\sum_{i=1}^{N} \quad log\frac{e^{\|W_{y_i}\|\|x_i\|cos\theta_{y_i}}}{\sum_{j=1}^{n} e^{\|W_j\|\|x_i\|cos\theta_j}}, \tag{3}$$

where $\|\cdot\|$ is the l_2 normalization and θ_j is the angle between W_j and x_i. In Eq. (4), the additive margin penalty (m) is only added to the angle (θ_{y_i}) between the target weight (W_{y_i}) and the features (x_i). The features are re-scaled using the scaling factor (s). The final loss is defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)} + \sum_{j=1,j\neq y_i}^{n} e^{s(cos\theta_j)}}. \tag{4}$$

Reference [39] demonstrated the Arcface loss term's ability to improve the performance of SER models. It is therefore employed in training the modules proposed in this study.

3 Methodology

In order to leverage speaker-specific speech characteristics to improve the performance of SER models, two wav2vec 2.0-based modules (the speaker-identification network and emotion classification network) trained with the Arcface loss are proposed. The speaker-identification network extends the wav2vec 2.0 model with a single attention block, and it encodes an input audio waveform into a speaker-specific representation. The emotion classification network uses a wav2vec 2.0-backbone as well as four attention blocks to encode the same input audio waveform into an emotion representation. These two representations are then fused into a single vector representation that contains both emotion and speaker-specific information.

3.1 Speaker-Identification and Emotion Classification Networks

The speaker-identification network (Fig. 1) encodes the vocal properties of a speaker into a fixed-dimension vector (d). The wav2vec 2.0 model encodes input utterances into a latent 2-d representation of shape $R^{768\times T}$, where T is the number of frames generated from the input waveform. This latent representation is passed to a single attention block prior to performing a max-pooling operation that results in a 1-d vector of length 768. Only a single attention block was used in the speaker-identification network because it is assumed that the core properties of a speaker's voice are unaffected by his or her emotional state. In other words, a speaker can be identified by his/her voice regardless of his/her emotional state. In order to achieve a more robust distinction between speakers, the R^d shape speaker-identification representation (H_{id}) and the $R^{\#ID\times d}$ shape Arcface center representation vector (W_{id}) for speaker classes are l_2 normalized, and their cosine similarity is computed. Configurations of the speaker-identification network using the cross-entropy loss were also explored. In experiments using the cross-entropy loss, the Arcface center representation vectors for speaker classes were replaced with a fully connected (FC) layer. Then, the FC outputs were fed into a softmax function, and the probability of each speaker class was obtained. In Fig. 1, "#ID" represents the index of each speaker class. For example, in the VoxCeleb1 dataset with 1,251 speakers, the final #ID is #1,251.

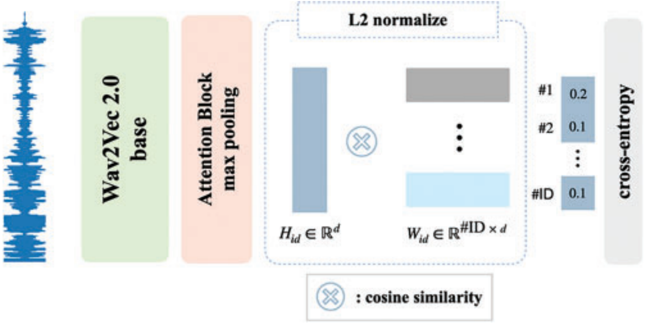
L2 normalize
Wav2Vec 2.0 base
Attention Block max pooling
$H_{id} \in \mathbb{R}^d$
$W_{id} \in \mathbb{R}^{\#ID \times d}$
#1
0.2
#2
0.1
#ID
0.1
cross-entropy
: cosine similarity

Figure 1: Architecture of the speaker-identification network with the extended wav2vec 2.0 model (left) and l_2 normalization, cosine similarity and cross-entropy loss computation (right), and a single output for each speaker class

In the emotion classification network (Fig. 2), the wav2vec 2.0 model encodes input utterances into a $R^{768\times T}$ shape representation. The encoding generated is passed to a ReLU activation layer before being fed into an FC layer and eventually passed to four attention blocks. The four attention blocks identify the parts of the generated emotion representation that are most relevant to SER. Experiments were also conducted for configurations with one, two, as well as three attention blocks. Max-pooling is applied across the time axis to the outputs of each attention block. The max pooled outputs of the attention blocks h_i are concatenated before the tensor fusion operation. During tensor fusion, an element-wise multiplication between $H_{emo} = \{h_1, h_2, \cdots, h_k\}$ and a trainable fusion matrix ($W_{fusion} \in R^{k\times d}$) is performed. As shown in Eq. (5), all the k vectors are summed to generate the final embedding.

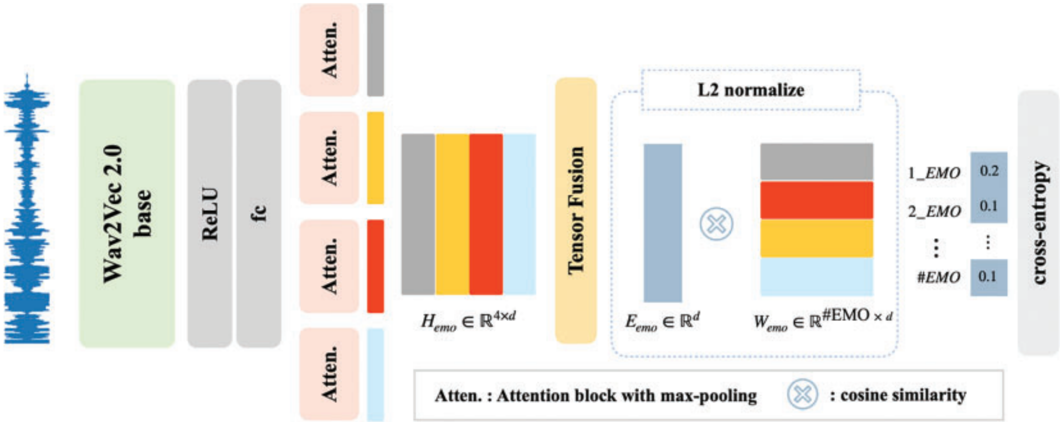
Wav2Vec 2.0 base
ReLU
fc
Atten.
Atten.
Atten.
Atten.
$H_{emo} \in \mathbb{R}^{4\times d}$
Tensor Fusion
L2 normalize
$E_{emo} \in \mathbb{R}^{d}$
$W_{emo} \in \mathbb{R}^{\#EMO\times d}$
1_EMO 0.2
2_EMO 0.1
⋮ ⋮
#EMO 0.1
cross-entropy
Atten. : Attention block with max-pooling
: cosine similarity

Figure 2: Architecture of the emotion classification network. Extended wav2vec 2.0 model (left) with four attention blocks and a tensor fusion operation. l_2 normalization, cosine similarity, and cross-entropy loss computation (right) for emotion classes with a single output for each emotion class

$$E = \sum_{i=1}^{k} e_i = \sum_{i=1}^{k} W_{fusion,i} \odot h_i, \tag{5}$$

where $e_i \in R^d$ and $W_{fusion,i} \in R^d$. The final embedding (E) is l_2 normalized prior to computing the cosine similarity between the Arcface center representation vectors ($W_{emo} \in R^{\#EMO\times d}$). In Fig. 2, "#EMO" represents the emotion class indices defined in the IEMOCAP dataset. Here, 1_EMO, 2_EMO, 3_EMO, and 4_EMO represent angry, happy, sad, and neutral emotion classes, respectively.

3.2 Speaker-Specific Emotion Representation Network

Fig. 3 shows the architecture of the proposed SER approach. The same waveform is passed to the speaker-identification network as well as the emotion classification network. The speaker representation generated by the pre-trained speaker-identification network is passed to the emotion classification network. More specifically, the output vector of the attention block from the speaker-identification network is concatenated to the outputs of the emotion classification network's four attention blocks, resulting in a total of five attention block outputs ($H \in R^{5\times d}$). The fusion operation shown in Eq. (5) combines these representations into a single speaker-specific emotion representation (E). The angular distance between the normalized tensor fused output vector and the normalized center of the four emotion representation vectors is calculated using Eq. (4). The emotion class predicted for any input waveform is determined by how close its representation vector is to an emotion class's center vector.

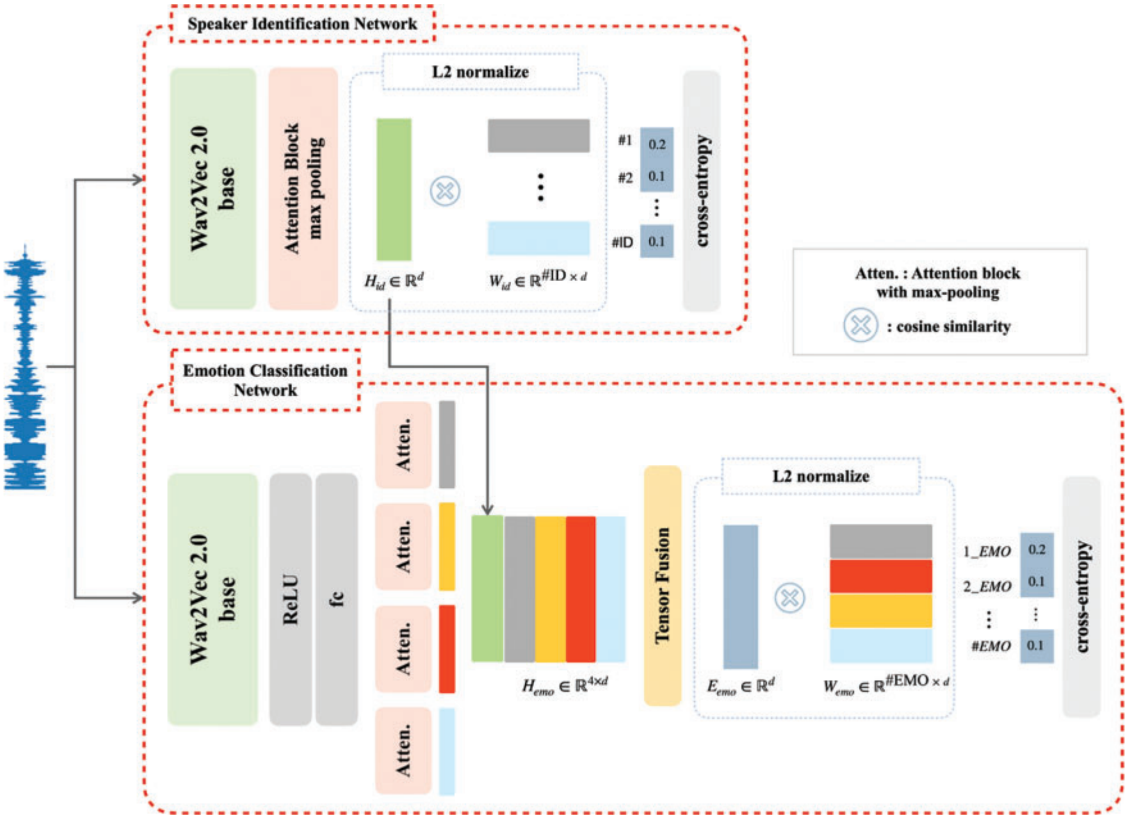
Speaker Identification Network
Wav2Vec 2.0 base
Attention Block max pooling
L2 normalize
$H_{id} \in \mathbb{R}^d$
$W_{id} \in \mathbb{R}^{\#ID \times d}$
#1 0.2
#2 0.1
#ID 0.1
cross-entropy
Atten. : Attention block with max-pooling
: cosine similarity
Emotion Classification Network
Wav2Vec 2.0 base
ReLU
fc
Atten.
Atten.
Atten.
Atten.
$H_{emo} \in \mathbb{R}^{4 \times d}$
Tensor Fusion
L2 normalize
$E_{emo} \in \mathbb{R}^d$
$W_{emo} \in \mathbb{R}^{\#EMO \times d}$
1_EMO 0.2
2_EMO 0.1
#EMO 0.1
cross-entropy

Figure 3: Architecture of the speaker-specific emotion representation model with the speaker-identification network (up) that generates a speaker representation and the emotion classification (down) that generates a speaker-specific emotion representation from emotion and speaker-identification representations

4 Experiment Details

4.1 Dataset

The IEMOCAP [7] is a multimodal, multi-speaker emotion database recorded across five sessions with five pairs of male and female speakers performing improvisations and scripted scenarios. It comprises approximately 12 h of audio-visual data, including facial images, speech, and text transcripts. The audio speech data provided is used to train and evaluate models for emotion recognition. Categorical (angry, happy, sad, and neutral) as well as dimensional (valence, activation, and dominance) labels are provided. Due to imbalances in the number of samples available for each label category, only neutral, happy (combined with exciting), sad, and angry classes have been used in line with previous studies [4,30–33,39,40]. The 16 kHz audio sampling rate used in the original dataset is retained. The average length of audio files is 4.56 s, with a standard deviation of 3.06 s. The minimum and maximum lengths of audio files are 0.58 and 34.14 s, respectively. Audio files longer than 15 s are truncated to 15 s because almost all of the audio samples in the dataset were less than 15 s long. For audio files shorter than 3 s, a copy of the original waveform is recursively appended to the end of the audio file until the audio file is at least 3 s long. Fig. 4 shows how often various emotions are expressed by male and female speakers over five sessions in the IEMOCAP dataset. As shown in

Fig. 4, the dataset is unevenly distributed across emotion classes, with significantly more neutral and happy samples in most sessions.

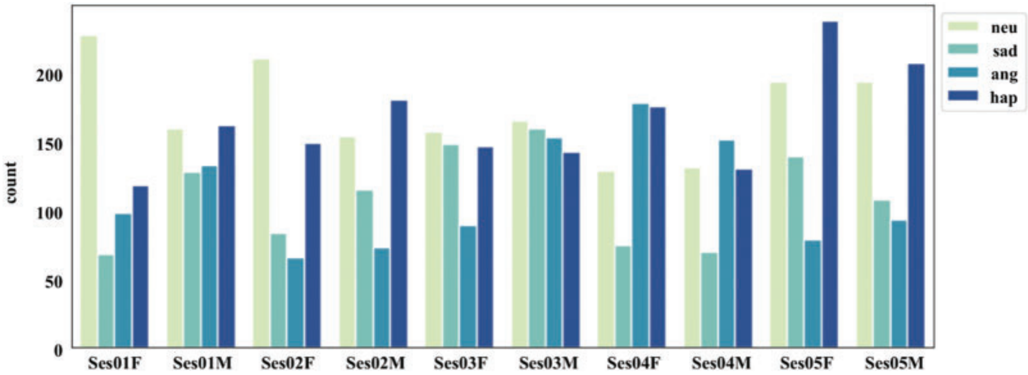

Figure 4: Distribution of male and female speakers across emotion classes in the IEMOCAP dataset

In order to generate an evenly distributed random set of samples at each epoch, emotion classes with more samples are under-sampled. This implies that samples of the training dataset are evenly distributed across all the emotional classes. Leave-one-session-out five-fold cross-validation is used.

In this study, VoxCeleb1's [8] large variation and diversity allow the speaker-identification module to be trained for better generalization to unseen speakers. VoxCeleb1 is an audio-visual dataset comprising 22,496 short interview clips extracted from YouTube videos. It features 1,251 speakers from diverse backgrounds and is commonly used for speaker identification and verification tasks. Its audio files have a sampling rate of 16 kHz with an average length of 8.2 s as well as minimum and maximum lengths of 4 and 145 s, respectively. Additionally, audio clips in VoxCeleb1 are also limited to a maximum length of 15 s for consistency in the experiments.

4.2 Implementation Details

In recent studies [31,32], pre-training the wav2vec model on the Librispeech dataset [41] (with no fine-tuning for ASR tasks) has been shown to deliver better performance for SER tasks. In this study, the wav2vec 2.0 base model was selected because the wav2vec 2.0 large model does not offer any significant improvement in performance despite an increase in computational cost [31,32]. The key difference between the "wav2vec2-large" and its base model is that it consists of an additional 12 Transformer layers that are intended to improve its generalization capacity. Using other versions of the wav2vec 2.0 model or weights may improve performance depending on the target dataset and the pre-training strategy [27–29,33]. This study proposes two networks based on the wav2vec 2.0 representation (Sub-section 2.5). In addition, reference [31] showed that either partially or entirely fine-tuning the wav2vec 2.0 segments results in the same boost in model performance on SER tasks despite the differences in computational costs. Therefore, the wav2vec 2.0 modules (the contextual encoder) used in this study were only partially fine-tuned. The model and weights are provided by Facebook research under the Fairseq sequence modeling toolkit [42].

A two-step training process ensures that the proposed network learns the appropriate attributes. First, the speaker-identification network and emotion network are trained separately. Then, the pre-trained networks are integrated and fine-tuned with the extended tensor fusion matrix to match the size of concatenated speaker-identification and emotion representations. In order to prevent over-fitting and exploding gradients, gradient values are clipped at 100 with n-step gradient accumulations. A 10^{-8}

weight decay is applied, and the Adam [43] optimizer with beta values set to (0.9, 0.98) is used. The LambdaLR scheduler reduces the learning rate by multiplying it by 0.98 after every epoch. An early stopping criterion is added to prevent over-fitting. Each attention block consists of four attention heads with a dropout rate of 0.1. In the Arcface loss calculation, the feature re-scaling factor (s) is set to 30 and the additive margin penalty (m) to 0.3 for the experiments. Experiments were conducted using Pytorch in an Ubuntu 20.04 training environment running on a single GeForce RTX 3090 GPU. The specific hyper-parameters used in the experiments are shown in Table 1.

Table 1: Hyper-parameters used during model evaluation

Module	Learning rate	Batch size	n-step gradient accumulation	Early stopping limit	Total epoch
Speaker-identification network	$3 * 10^{-5}$	16	2	5	50
Emotion network	$3 * 10^{-5}$	6	4	10	150
Integrate and fine-tune networks	10^{-5}	6	4	10	100

4.3 Evaluation Metrics

In this paper, weighted and unweighted accuracy metrics were used to evaluate the performance of the proposed model. Weighted accuracy (WA) is an evaluation index that intuitively represents model prediction performance as the ratio of correct predictions to the overall number of predictions. WA can be computed from a confusion matrix containing prediction scores as $WA = \frac{TP + TN}{TP + TN + FP + FN}$, where the number of true positive, true negative, false positive, and false negative cases are TP, TN, FP, and FN, respectively. In order to mitigate the biases associated with the weighted accuracy in imbalanced datasets such as the IEMOCAP dataset, unweighted accuracy (UA), also called average recall, is widely employed and can be computed using $UA = \frac{1}{C}\sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i}$, where C is the total number of emotion classes and is set to four for all the results presented in this study.

5 Experimental Results

5.1 Performance of Speaker-Identification Network and Emotion Classification Network

Table 2 shows the performance of the speaker-identification network on the VoxCeleb1 identification test dataset. Training the speaker identification network using the Arcface loss resulted in significantly better speaker classification than training with the cross-entropy loss. This indicates that the angular margin in the Arcface loss improves the network's discriminative abilities for speaker identification. Fig. 5 shows a t-distributed stochastic neighbor embedding (t-SNE) plot of speaker-specific representations generated from the IEMOCAP dataset using two configurations of the speaker-identification network. As shown in Fig. 5, training with the Arcface loss results in more distinct separations between speaker representations than training with the cross-entropy loss. As shown in Fig. 6, the speaker identification network may be unable to generate accurate representations for audio samples that are too short. Representations of audio clips that are less than 3 s long are particularly likely to be misclassified. In order to ensure that input audio waveforms have the information necessary to generate a speaker-specific emotion representation, a 3-s requirement is imposed. In cases where the audio waveform is shorter than 3 s, a copy of the original waveform is recursively appended to the end of the waveform until it is at least 3 s long.

Table 2: Overall performance of the proposed method when either the cross-entropy or Arcface loss was used in the speaker-identification network

Loss	WA (%)	UA (%)
Cross-entropy	87.98	87.19
Arcface	93.89	94.22

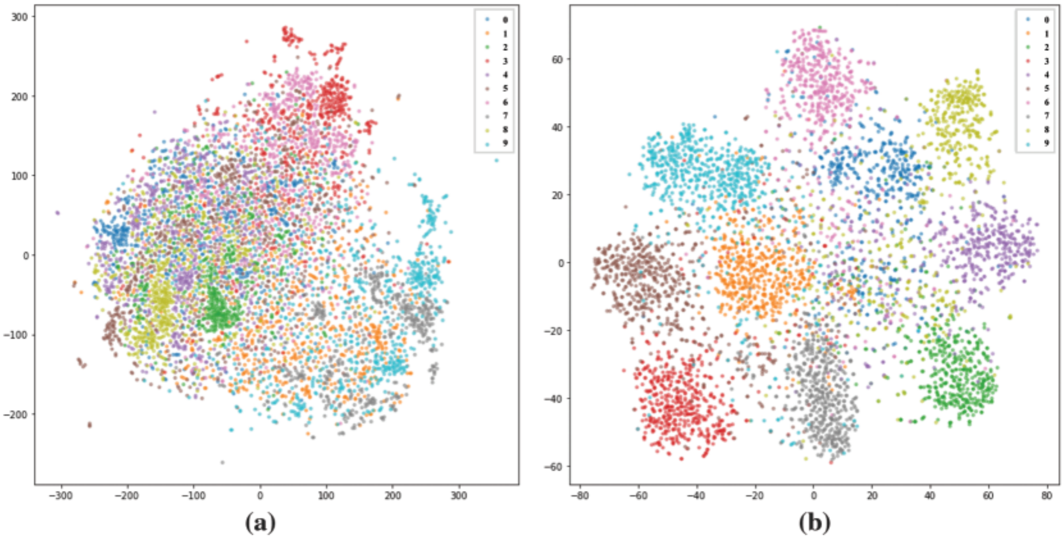

Figure 5: t-SNE plot of speaker-specific representations generated by the speaker-identification network when trained with different loss functions: (a) Cross-entropy (b) Arcface

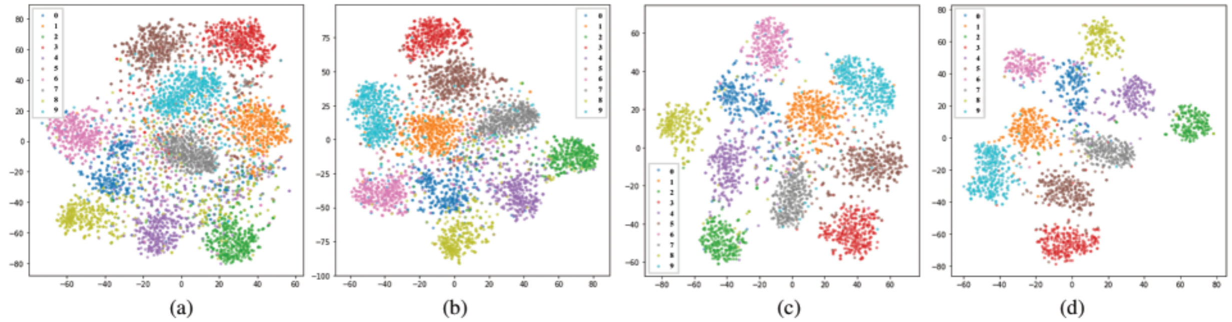

Figure 6: t-SNE plot of speaker-specific representations generated by the speaker-identification network when trained with audio segments of varying minimum lengths: (a) 1 s (b) 2 s (c) 3 s (4) 4 s

Table 3 shows a comparison of the proposed methods' performance against that of previous studies. The first four methods employed the wav2vec 2.0 representation and used the cross-entropy loss [30–32]. Tang et al. [39] employed hand-crafted features and used the Arcface loss. Here, the individual vocal properties provided by the speaker-identification network are not used. Table 3 shows that the method proposed by Tang et al. [39] has a higher WA than UA. This implies that emotion classes with more samples, particularly in the imbalanced IEMOCAP dataset, are better recognized. The wav2vec 2.0-based methods [30–32] used average time pooling to combine features across the time axis. Reference [32] also included a long short-term memory (LSTM) layer to better model the temporal features. In the proposed method, the Arcface loss is used instead of the cross-entropy loss, and an attention block is used to model temporal features. Table 3 shows that the proposed attention-based method outperforms previous methods with similar training paradigms. It also demonstrates that using four attention blocks results in significantly better performance than using one, two, or three attention blocks. This is because four attention blocks can more effectively identify the segments of the combined emotion representation that are most relevant to SER. Reference [33]'s outstanding performance can be attributed to the use of a pseudo-task adaptive pretraining (P-TAPT) strategy that is described in Subsection 2.5.

Table 3: Comparing the performance of the proposed emotion classification approach (with a varying number of attention blocks) against that of previous methods also trained with the Arcface loss

Models	# of emotion attention blocks	WA (%)	UA (%)
SUPERB [30]: wav2vec 2.0 (base/large)	–	63.43/65.64	–
Wang et al. [31]: w2v partly Fine-tuned (base/large)	–	70.21/70.99	–
Pepino et al. [32]: Global normalized	–	–	66.3 ± 0.7
Pepino et al. [32]: Speaker normalized	–	–	67.2 ± 0.7
Chen et al. [33]: P-TAPT	–	–	74.30
Tang et al. [39]: Audio only	–	71.80	69.60
Proposed method	1	70.03	71.37
	2	70.26	71.14
	3	70.36	71.14
	4	71.05	72.15

5.2 *Partially and Entirely Fine-Tuning Networks*

The proposed speaker-identification network was fine-tuned under three different configurations: fine-tuning with the entire pre-trained network frozen (All Frozen), fine-tuning with the wav2vec 2.0 segment frozen and the Arcface center representation vectors unfrozen (Arcface Fine-tuned), and fine-tuning with both the wav2vec 2.0 weights and the Arcface center representation vectors unfrozen (All Fine-tuned). The wav2vec 2.0 feature encoder (convolutional layers) is frozen in all cases [31]. The IEMOCAP dataset only has 10 individuals. Therefore, the Arcface center representation vectors are reduced from 1,251 (in the VoxCeleb1 dataset) to 8 while jointly fine-tuning both the speaker-identification network and the emotion classification network. While fine-tuning with both the wav2vec 2.0 weights and the Arcface vectors unfrozen, the loss is computed as a combination of emotion and identification loss terms as shown in Eq. (6):

$$L = \alpha \times L_{emotion} + \beta \times L_{identification}, \tag{6}$$

α and β are used to control the extent to which emotion and identification losses, respectively, affect the emotion recognition results. Since training the emotion classification network with four attention blocks showed the best performance in prior experiments, fine-tuning performance was evaluated under this configuration. Fig. 7 shows that freezing the speaker-identification network provides the best overall performance. Due to the small number of speakers in the IEMOCAP dataset, the model quickly converged on a representation that could distinguish speakers it was trained on but was unable to generalize to unseen speakers. More specifically, the frozen version of the speaker-identification module was trained on the VoxCeleb1 dataset and frozen because it has 1,251 speakers' utterances. These utterances provide significantly larger variation and diversity than the utterances of the 8 speakers (training dataset) in the IEMOCAP dataset. This implies that the frozen version can better generalize to unseen speakers than versions fine-tuned on the 8 speakers of the IEMOCAP dataset, as shown in Figs. 7b and 7c.

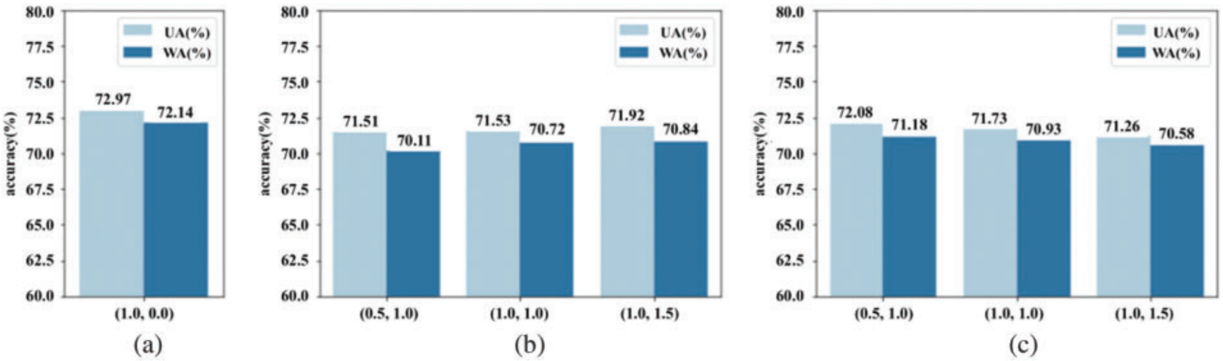

Figure 7: Performance of the proposed method with the speaker-identification network fine-tuned to various levels: (a) All Frozen (b) Arcface Fine-tuned (c) All Fine-tuned

Fig. 7b shows that increasing β, which controls the significance of the identification loss, improves emotion classification accuracy when the Arcface center representation vectors are frozen. Conversely, Fig. 7c shows that increasing β, causes the emotion classification accuracy to deteriorate when the entire model is fine-tuned. This implies that partly or entirely freezing the weights of the speaker-identification network preserves the representation learned from the 1,251 speakers of the VoxCeleb1 dataset, resulting in better emotion classification performance. On the other hand, fine-tuning the entire model on the IEMOCAP dataset's eight speakers degrades the speaker-identification network's generalization ability. More specifically, in the partly frozen version, only the attention-pooling and speaker classification layers are fine-tuned, leaving the pre-trained weights of the speaker-identification network intact.

Figs. 8 and 9 show t-SNE plots of emotion representations generated by the emotion classification network under various configurations. In Figs. 8a and 8b, the left column contains representations generated from the training set, and the right column contains those generated from the test set. In the top row of Figs. 8a and 8b, a representation's color indicates its predicted emotion class, and in the bottom row, it indicates its predicted speaker class. The same descriptors apply to Figs. 9a and 9b. More specifically, Fig. 8 illustrates the effect of employing the speaker-specific representations generated by the frozen speaker-identification network in the emotion classification network. As shown in Fig. 8, using the speaker-specific representations improves intra-class compactness and increases inter-class separability between emotional classes compared to training without the speaker-specific representation. The emotion representations generated when speaker-specific information was utilized how a clear distinction between the eight speakers of the IEMOCAP dataset and their corresponding emotion classes.

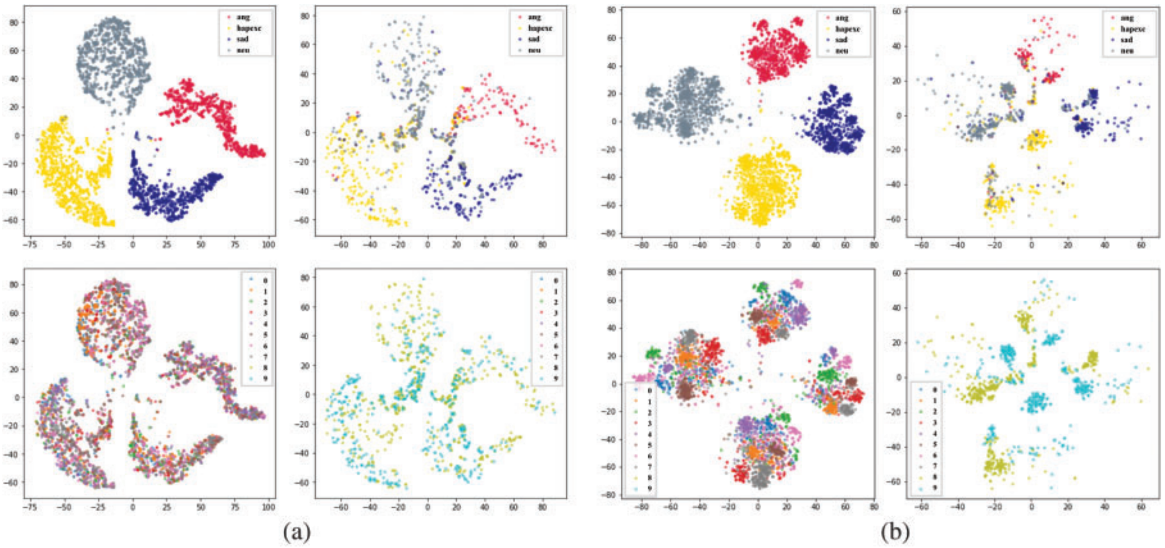

Figure 8: t-SNE plot of emotion representations generated by the emotion classification network under two configurations: (a) without the speaker-specific representation (b) with the speaker-specific representation

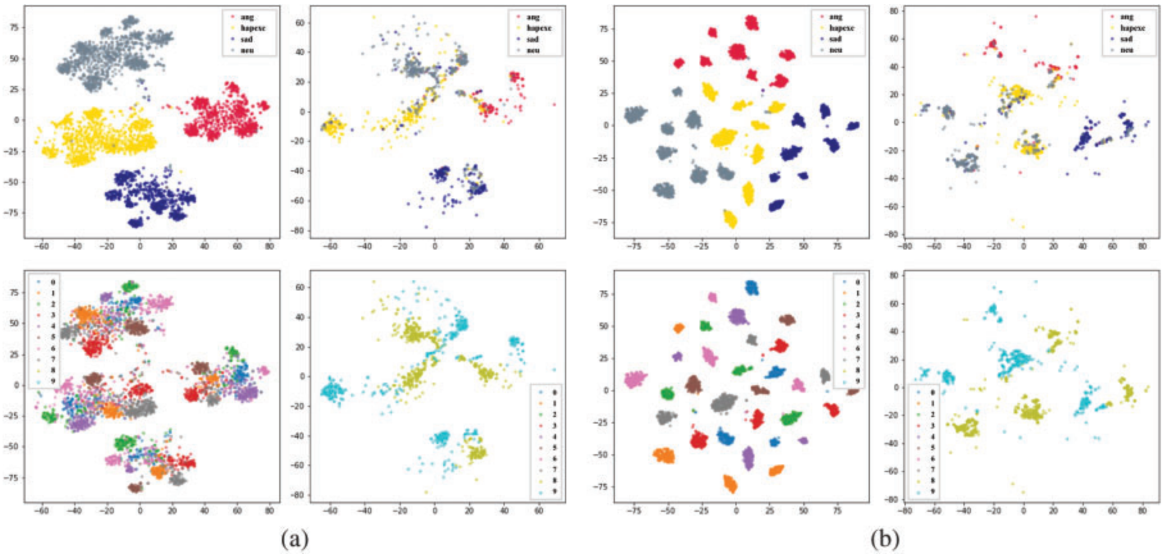

Figure 9: t-SNE plot of emotion representations generated by the emotion classification network under two configurations: (a) Only Arcface vector weights fine-tuned (b) All fine-tuned

In contrast to Figs. 9a and 9b, the result shows that fine-tuning both the speaker-identification network and the emotion classification network increases inter-class separability between the emotion representations of speakers while retaining speaker-specific information. This results in a slight improvement in the overall SER performance, which is in line with the findings shown in Figs. 7b and 7c.

5.3 Comparing the Proposed Method against Previous Methods

In Table 3, the proposed method is compared against previous SER methods that are based on the wav2vec 2.0 model or employ the Arcface loss. In Table 4, the performance of the proposed method under various configurations is compared against that of existing approaches on the IEMOCAP dataset. In Table 4, "EF" and "PF" stand for "entirely fine-tuned" and "partially fine-tuned," respectively. Experiments showed that the configuration using four attention blocks in the emotion network and fine-tuning with the speaker-identification network frozen (Fig. 7a) provided the best performance. Therefore, this configuration was used when comparing the proposed method against previous methods. The proposed method significantly improves the performance of SER models, even allowing smaller models to achieve performance close to that of much larger models. As shown in Table 4, reference [33] achieved better performance than the proposed method because it uses a pseudo-task adaptive pretraining (P-TAPT) strategy, as described in Subsection 2.5.

Reference [44] was a HuBERT-large based model which employs label-adaptive mixup as a data augmentation approach. It achieved the best performance among the approaches listed in Table 4. This is because they created a label-adaptive mixup method in which linear interpolation is applied in the feature space. Reference [45] employed balanced augmentation sampling on triple channel log Mel-spectrograms before using a CNN and attention-based bidirectional LSTM. Although this method was trained for several tasks, such as gender, valence/arousal, and emotion classification, it did not perform as well as the proposed method. This is because the proposed method uses speaker-specific properties while generating emotional representations from speaker utterances.

Table 4: Comparing the performance of the proposed method against previous SER methods

Models	Configuration	# of folds	WA (%)	UA (%)
SERAB [46]	BYOL-S/CvT 2048	–	65.10	–
SUPERB [30]	wav2vec 2.0 (base/large)	5	63.43/65.64	–
SUPERB [30]	HuBERT (base/large)		64.92/67.92	
Pepino et al. [32]	Fusion-global normalized		–	66.3 ± 0.7
Pepino et al. [32]	Dense-speaker normalized		–	67.2 ± 0.7
Chen et al. [33]	P-TAPT		–	74.30
Hou et al. [40]	–		–	66.64
Fan et al. [4]	–		70.43	65.02
Kang et al. [44]	HuBERT-large		75.37	76.04
Wang et al. [31]	wav2vec-base (EF/PT)	10	70.75/70.21	–
Wang et al. [31]	wav2vec-large (EF/PT)		70.96/70.99	
Wang et al. [31]	HuBERT-base (EF/PT)		69.83/69.68	–
Wang et al. [31]	HuBERT-large (EF/PT)		72.31/73.01	

(Continued)

Table 4 (continued)

Models	Configuration	# of folds	WA (%)	UA (%)
Liu et al. [45]	Combination of valence and arousal		70.27	66.27
Tang et al. [39]	Audio only		71.80	69.60
Proposed method	Four attention blocks with speaker-identification network (all frozen)	5	72.14	72.97

5.4 Ablation Study

Since the audio segments in the IEMOCAP are unevenly distributed across emotion classes, emotion classes with more samples were under-sampled. In order to examine the effects of an imbalanced dataset, additional experiments were conducted with varying amounts of training data. More specifically, the model was trained on the entire dataset with and without under-sampling to examine the effects of an imbalanced dataset. The best-performing configuration of the proposed model (the speaker-specific emotion representation network with four attention blocks and the speaker-identification network frozen) was used in these experiments. Table 5 shows the results of experiments conducted under four configurations. In the experiment results, both pre-trained and fine-tuned model variations showed their best performance when trained using the undersampled version of the IEMOCAP dataset. This is because under-sampling addresses the dataset's imbalance problem adequately.

Table 5: Performance of the speaker-specific emotion representation network trained under four training configurations (with under-sampled and complete versions of the IEMOCAP dataset)

		Pre-training			
		Under-sampled dataset		Complete dataset	
Accuracy (%)		WA (%)	UA (%)	WA (%)	UA (%)
Fine-tuning	Under-sampled dataset	**72.14**	**72.79**	70.86	71.84
	Complete dataset	70.92	71.96	71.68	72.26

In order to investigate the effects of using the speaker-specific representation, experiments were conducted at first using just the emotion classification network and then using the speaker-specific emotion representation network. More specifically, cross-entropy and Arcface losses, as well as configurations of the networks with 1, 2, 3, and 4 attention blocks, were used to investigate the effects of using the speaker-specific representation. As shown in Table 6, the inter-class compactness and inter-class separability facilitated by the Arcface loss results in better performance than when the cross-entropy loss is used for almost all cases. Using the speaker-specific emotion representation outperformed the bare emotion representation under almost all configurations.

Table 6: Performance (accuracy) of the speaker-specific emotion representation network under 1, 2, 3, and 4 attention block configurations and trained with cross-entropy and Arcface losses

		# of attention blocks	1	2	3	4
Emotion classification network	Cross-entropy	WA	70.16	69.10	68.96	68.48
		UA	70.36	69.91	70.45	70.00
	Arcface	WA	70.03	70.26	70.36	71.05
		UA	71.37	71.14	71.14	72.15
Speaker-specific emotion representation network	Cross-entropy	WA	70.14	70.26	70.02	69.33
		UA	70.91	71.00	70.99	70.64
	Arcface	WA	71.02	70.67	70.03	**72.14**
		UA	71.35	71.37	71.74	**72.97**

The computation time of the proposed method under various configurations was examined. The length of input audio segments (3, 5, 10, and 15 s) and number of attention blocks (1, 2, 3, and 4) were varied. The proposed model (speaker-specific emotion representation network) consists of two networks (speaker-identification and emotion classification). Table 6 shows the two networks' separate and combined computation times under the abovementioned configurations. As shown in Table 7, computation time increases as the length of input audio segments and the number of attention blocks increases. Experiments show that the proposed model's best-performing configuration is that in which the speaker-specific emotion representation network has four attention blocks. Under this configuration, the model can process an audio segment in 27 ms.

Table 7: Computation time (ms) of the proposed networks (speaker-identification, emotion classification, and speaker-specific emotion representation networks) for input audio segments of varying lengths (3, 5, 10, and 15 s)

			Input audio segment length (seconds)			
			3	5	10	15
Speaker-identification network			2.52	4.20	8.32	13.08
Emotion classification network	# of attention blocks	1	2.49	4.16	8.24	12.96
		2	2.57	4.28	8.52	13.41
		3	2.64	4.40	8.78	13.84
		4	2.71	4.52	9.00	14.22
Speaker-specific emotion representation network (four attention blocks)			5.28	8.79	17.39	27.26

6 Conclusion

This study proposes two modules for generating a speaker-specific emotion representation for SER. The proposed emotion classification and speaker-identification networks are based on the wav2vec 2.0 model. The networks are trained to respectively generate emotion and speaker representations from an input audio waveform using the Arcface loss. A novel tensor fusion approach was used to combine these representations into a speaker-specific representation. Employing attention blocks and max-pooling layers improved the performance of the emotion classification network. This was associated with the attention blocks' ability to identify which segments of the generated representation were most relevant to SER. Training the speaker-identification network on the VoxCeleb1 dataset (1,251 speakers) and entirely freezing it while using four attention blocks in the emotion network provided the best overall performance. This is because of the proposed method's robust generalization capabilities that extend to unseen speakers in the IEMOCAP dataset. The experiment results showed that the proposed approach outperforms previous methods with similar training strategies. In future works, various wav2vec 2.0 and HuBERT model variations are to be employed to improve the proposed method's performance. Novel pre-training and fine-tuning strategies, such as TAPT and P-TAPT, are also to be explored.

Acknowledgement: The authors extend their appreciation to the University of Oxford's Visual Geometry Group as well as the University of Southern California's Speech Analysis & Interpretation Laboratory for their excellent work on the VoxCeleb1 and IEMOCAP datasets. The authors are also grateful to the authors of the wav2vec 2.0 (Facebook AI) for making their source code and corresponding model weights available.

Funding Statement: This research was supported by the Chung-Ang University Graduate Research Scholarship in 2021.

Author Contributions: Study conception and model design: S. Park, M. Mpabulungi, B. Park; analysis and interpretation of results: S. Park, M. Mpabulungi, H. Hong; draft manuscript preparation: S. Park, M. Mpabulungi, H. Hong. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The implementation of the proposed method is available at https://github.com/ParkSomin23/2023-speaker_specific_emotion_SER.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg *et al.,* "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. of the 33rd Int. Conf. on Machine Learning*, New York, NY, USA, vol. 48, pp. 173–182, 2016.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly *et al.,* "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Calgary, AB, Canada, pp. 4779–4783, 2018.

[3] L. Marianne and P. Belin, "Human voice perception," *Current Biology*, vol. 21, no. 4, pp. R143–R145, 2011.

[4] W. Fan, X. Xu, B. Cai and X. Xing, "ISNet: Individual standardization network for speech emotion recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1803–1814, 2022.

[5] C. Le Moine, N. Obin and A. Roebel, "Speaker attentive speech emotion recognition," in *Proc. of Interspeech*, Brno, Czechia, pp. 2866–2870, 2021.

[6] 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. 12449–12460, 2020.

[7] C. Busso, M. Bulut, C. C. Lee, E. A. Kazemzadeh, E. Provost *et al.,* "IEMOCAP: Interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[8] A. Nagrani, J. S. Chung and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. of Interspeech*, Stockholm, Sweden, pp. 2616–2620, 2017.

[9] J. Deng, J. Guo, N. Xue and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 4685–4694, 2019.

[10] B. Schuller, G. Rigoll and M. Lang, "Hidden markov model-based speech emotion recognition," in *2003 Int. Conf. on Acoustics, Speech, and Signal Processing*, Hong Kong, China, pp. II-1, 2003.

[11] 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.

[12] W. Zhu and X. Li, "Speech emotion recognition with global-aware fusion on multi-scale feature representation," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Singapore, pp. 6437–6441, 2022.

[13] A. Dutt and G. Paul, "Wavelet multiresolution analysis based speech emotion recognition system using 1D CNN LSTM networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2043–2054, 2023.

[14] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schaiz, G. Synnaeve *et al.,* "Learning filterbanks from raw speech for phone recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Calgary, AB, Canada, pp. 5509–5513, 2018.

[15] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, New Orleans, LA, USA, pp. 5670–5674, 2017.

[16] N. Zeghidour, O. Teboul, F. de Chaumont Quitry and M. Tagliasacchi, "Leaf: A learnable frontend for audio classification," in *Int. Conf. on Learning Representations*, Vienna, Austria, 2021.

[17] S. Liu, A. M. Ragolta, E. P. Cabaleiro, K. Qian, X. Jing *et al.,* "Audio self-supervised learning: A survey," *Patterns*, vol. 3, no. 12, pp. 100616, 2022.

[18] M. Tagliasacchi, B. Gfeller, F. de Chaumont Quitry and D. Roblek, "Pre-training audio representations with self-supervision," *IEEE Signal Processing Letters*, vol. 27, pp. 600–604, 2020.

[19] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. of Workshop at Int. Conf. on Learning Representations*, Scottsdale, Arizona, USA, 2013.

[20] A. T. Liu, S. Yang, P. Chi, P. Hsu and H. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Barcelona, Spain, pp. 6419–6423, 2020.

[21] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada and K. Kashino, "BYOL for audio: Self-supervised learning for general-purpose audio representation," in *Int. Joint Conf. on Neural Networks*, Shenzhen, China, pp. 1–8, 2021.

[22] W. Hsu, B. Bolte, Y. H. H. Tsai, K. Lakhotia, R. Salakhutdinov *et al.,* "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.

[23] D. Jacob, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/pdf/1810.04805.pdf

[24] H. Al-Tahan and Y. Mohsenzadeh, "Clar: Contrastive learning of auditory representations," in *Proc. of the 24th Int. Conf. on Artificial Intelligence and Statistics*, Virtual, vol. 130, pp. 2530–2538, 2021.

[25] S. Schneider, A. Baevski, R. Collobert and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. of Interspeech*, Graz, Austria, pp. 3465–3469, 2019.

[26] A. Baevski, S. Schneider and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Int. Conf. on Learning Representations*, Virtual, 2020.

[27] W. N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu *et al.,* "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training," arXiv preprint arXiv:2104.01027, 2021.

[28] A. Conneau, A. Baevski, R. Collobert, A. Mohamed and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," in *Proc. of Interspeech*, Brno, Czechia, pp. 2426–2430, 2021.

[29] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu *et al.,* "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. of Interspeech*, Incheon, Korea, pp. 2278–2282, 2022.

[30] S. Yang, P. H. Chi, Y. S. Chuang, C. I. J. Lai, K. Lakhotia *et al.,* "SUPERB: Speech processing universal performance benchmark," in *Proc. of Interspeech*, Brno, Czechia, pp. 1194–1198, 2021.

[31] 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," arXiv preprint arXiv:2111.02735, 2021.

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

[33] L. W. Chen and A. Rudnicky, "Exploring wav2vec 2.0 fine tuning for improved speech emotion recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Rhodes Island, Greece, pp. 1–5, 2023.

[34] W. Y. Liu, Y. D. Wen, Z. D. Yu and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Int. Conf. on Machine Learning*, New York, USA, vol. 48, pp. 507–516, 2016.

[35] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 815–823, 2015.

[36] Y. Wen, K. Zhang, Z. Li and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Computer Vision-European Conf. on Computer Vision 2016*, Amsterdam, The Netherlands, pp. 499–515, 2016.

[37] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj *et al.,* "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 212–220, 2017.

[38] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong *et.al*, "Cosface: Large margin cosine loss for deep face recognition," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 5265–5274, 2018.

[39] Y. Tang, Y. Hu, L. He and H. Huang, "A bimodal network based on audio-text-interactional-attention with arcface loss for speech emotion recognition," *Speech Communication*, vol. 143, pp. 21–32, 2022.

[40] M. Hou, Z. Zhang, Q. Cao, D. Zhang and G. Lu, "Multi-view speech emotion recognition via collective relation construction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 218–229, 2022.

[41] V. Panayotov, G. Chen, D. Povey and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, South Brisbane, QLD, Australia, pp. 5206–5210, 2015.

[42] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross *et al.,* "fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, Minnesota, pp. 48–53, 2019.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. on Learning Representations*, San Diego, CA, USA, 2015.

[44] L. Kang, L. Zhang and D. Jiang, "Learning robust self-attention features for speech emotion recognition with label-adaptive mixup," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Rhodes Island, Greece, pp. 1–5, 2023.

[45] Z. T. Liu, M. T. Han, B. H. Wu and A. Rehman, "Speech emotion recognition based on convolutional neural network with attention-based bidirectional long short-term memory network and multi-task learning," *Applied Acoustics*, vol. 202, pp. 109178, 2023.

[46] N. S. Clow, M. Kegler, P. Beckmann and M. Cernak, "SERAB: A multi-lingual benchmark for speech emotion recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Singapore, pp. 7697–7701, 2022.