big data and cognitive computing

Article

Real-Time End-to-End Speech Emotion Recognition with Cross-Domain Adaptation

Konlakorn Wongpatikaseree [1,†], Sattaya Singkul [2,†], Narit Hnoohom [1,†] and Sumeth Yuenyong [1,*,†]

[1] Department of Computer Engineering, Faculty of Engineering, Mahidol University, Nakhon Pathom 73170, Thailand; konlakorn.won@mahidol.edu (K.W.); narit.hno@mahidol.edu (N.H.)

[2] Department of Deep Innovation, SpeeChance Lab, SpeeChance Co., Ltd., Bangkok 10510, Thailand; sattaya.sin@speechance-tech.com

* Correspondence: sumeth.yue@mahidol.edu

† These authors contributed equally to this work.

Abstract: Language resources are the main factor in speech-emotion-recognition (SER)-based deep learning models. Thai is a low-resource language that has a smaller data size than high-resource languages such as German. This paper describes the framework of using a pretrained-model-based front-end and back-end network to adapt feature spaces from the speech recognition domain to the speech emotion classification domain. It consists of two parts: a speech recognition front-end network and a speech emotion recognition back-end network. For speech recognition, Wav2Vec2 is the state-of-the-art for high-resource languages, while XLSR is used for low-resource languages. Wav2Vec2 and XLSR have proposed generalized end-to-end learning for speech understanding based on the speech recognition domain as feature space representations from feature encoding. This is one reason why our front-end network was selected as Wav2Vec2 and XLSR for the pretrained model. The pre-trained Wav2Vec2 and XLSR are used for front-end networks and fine-tuned for specific languages using the Common Voice 7.0 dataset. Then, feature vectors of the front-end network are input for back-end networks; this includes convolution time reduction (CTR) and linear mean encoding transformation (LMET). Experiments using two different datasets show that our proposed framework can outperform the baselines in terms of unweighted and weighted accuracies.

Keywords: speech emotion recognition; cross-domain adaption; Wav2Vec2; XLSR; vocal tract length perturbation augmentation; embedded analysis

check for updates

Citation: Wongpatikaseree, K.; Singkul, S.; Hnoohom, N.; Yuenyong, S. Real-Time End-to-End Speech Emotion Recognition with Cross-Domain Adaptation. *Big Data Cogn. Comput.* **2022**, *6*, 79. https://doi.org/10.3390/bdcc6030079

Academic Editor: Fabrizio Marozzo

Received: 24 May 2022
Accepted: 13 July 2022
Published: 15 July 2022

1. Introduction

Speech emotion recognition (SER) has been an active research area [1–4] and represents one of the emerging fields in human–computer interaction. The quality of the human–computer interface that mimics human speech emotions relies heavily on the types of features used and on the classifier employed for recognition. Feature selection can be challenging based on speech characteristics, which depend on cultural language phonation [5], articulation [6], prosody [7], phonology [8], and speaking rate [9].

Recently, end-to-end speech emotion recognition was proposed to solve the problems of feature selection that use low engineering effort and less hyperparameter tuning [10–12]. Wav2Vec2 [13] was developed by Facebook in 2020 to present end-to-end deep learning for speech understanding based on the speech recognition task. Wav2Vec2 uses raw speech information features with an encoder model for speech encoding before predicting words. This creates a simpler workflow that requires only input speech information for processing. On the other hand, Wav2Vec2 does not perform well when applied to a low-resource language. XLSR [14] was proposed for solving low-resource language problems by sharing language knowledge. Additionally, for use in real-time applications, RT-AlexNet [4] was proposed by applying a pretrained ImageNet model and then adapting it to the speech emotion domain. This achieves better real-time performance than the baselines.

However, end-to-end deep learning requires a large dataset for good performance [15]. The Thai language is a low-resource language [12,16,17], especially in the SER task, and has only one public dataset called ThaiSER proposed by VISTEC [18]. Nevertheless, Thai speech recognition has more resources than Thai SER. This paper assumes that all speech domains contain overlapping information, which can represent many speech tasks. Therefore, for use in low-resource languages, this paper proposes end-to-end speech emotion recognition based on cross-domains, which transfer from the speech recognition domain to the speech emotion recognition domain by using a front-end and a back-end network. Additionally, we perform experiments based on real-time performance, which is evaluated by the high-resource-language Berlin German dataset (Emo-DB) [19] and low-resource-language ThaiSER [18]. Moreover, our work experiment is designed for network analysis to understand the model learning in each part by the attention weight pattern and correctness of the prediction result.

The main contributions of this paper are as follows:

- Real-time end-to-end speech emotion recognition from the cross-domain (E2ESER-CD) is proposed. E2ESER-CD transfers the speech recognition domain to the speech emotion recognition domain based on a speech recognition front-end network and speech emotion recognition back-end network, thus achieving better performance than the baselines.
- A comparison study across pretrained and fine-tuned models and across different baseline models.
- The proposed speech emotion back-end network in E2ESER-CD is built to meet two criteria: convolution time reduction (CTR) and linear mean encoding transformation (LMET).
- Network and error analysis are proposed to understand the model learning for front-end and back-end networks by the attention weight pattern of the model and the correctness of the prediction result.

2. Related Work

Speech signals [20] contain both linguistic and paralinguistic information. Linguistic information is related to the meaning of words and context; on the other hand, paralinguistic information [21,22] is related to non-verbal aspects such as pitch, tone, and rate of speaking. This is the "mood" of speech. In order to represent speech information for further analysis, researchers realized that speech is produced by vibration passing through the vocal tract, where the complete pathway also includes the tongue and the teeth [23]. In signal processing, if the shape of the vocal tract is known precisely, one is able to represent the phoneme being spoken. This is because the shape of the vocal tract appears as the envelope of the power spectrum of the short-time Fourier transform. Therefore, the Mel filter-bank (MF) can be used to accurately represent this envelope. The MF features are characterized by a bank of filters that span the range from 20 Hz to 20 kHz, which is the frequency range that humans are able to perceive and is thus widely used to represent speech signals. However, the filter-bank becomes a limitation of handcrafted features, and it is difficult to find suitable hyperparameters that keep important information to related targets. Therefore, deep learning feature extraction [13,14] is proposed to solve feature engineering problems, and this eliminates the need for manual feature engineering; unfortunately, deep learning requires a large amount of data to be effective [15]. Unlike in other domains, the amount of data in the speech domain is relatively small; thus, data augmentation has been used to help increase the amount of data. Specifically, for this work, which focuses on speech emotion recognition, it is known that speech emotion states are based on cultural variations and language styles [5–9]. Vocal tract length perturbation (VTLP) [24,25] was proposed to simulate new vocal track information in speech and increase the informative number of speakers by perturbing the vocal tract length. VTLP can provide a more informative vocal speaker, allowing the model to learn more perspectives.

Regarding the model architecture for SER, many studies have applied deep learning to this task. Venkataramanan [26] investigated deep learning in SER, and it was found that CNN was able to achieve better performance than standard machine learning. Zhao [2] proposed using CNNs together with LSTM for feature extraction. The method used a sequence of CNNs in a block style. Local features were learned using what was called the local feature learning block (LFLB), which is basically a CNN layer followed by batch normalization and pooling layers. As the name implies, LFLB learns local features within the receptive field of the convolution operation. After LFLB blocks, LSTM layers are used to extract time series contextual information to obtain global-level features. Regarding the model architecture of the SER task, many studies have applied deep learning to this task. This allows for the extraction/learning of both local and global features. An improvement to LFLB using a residual connection called DeepResLFLB was proposed in [3]. The concept was inspired by ResNet [27,28], where the residual connection combined with a repeated learning style [29] allows the model to learn more effectively, leading to overall improved performance.

Nevertheless, model training in the methods mentioned above contains more hyperparameter tuning, and feature selection is more difficult. This is a common challenge across many tasks and end-to-end learning, where the entire processing pipeline is trained as one model and requires less hyperparameter tuning [10,11], and it now appears to be the preferred approach. More directly related to this work, Wav2Vec2 [13] was recently proposed with an end-to-end learning concept. Wav2Vec2 requires only the overall speech feature and labels, and the model can find the relationship between features and labels based on representative feature spaces. The representative feature spaces are generalized because the model implements contrastive loss to consider between quantized feature encoding and context representation. This achieved better performance than traditional baselines. However, Wav2Vec2 requires more representative data to achieve high performance. XLSR [14] was proposed and requires less representative data. XLSR can outperform Wav2Vec2 based on low-resource languages by using shared quantization and shared context representation between languages.

3. Proposed Method

Our proposed method consists of a front-end network, which turns the raw speech waveform into vector embedding and a back-end network, which actually performs the speech classification. Based on the end-to-end learning concept, the two networks are connected and trained together during the fine-tuning stage. We made the assumption that all speech domains contain overlapping information, which can represent many speech tasks, i.e., both speech recognition and SER. We first introduce raw speech preparation to chop the speech into small chunks that are consistent in real-time and then normalize the raw speech chunks before feeding to model learning. Then, our front-end network is introduced using pretrained Wav2Vec2 and XLSR in the speech recognition domain. Finally, our back-end network is based on a CNN and a multilayer perceptron (MLP) pattern for mapping feature spaces to the speech emotion domain. This is shown in Figure 1.

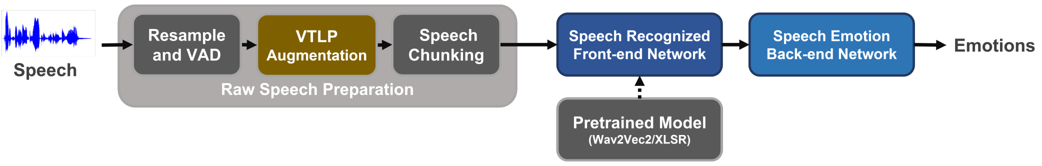
Speech
Resample and VAD
VTLP Augmentation
Speech Chunking
Raw Speech Preparation
Speech Recognized Front-end Network
Pretrained Model (Wav2Vec2/XLSR)
Speech Emotion Back-end Network
Emotions

Figure 1. Real-time E2ESER-CD framework.

3.1. Raw Speech Preparation

Cultural variations [9] and language resources are influential factors for SER datasets. Fortunately, two publicly available datasets, ThaiSER and Emo-DB, are available. ThaiSER in Thai is mixed vocalization in a relatively low-resource language, while Emo-DB in Berlin German is fast vocalization in a relatively high-resource language. These datasets represent different cultures and data sizes, which we used to contrast the difference in the performance of the proposed method across high- and low-resource languages.

SER captures emotional states from an informative speaker. Voice activity detection (VAD) [30] is chosen for filtering only speech frames and then resampling speech to a sampling rate of 16 kHz. Furthermore, deep learning benefits from larger and more varied data [15]. Thus, we utilized data augmentation based on VTLP [24,25] to add simulated vocal speaker information, which provides the variation in training data for the models.

VTLP was first proposed in [31]. The basic idea is to modify the Mel filter-bank center frequencies using (2) in [31], where the parameter α, called the warp factor, is sampled from a uniform distribution, typically in the domain $[0.9, 1.1]$. The randomness of the augmentation comes from the randomness of the value of α. By doing so, the spectrum of the speech signal is perturbed in such a way that it looks like noise had been added to each frequency bin, but the overall envelope shape of the spectrum remains the same. It is a more gentle augmentation compared to, for example, speed, pitch, or loudness augmentations. During training, the VTLP augmentation is applied to every input speech, unlike other augmentations, where the probability of being applied is <1.0, for VTLP; it can be $=1.0$ since the augmentation is gentle. Each input obtains a randomly sampled value of α. The number of batches of each training epoch is unmodified by data augmentation; in other words, the total number of batches that the models see during training is the same as if no data augmentation were performed. However, with data augmentation, the models never see exactly the same input speech twice.

Then, the speech signal is chopped with extracted features for the proper form in real-time cases. In this paper, speech is segmented into small chunks with a one-second duration, and then, features are normalized from chopped speech. Each chunk (γ) is normalized by the z-score [32,33] as in (1).

$$\gamma_{norm} = \frac{\gamma - \mu_{chunk}}{\sigma_{chunk}} \tag{1}$$

where μ_{chunk} and σ_{chunk} are the mean and standard deviation of the samples in a chunk, respectively. After normalization, the chunks are called normalized chunks (N-Chunks). We used N-Chunks as features for learning in the speech recognition front-end network and speech emotion back-end network on overall speech representative features.

3.2. Speech Recognition Front-End Network

We used the speech recognition front-end network as the first step in encoding from the speech recognition domain. The speech recognition front-end network is implemented in the same way as the Wav2Vec2 architecture using three steps: quantized feature encoding, context representation, and loss calculation. For quantized feature encoding, our framework uses a CNN encoder to encode N-Chunks as generalized latent spaces. Additionally, a masked transformer locates contextual dependencies between latent spaces based on contextual representation. Then, contrastive loss is applied to consider between quantized feature encoding and contextual representation. This is useful for generalized representative features. Contrastive loss takes the output of the network for a positive example and calculates its distance to an example of the same class and contrasts that with the distance to negative examples. The feature space distribution is intraclass compactness and interclass separation.

In addition, for a low-resource language, the XLSR architecture is implemented instead of Wav2Vec2 in the speech recognition front-end network. XLSR changes the constraint in the CNN encoder and masked transformer to a shared constraint. The shared constraint can share the language information on feature spaces, allowing the models to use information in one language on another language.

3.3. Speech Emotion Recognition Back-End Network

A speech emotion recognition back-end network was proposed for the second step as training to adapt the feature representation of the speech recognition domain to the speech emotion domain, similar to a downstream model. We propose two model patterns: convolution time reduction (CTR) and linear mean encoding transformation (LMET). CTR combines all time frames into one embedding, while LMET transforms the mean embedding of feature encoding into a linear pattern while keeping time frame information to the related mean embedding. This is shown in Figure 2.

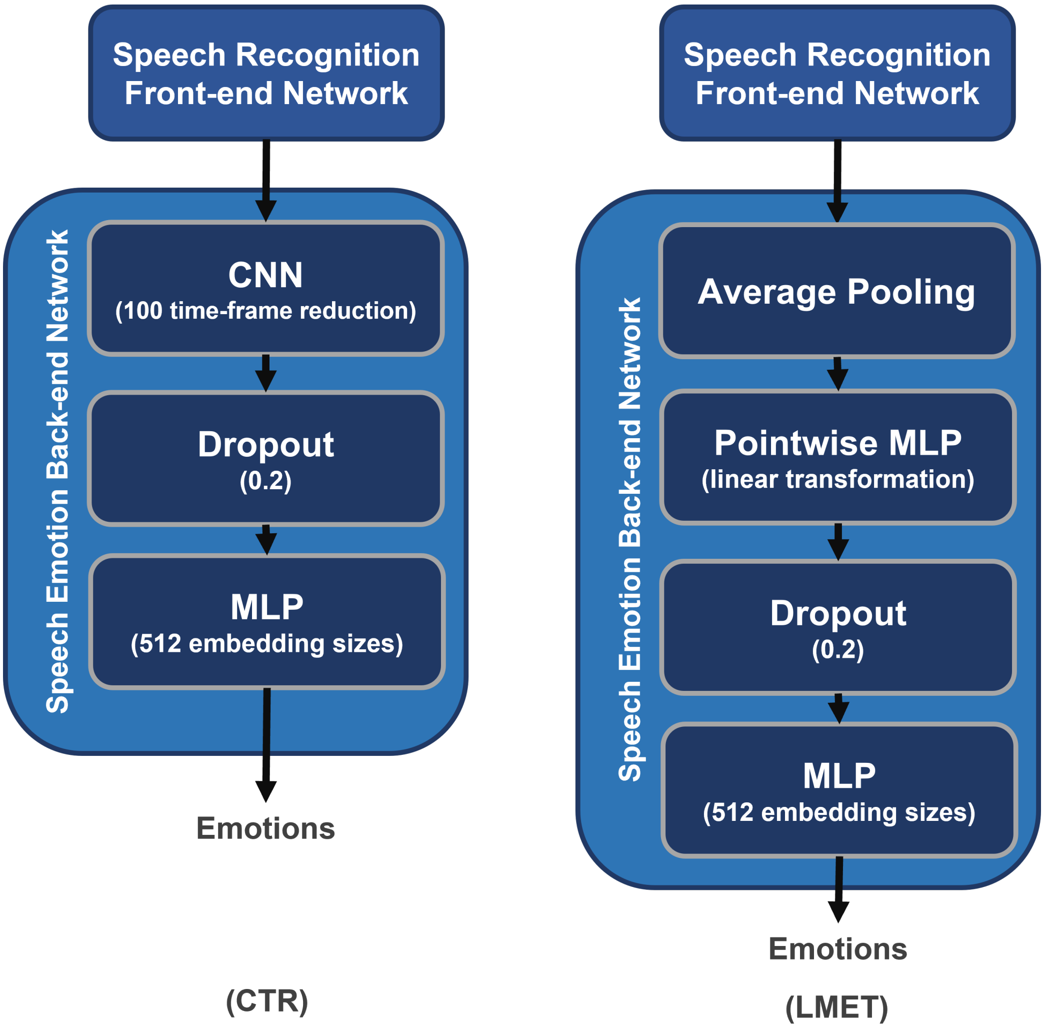
Speech Recognition Front-end Network
Speech Emotion Back-end Network
CNN
(100 time-frame reduction)
Dropout
(0.2)
MLP
(512 embedding sizes)
Emotions
(CTR)
Speech Recognition Front-end Network
Speech Emotion Back-end Network
Average Pooling
Pointwise MLP
(linear transformation)
Dropout
(0.2)
MLP
(512 embedding sizes)
Emotions
(LMET)

Figure 2. Speech emotion back-end network process. (**Left**) convolution time reduction (CTR). (**Right**) linear mean encoding transformation (LMET).

4. Experimental Setup

4.1. Datasets and Preprocessing

We performed experiments on two datasets, the Berlin German dataset (Emo-DB) [19] and ThaiSER [18]. Emo-DB consists of seven emotions: anger, disgust, boredom, joy, sadness, neutral, and fear. The dataset includes 535 utterances in German from ten native German actors—five men and five women. The audio was record with 16 bit resolution and a sampling frequency of 16 kHz. The average length of the utterances is 3 s.

ThaiSER contains five emotional states: anger, sadness, neutral, frustration, and happiness. The utterances came from 200 actors—112 men and 88 women, in various Thai accents. The audio was recorded with 16 bit resolution and a 44.1 kHz sampling frequency.

The average length of the audio files is 5.25 ± 3.85 s; the longest and shortest duration are 88.92 and 0.44 s, respectively.

In order to be able to compare across Emo-DB and ThaiSER, which have different sets of emotions, we considered only those emotions that are common across the two datasets. Thus, we fine-tuned SER only for these 4 emotions: neutral, anger, happiness (called joy in Emo-DB), and sadness. Thus, both datasets have the same number of classes for the experiments.

We first fine-tuned the pretrained wav2vec2-base-960h and XLSR (facebook/wav2vec2-large-xlsr-53) on the ASR task before actually performing SER. This is because there are more data available for the ASR task; therefore, fine-tuning on it first should benefit the downstream SER task. For both pretrained models, the tokenizer used for ASR fine-tuning was the Huggingface Wav2Vec2CTCTokenizer class. The behavior of this tokenizer is to split a text into tokens by the space character (This can be seen from the file tokenization_wav2vec2.py at Line 243). Since German has a space between words, this tokenizer can be used directly. On the other hand, Thai does not have a space between words, so we first pre-tokenized the texts by a Thai-specific tokenizer called SEFR_cut [34] and then re-joined the tokens into a single string with a space character between each pair of tokens. After this preprocessing, the same Wav2Vec2CTCTokenizer can be used for both German and Thai texts. The data for fine-tuning ASR were Common Voice 7.0 for German and Thai, respectively. The details of the datasets used for fine-tuning ASR can be found in Appendix A. Once ASR fine-tuning was complete, the model weights from the best epoch were used to initialize the models for SER fine-tuning.

For the actual SER task, the tokenizer is irrelevant since tokenization is used only in ASR in order to provide target tokens for the conversion from speech to text. SER, on the other hand, is a single-label classification task, where the output of the model is the softmax probability over the different emotion types. Therefore, the tokenizer is not needed (In the codes, the tokenizer is loaded anyway because the Huggingface API for wav2vec2, specifically the Wav2Vec2Processor class, is designed that way, but it is never actually used.). The Emo-DB and ThaiSER datasets were used for German and Thai, respectively. Regardless of the pretrained model, whether it is XLSR or wav2vec2-base-960h, both were fine-tuned (or their weights were fixed and only the downstream layers were trained, in the case where we performed transfer learning only) with the same hyperparameters.

For the experiments, the two datasets were resampled to 16 kHz and 16 bit PCM resolution.

4.2. Parameter Setting

The parameters of the front-end speech recognition networks were obtained from pretrained weights. The Wav2Vec2 [13] and XLSR [14] architectures were implemented in the same way as in the original research. Wav2Vec2 was pretrained and fine-tuned on 960 h of Librispeech dataset. The model is provided by Facebook (wav2vec2-base-960h). XLSR is a model pretrained on 53 languages and also provided by Facebook (wav2vec2-large-xlsr-53). Then, fine-tuning with the same pretraining loss enables the model to adapt to the target language dataset: the common voice corpus 7.0 [35].

We experimented on our frameworks by using two learning strategies: transfer learning only the back-end networks and fine-training end-to-end. Transfer learning uses the pretrained model from original research and is frozen to extract feature spaces before forwarding to the back-end network, allowing the model to only learn in the back-end network. On the other hand, in fine-training, our framework learns in all layers, except in the CNN feature encoding layer.

For the experiments, the speech recognition front-end networks had output feature spaces as 512 embedding dimensions. The learning rate was 0.0003; loss was connectionist temporal classification (CTC) with padding word tokens; CTC loss reduction was the mean. In addition, for adaptation to the speech emotion domain, speech emotion back-end networks were proposed that set the feature space input size to 512, the same as the output dimension of the speech recognition front-end network. We set the learning rate to 0.00002, and the loss function was set as categorical cross-entropy.

All of our proposed front-end and back-end networks used the AdamW optimizer. The batch size was 8; the number of epochs was 30; the best weight was updated whenever validation accuracy was higher than that of the previous epoch. All the hyperparameters are summarized in Table 1.

Table 1. Summary of training hyperparameters.

Parameter	Value
embedding size	512
learning rate (ASR)	3.0×10^{-4}
learning rate (SER)	2.0×10^{-5}
optimizer	AdamW
batch size	8
number of epochs	20

4.3. Evaluation Metrics

We performed 10-fold cross-validation with 80% training and 10% each for the validation and test split. The performance was measured by weighted accuracy (WA) and unweighted accuracy (UA) [36]. Weighted accuracy is the classification accuracy on the whole test set, while unweighted accuracy tests each emotion class separately and then averages the results. The number reported for all experiment is the average over all test splits. All the hyperparameters were tuned to maximize the unweighted accuracy. Due to the imbalance between the classes in the datasets, we evaluated the UA and WA results at the utterance level by first performing inference at the frames level, then used majority voting to obtain utterance-level predictions. Since the training was performed at the frame level, there are more samples overall, which lessens the impact data imbalance has on the performance of the model.

Furthermore, the word error rate (WER) was chosen to evaluate the fine-tuning of the front-end model. WER is a common metric for measuring speech-to-text accuracy in the speech recognition domain, which is basically the number of errors divided by the total number of words, as expressed in (2).

$$WER = \frac{S + I + D}{Number\ of\ Words\ Spoken} \tag{2}$$

where S is the substitution word error when a word is replaced, I is the insertion word error when a word is added that was not said, and D is the deletion word error when a word is omitted from the transcript label.

5. Results and Discussions

5.1. Results

The speech recognition front-end network and speech emotion back-end network are the main parts of our framework. The speech recognition front-end network uses a pretrained model. For pretrained performance in the Common Voice 7.0 dataset, Table 2 shows that XLSR has an approximately 16% lower WER than Wav2Vec2, and XLSR outperforms Wav2Vec2 in terms of WER in Thai. In contrast, for German, which is a high-resource language, Wav2Vec2 outperforms XLSR by approximately 3%.

For our framework, Tables 3 and 4 report the UA and WA performance based on ThaiSER and Emo-DB, respectively. As the ThaiSER results, in Table 3, XLSR with LMET-based fine-training achieves the best performance at approximately 70.73% UA and 71.27% WA. On the Emo-DB dataset, Table 4 shows that Wav2Vec2 with CTR-based pretraining achieves the best performance at approximately 88.69% UA and 91.18% WA. Nevertheless, the CTR is not significantly improved on Emo-DB, which can improve only one experimental result (Wav2Vec2 with CTR-based pre-training), as shown in Table 4. LMET is better at this point, and LMET has better generalization performance than CTR when used on both datasets, which can perform well (see Section 5.3).

With the results explored above, our framework can outperform the baselines. XLSR with LMET-based fine-training has the best performance on ThaiSER. Additionally, Wav2Vec2 with CTR-based pretraining outperforms the baselines on Emo-DB. These results show that the assumption of Sections 1 and 3 about all speech domains containing overlapping information that can represent many speech tasks is true.

Table 2. Word error rate (WER) of our proposed front-end network fine-tuned by the speech recognition domain on the Common Voice 7.0 dataset.

Model	Language	WER (%)
Wav2Vec2	German	15.6
Wav2Vec2	Thai	44.46
XLSR	German	18.5
XLSR	Thai	28.64

Table 3. Comparison of our proposed framework and baseline average UA and WA metrics on the ThaiSER dataset. The numbers in bold indicate the row with the best performance.

Model	Learning	Back-End	ThaiSER	
			UA	WA
1DLFLB+LSTM [2]	scratch	-	58.07	58.38
DeepResLFLB [3]	scratch	-	60.73	60.60
RT-AlexNet [4]	scratch	-	61.58	64.96
Wav2Vec2	Transfer Learning	CTR	69.25	68.89
Wav2Vec2	Transfer Learning	LMET	69.34	71.11
XLSR	Transfer Learning	CTR	66.61	66.57
XLSR	Transfer Learning	LMET	67.57	68.21
Wav2Vec2	Fine-training	CTR	59.98	62.56
Wav2Vec2	Fine-training	LMET	66.60	68.38
XLSR	Fine-training	CTR	65.30	65.81
XLSR	Fine-training	LMET	**70.73**	**71.27**

Table 4. Comparison of our proposed framework and baseline average UA and WA metrics on the Emo-DB dataset. The numbers in bold indicate the row with the best performance.

Model	Learning	Back-End	Emo-DB	
			UA	WA
1DLFLB+LSTM [2]	scratch	-	78.30	79.41
DeepResLFLB [3]	scratch	-	79.02	82.35
RT-AlexNet [4]	scratch	-	83.20	85.29
Wav2Vec2	Transfer Learning	CTR	**88.69**	**91.18**
Wav2Vec2	Transfer Learning	LMET	81.55	85.29
XLSR	Transfer Learning	CTR	54.93	58.82
XLSR	Transfer Learning	LMET	85.11	88.24

Table 4. *Cont.*

Model	Learning	Back-End	Emo-DB	
			UA	WA
Wav2Vec2	Fine-training	CTR	56.25	61.76
Wav2Vec2	Fine-training	LMET	53.13	58.82
XLSR	Fine-training	CTR	33.70	38.24
XLSR	Fine-training	LMET	78.42	82.35

5.2. Network Analysis

The speech recognition front-end network and speech emotion back-end network are both important for overall performance. XLSR + LMET with fine-tuning was the best-performing configuration on the ThaiSER dataset, while Wav2vec2 + CTR with transfer learning was the best on the Emo-DB dataset. In this subsection, we explain this result and present an analysis of the relationship between the Mel spectrum of the input, the attention weight pattern of the model, and the correctness of the prediction result.

5.2.1. Front-End Network Analysis

Both Wav2vec2 and XLSR are based on the transformer architecture [37], which has self-attention layers. Self-attention calculates the weight between each pair of positions in a sequence, called the attention weights. In front-end network behavior, we explored the attention weights in the last attention encoder layer for visualization. Figures 3–6 show the five main steps of the self-attention mechanism: Q, K, $softmax(Q \cdot K)$, $softmax(Q \cdot K) \cdot V$, and the linear transformation (projected attention) for final attention output. The visualization was obtained by feeding four raw speech input cases into the model: normal speech chunk, chunk containing normal silence, chunk containing mostly silence, and chunk in-between speech, as shown in Subfigures (a), (b), (c), and (d), respectively. The left column shows the input speech in Mel spectrogram form (the actual Wav2vec model takes the time-domain speech waveform as the input). The second and third columns show the Q and K values, respectively. The fourth column shows the softmax weight of the product $Q \cdot K$ for all possible pairs of locations. From the attention scores, it can be seen, for example, that one position is more correlated with the others; this can be seen as a vertical line in the attention pattern. This occurs when the speech input contains long silent intervals and is more pronounced for the case where we only perform transfer learning for the model, but do not fine-tune it. Figure 3c clearly shows the vertical line attention pattern when the input signal has a large silent interval. In the other panels of the figure, although there is a visible vertical line, it is not as dark and the attention weights at other locations are still visible. In Figure 4, it can be seen that the pattern of the attention weights is more diffused throughout different locations when the model was fine-tuned. Figures 5 and 6 repeat the same set of experiments, but for the ThaiSER dataset. It can be seen that the same general pattern as for the Emo-DB dataset still holds.

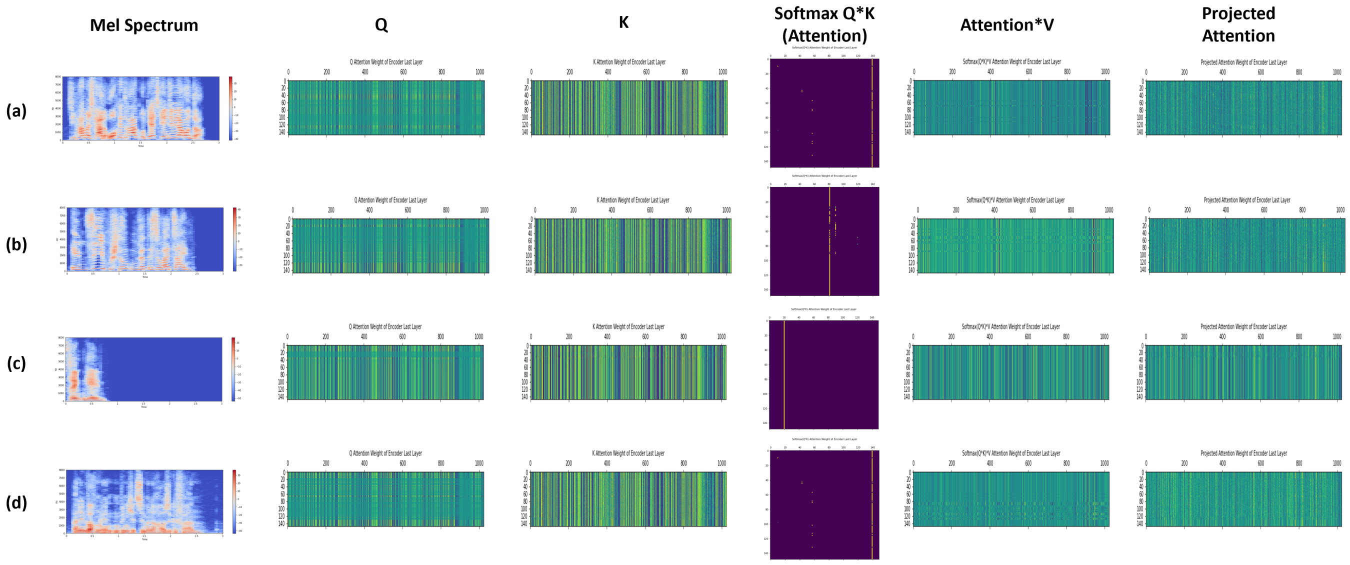

Figure 3. Attention weights of the last layer encoded feature using transfer learning of Wav2Vec2 based on the Emo-DB dataset: (**a**) normal speech chunk, (**b**) chunk containing normal silence, (**c**) chunk containing mostly silence, and (**d**) chunk between speech. Please see the full resolution image in the supplementary material.

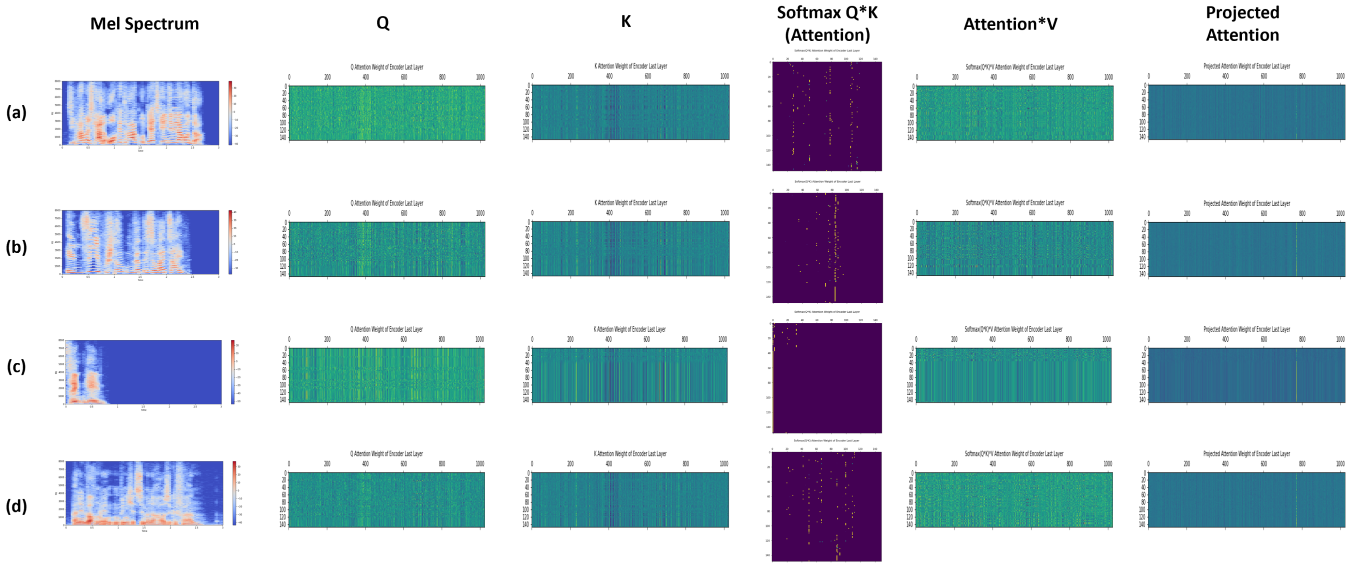

Figure 4. Attention weights of last layer encoded feature using fine-training of XLSR based on the Emo-DB dataset: (**a**) normal speech chunk, (**b**) chunk containing normal silence, (**c**) chunk containing mostly silence, and (**d**) chunk between speech. Please see the full resolution image in the supplementary material.

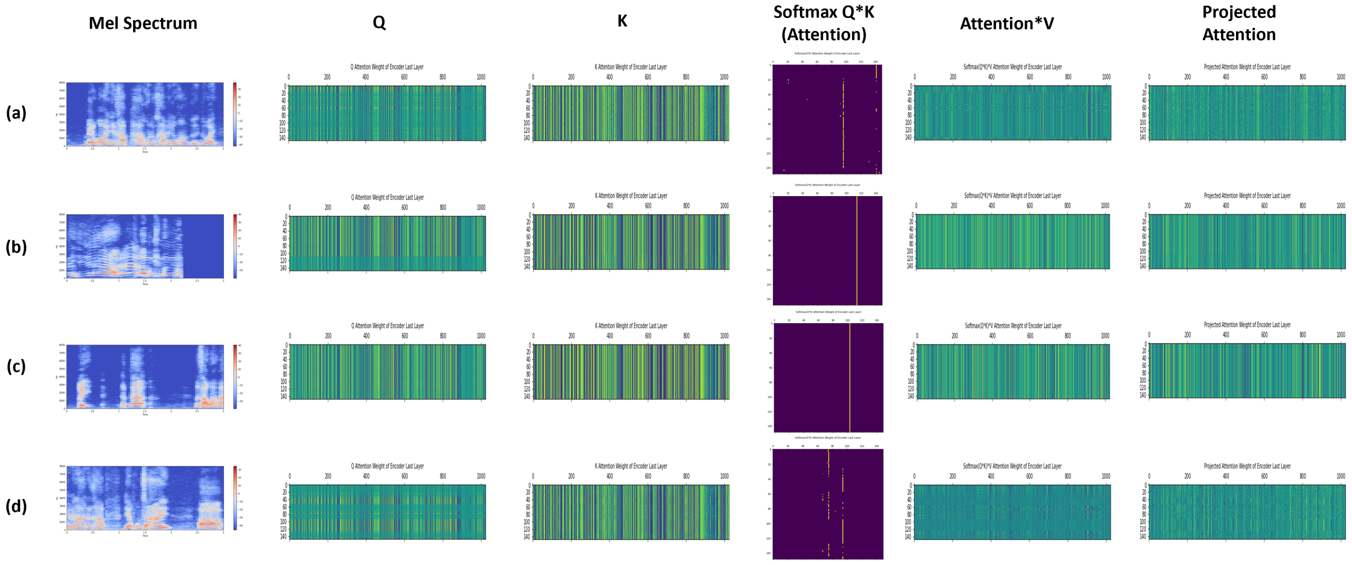

Figure 5. Attention weights of last layer encoded feature using transfer learning of Wav2Vec2 based on the ThaiSER dataset: (**a**) normal speech chunk, (**b**) chunk containing normal silence, (**c**) chunk containing mostly silence, and (**d**) chunk between speech. Please see the full resolution image in the supplementary material.

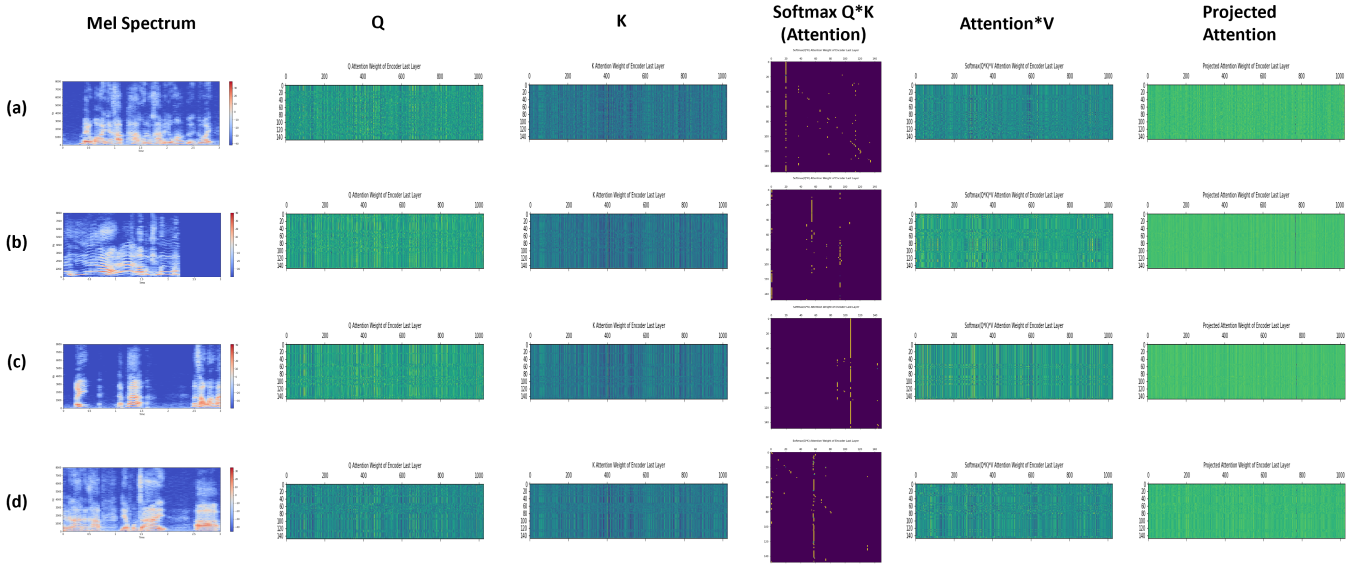

Figure 6. Attention weights of last layer encoded feature using fine-training of XLSR based on the ThaiSER dataset: (**a**) normal speech chunk, (**b**) chunk containing normal silence, (**c**) chunk containing mostly silence, and (**d**) chunk between speech. Please see the full resolution image in the supplementary material.

5.2.2. Back-End Network Analysis

The speech emotion recognition back-end network is designed for domain adaptation from speech recognition to speech emotion recognition. From the previous analysis, the size of the silent interval greatly affects the output embedding features of the front-end network, which in turn has a significant performance impact on the back-end network. As previously described, we have two different models for the back-end networks, namely CTR and LMET. The difference between them is that CTR aggregates the information from different time steps by the use of the convolution operation, while LMET uses global average pooling to collapse the information across the time dimension. This results in a significant difference between the information contained at the output of both models, which is just before the last softmax layer.

We used the Pearson product-moment correlation coefficient technique [38] to analyze the relationship between the features at various depths through the model: after the convolutional feature extraction (Z in Figure 1 of [13]), projected features (after the Wav2Vec2FeatureProjection class in the Huggingface Transformers implementation of Wav2vec2), and encoded feature (C in Figure 1 of [13]). Our hypothesis was that if the variables of the internal activation of the model are not highly correlated with each other, this leads to better performance.

Observing the projected feature column in Figures 7–10 and comparing it with the speech input, it can be seen that the high correlation areas (the yellow patches) are related to silent intervals. For Inputs (a) and (d), where there are no large silent intervals, the distribution of the high correlation areas is spread out all over the projected feature position. In contrast, for Inputs (b) and (c), where there is a long silent interval, there are large yellow patches that align with the position of the silent interval. Additionally, the projected feature is input for the encoded feature, which causes the correlation distributions of the encoded feature to be similar to those of the projected feature. Again looking at Inputs (a) and (d), we see that the distribution of the correlation values tends to be more evenly distributed than those of Inputs (b) and (c), except for the large yellow patch, which corresponds to the silent interval.

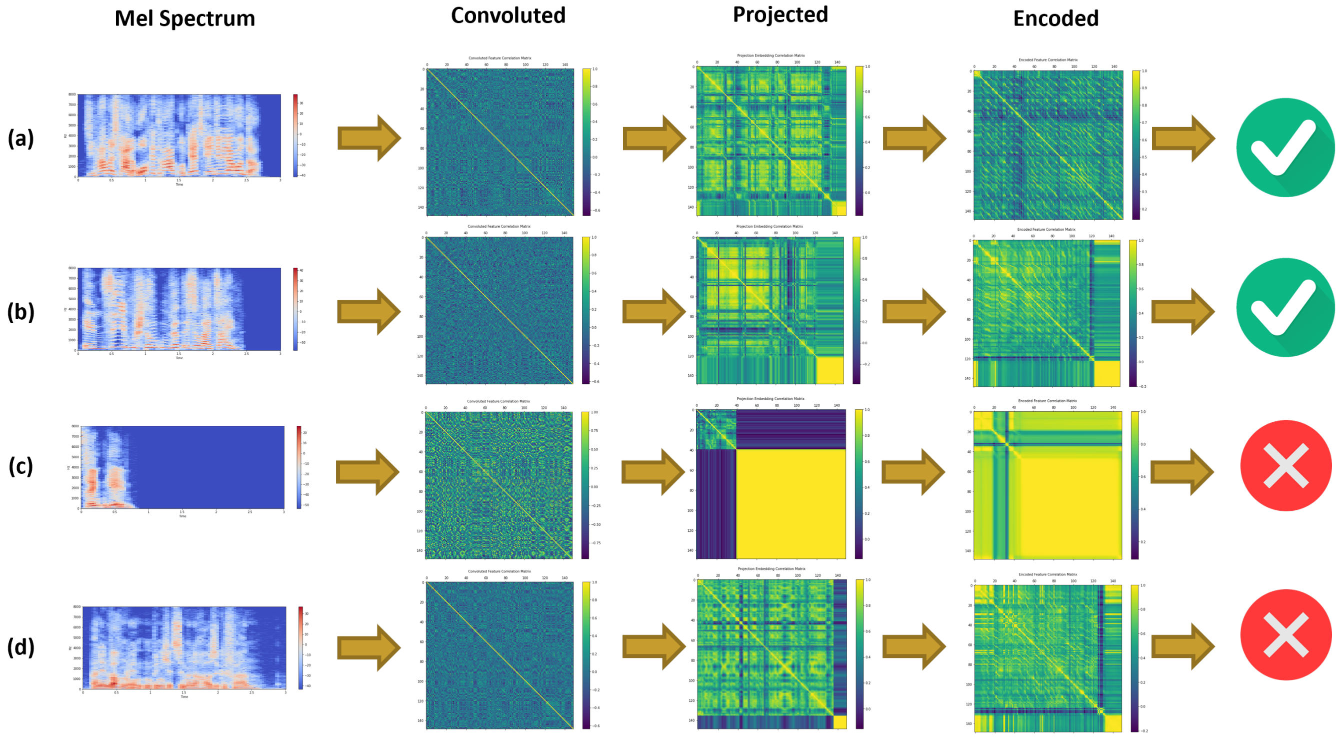

Figure 7. Correlation coefficients of E2ESER-CD using transfer learning of Wav2Vec2 with the CTR back-end network based on the Emo-DB dataset: (**a**) normal speech chunk, (**b**) chunk containing normal silence, (**c**) chunk containing mostly silence, and (**d**) chunk between speech. Note that the correct (green circle) and incorrect (red circle) signs are model results when compared with emotion labels. Please see the full resolution image in the supplementary material.

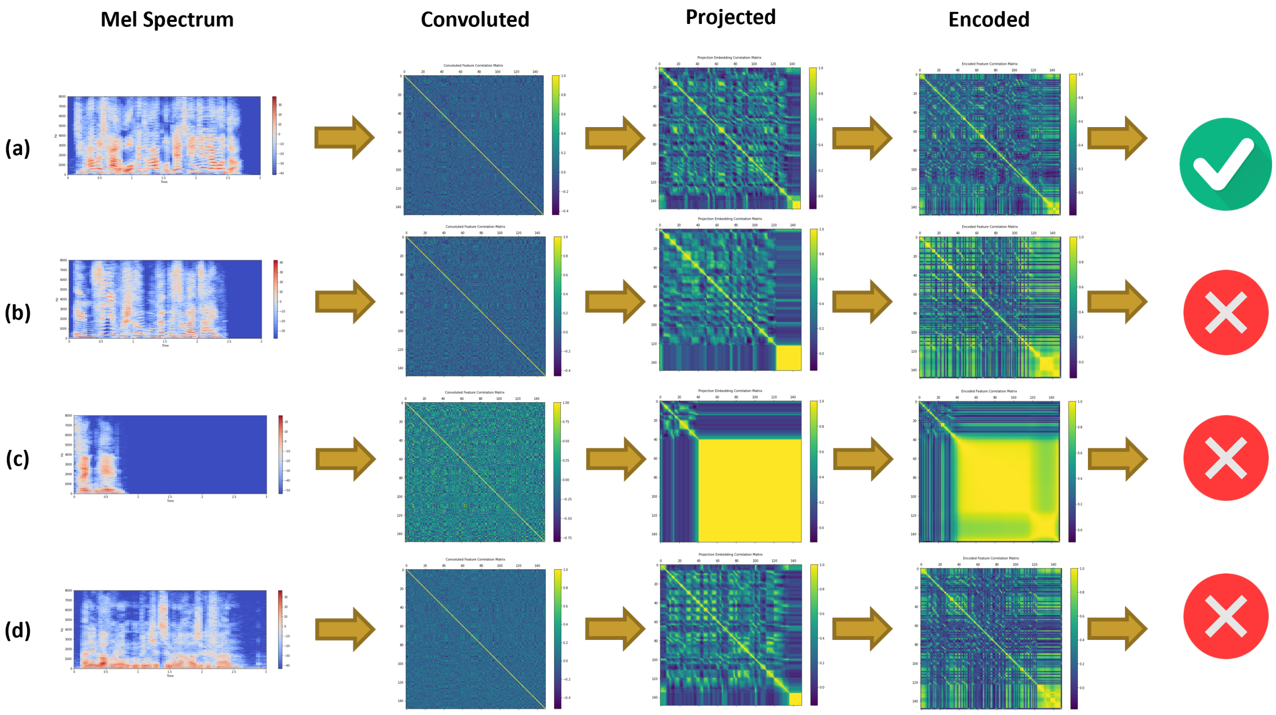

Figure 8. Correlation coefficients of E2ESER-CD using fine-training of XLSR with the LMET back-end network based on the Emo-DB dataset: (**a**) normal speech chunk, (**b**) chunk containing normal silence, (**c**) chunk containing mostly silence, and (**d**) chunk between speech. Note that the correct (green circle) and incorrect (red circle) signs are model results when compared with emotion labels. Please see the full resolution image in the supplementary material.

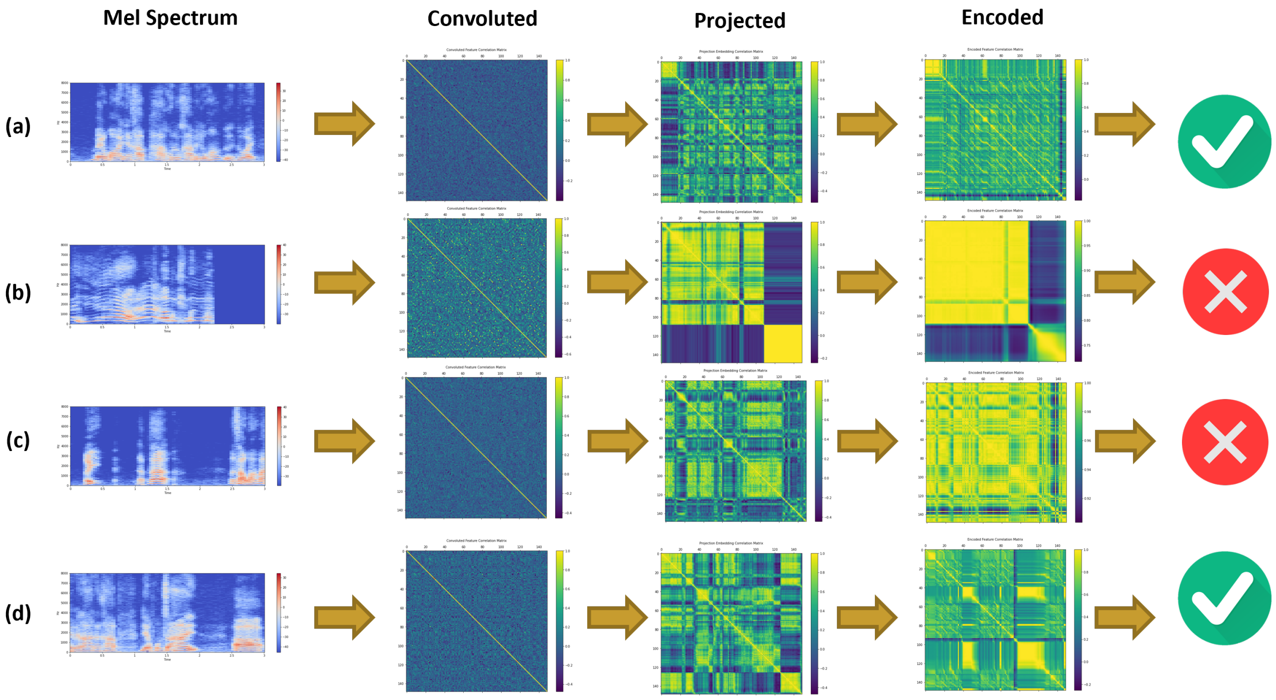

Figure 9. Correlation coefficients of E2ESER-CD using transfer learning of Wav2Vec2 with the CTR back-end network based on the ThaiSER dataset: (**a**) normal speech chunk, (**b**) chunk containing normal silence, (**c**) chunk containing mostly silence, and (**d**) chunk between speech. Note that the correct (green circle) and incorrect (red circle) signs are model results when compared with emotion labels. Please see the full resolution image in the supplementary material.

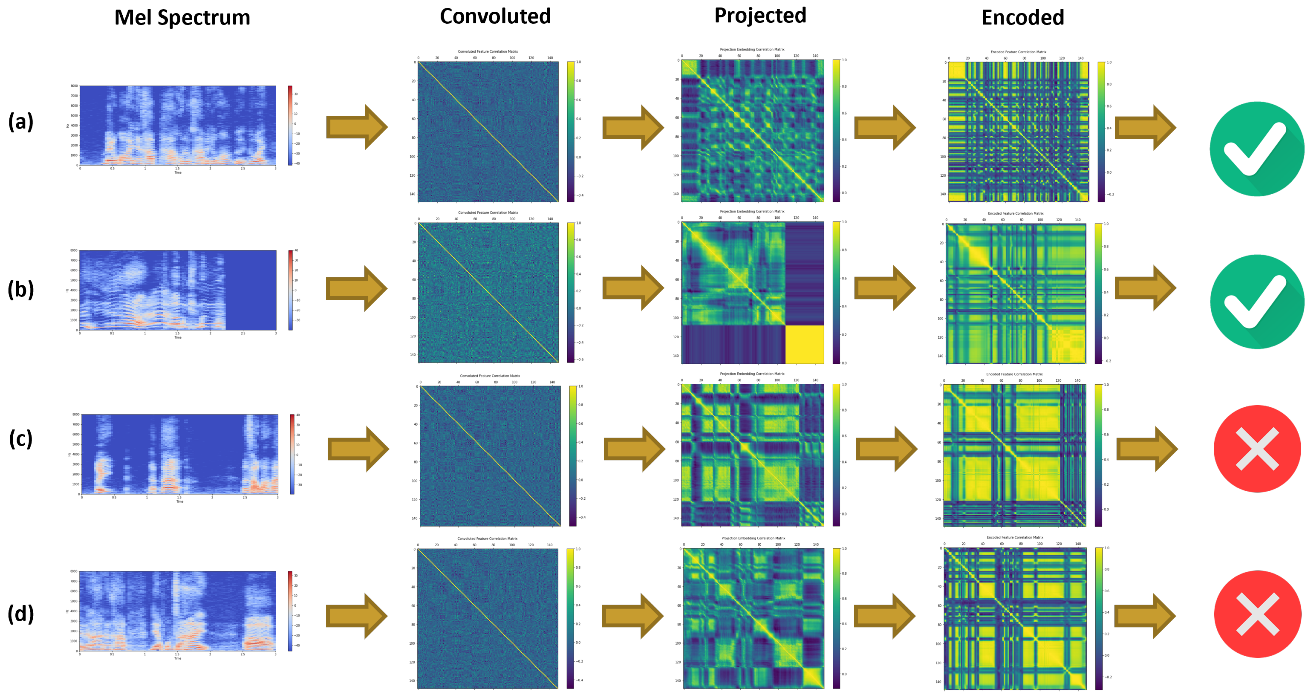

Figure 10. Correlation coefficients of E2ESER-CD using fine-training of XLSR with the LMET back-end network based on the ThaiSER dataset: (**a**) normal speech chunk; (**b**) chunk containing normal silence; (**c**) chunk containing mostly silence; and (**d**) chunk between speech. Note that the correct (green circle) and incorrect (red circle) signs are model results when compared with emotion labels. Please see the full resolution image in the supplementary material.

Evidence in support of our hypothesis can be seen in Figures 7 and 8 comparing CTR vs. LMET when the dataset is Emo-DB. This dataset has fewer long silent intervals than ThaiSER. The rightmost columns of both figures show that the false prediction (red circle) is higher for LMET than for CTR. This agrees with the results shown in Table 3. Thus, CTR is a more powerful back-end model due to the use of the convolution layer, but is sensitive to the distribution of the correlation value in the encoded features. For the ThaiSER dataset, considering Figures 9 and 10, CTR does not perform as well for this dataset because the classification accuracy is related to the uniformity of the distribution of the correlation values. If the distribution is not very uniform, then the prediction results are wrong because CTR uses convolution based on the time sequence domain, causing the nonuniform distribution pattern to persist, especially in (b) and (c). On the other hand, LMET uses average pooling when connecting with the front-end network. Average pooling looks at global information, which makes it more robust to the nonuniform distribution of correlation values. Thus, LMET works better on the ThaiSER dataset than the Emo-DB dataset, agreeing with the results shown in Table 3. In conclusion, we observed that large silent intervals seem to have more negative impact on the CTR back-end model than on LMET. We also observed that ThaiSER contains more examples with long silent intervals than Emo-DB. Large silent intervals create highly correlated features in each layer of the back-end network, as can be seen in Figures 7–10, and LMET is able to better tolerate this because it has global average pooling as the first layer, which decorrelates its output, while CTR has 1D convolution, which preserves the correlation.

5.3. Error Analysis

Wav2Vec2 with CTR using transfer learning has the best performance based on Emo-DB, as shown in Table 4. In contrast, many results using CTR do not outperform LMET, especially when using fine-training. In fine-training based on Emo-DB, Wav2Vec2 with CTR decreases by approximately 32.44% UA and 29.42% WA compared with transfer learning.

Moreover, the XLSR with CTR performance decreases by approximately 21.23% UA and 20.58% WA based on Emo-DB. In the same way, the CTR performance decreases based on ThaiSER; Wav2Vec2 decreases by approximately 9.27% UA and 6.33% WA; XLSR decreases by approximately 1.31% of UA and 0.76% of WA. In addition, in the correlation coefficient visualization, Figures 7 and 9 show that the prediction results are wrong (red circle) when the silence correlation is a highly clustered area, as noted in Section 5.2. All of these factor cause CTR to have lower silence robustness than LMET.

Nevertheless, the model performance depends on a number of speech factors. Wav2Vec2 requires a high number of speech segments for fine-training [14], and the original speech information of both datasets is not enough to explain why the Wav2Vec2 performance is lower than that of XLSR when using fine-training instead of transfer learning, as shown in Tables 3 and 4.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/bdcc6030079/s1.

Author Contributions: Conceptualization, K.W.; methodology, N.H. and S.Y.; software, S.S.; resources, K.W.; writing—original draft preparation, S.S.; writing—review and editing, S.Y.; funding acquisition, K.W. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the financial support provided by the Thammasat University Research fund under the TSRI, Contract Nos. TUFF19/2564 and TUFF24/2565, for the project of "AI Ready City Networking in RUN", based on the RUN Digital Cluster collaboration scheme.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki and approved by the Institutional Review Board from The Center of Ethical Reinforcement for Human Research, Mahidol University, with COA No. MU-CIRB 2020/214.1812.

Informed Consent Statement: This study was granted ethical approval from The Center of Ethical Reinforcement for Human Research, Mahidol University, with COA No. MU-CIRB 2020/214.1812.

Data Availability Statement: We plan to release the data at a later date after the conclusion of the project; in the meantime, please send an email to the first author: konlakorn.won@mahidol.edu; we can provide the data on a case-by-case basis.

Acknowledgments: We would like to thank Thammasat University Research fund and Virach Sornlertlamvanich who is the head of the "AI Ready City Networking in RUN" which this research is under. Also special thank to Thailand Mental Health Technology and Innovation Center for providing data for this research.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SER	Speech emotion recognition
CTR	Convolution time reduction
LMET	Linear mean encoding transformation
Emo-DB	Berlin German Dataset
E2ESER-CD	Real-time End-to-End Speech Emotion Recognition from Cross-Domain
VTLP	Vocal tract length perturbation
VAD	Voice activity detection
WER	Word error rate
WA	Weighted accuracy
UA	Unweighted accuracy

Appendix A. Details of Common Voice 7.0 German and Thai

The details of the train/validation examples from the Common Voice 7.0 dataset that was used to fine-tune ASR can be found in the following table. Please note that the table does not include the "reported" and "invalidated" splits, as we did not use them.

For training, the splits used were the train+validated+other splits, while for validation, the dev+test splits were used.

Table A1. Number of examples in Common Voice 7.0 German and Thai.

Language\Data Split	Dev	Other	Test	Train	Validated
German (de)	15,907	8836	15,907	360,664	684,794
Thai (th)	9712	90,315	9712	23,332	107,747

References

1. Singkul, S.; Woraratpanya, K. Vector Learning Representation for Generalized Speech Emotion Recognition. *Heliyon* **2022**, *8*, e09196. [CrossRef]
2. Zhao, J.; Mao, X.; Chen, L. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed. Signal Process. Control* **2019**, *47*, 312–323.
3. Singkul, S.; Chatchaisathaporn, T.; Suntisrivaraporn, B.; Woraratpanya, K. Deep Residual Local Feature Learning for Speech Emotion Recognition. In *Neural Information Processing*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; pp. 241–252. [CrossRef]
4. Lech, M.; Stolar, M.; Best, C.; Bolia, R. Real-time speech emotion recognition using a pre-trained image classification network: Effects of bandwidth reduction and companding. *Front. Comput. Sci.* **2020**, *2*, 14. [CrossRef]
5. Protopapas, A.; Lieberman, P. Fundamental frequency of phonation and perceived emotional stress. *J. Acoust. Soc. Am.* **1997**, *101*, 2267–2277. [CrossRef] [PubMed]
6. Lee, S.; Bresch, E.; Adams, J.; Kazemzadeh, A.; Narayanan, S. A study of emotional speech articulation using a fast magnetic resonance imaging technique. In Proceedings of the Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, 17–21 September 2006.
7. Samantaray, A.K.; Mahapatra, K.; Kabi, B.; Routray, A. A novel approach of speech emotion recognition with prosody, quality and derived features using SVM classifier for a class of North-Eastern Languages. In Proceedings of the 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), Kolkata, India, 9–11 July 2015; pp. 372–377.
8. Wang, W.; Watters, P.A.; Cao, X.; Shen, L.; Li, B. Significance of phonological features in speech emotion recognition. *Int. J. Speech Technol.* **2020**, *23*, 633–642. [CrossRef]
9. Breitenstein, C.; Lancker, D.V.; Daum, I. The contribution of speech rate and pitch variation to the perception of vocal emotions in a German and an American sample. *Cogn. Emot.* **2001**, *15*, 57–79. [CrossRef]
10. Dieleman, S.; Schrauwen, B. End-to-end learning for music audio. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 6964–6968. [CrossRef]
11. Li, Y.; Zhao, T.; Kawahara, T. Improved End-to-End Speech Emotion Recognition Using Self Attention Mechanism and Multitask Learning. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019; pp. 2803–2807.
12. Yuenyong, S.; Hnoohom, N.; Wongpatikaseree, K.; Singkul, S. Real-Time Thai Speech Emotion Recognition with Speech Enhancement Using Time-Domain Contrastive Predictive Coding and Conv-Tasnet. In Proceedings of the 2022 7th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 19–20 May 2022; pp. 78–83.
13. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12449–12460.
14. Conneau, A.; Baevski, A.; Collobert, R.; Mohamed, A.; Auli, M. Unsupervised cross-lingual representation learning for speech recognition. *arXiv* **2020**, arXiv:2006.13979.
15. Soekhoe, D.; Van Der Putten, P.; Plaat, A. On the impact of data set size in transfer learning using deep neural networks. In Proceedings of the International Symposium on Intelligent Data Analysis, Stockholm, Sweden, 13–15 October 2016; Springer: Cham, Switzerland, 2016; pp. 50–60.
16. Singkul, S.; Khampingyot, B.; Maharattamalai, N.; Taerungruang, S.; Chalothorn, T. Parsing Thai Social Data: A New Challenge for Thai NLP. In Proceedings of the 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Chiang Mai, Thailand, 7–9 November 2019; pp. 1–7.
17. Singkul, S.; Woraratpanya, K. Thai Dependency Parsing with Character Embedding. In Proceedings of the 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE), Pattaya, Thailand, 10–11 October 2019; pp. 1–5.
18. Chaksangchaichot, C. Vistec-AIS Speech Emotion Recognition. 2021. Available online: https://github.com/vistec-AI/vistec-ser (accessed on 1 November 2021).
19. Burkhardt, F.; Paeschke, A.; Rolfes, M.; Sendlmeier, W.F.; Weiss, B. A database of German emotional speech. In Proceedings of the Ninth European Conference on Speech Communication and Technology, Lisbon, Portugal, 4–8 September 2005.
20. Farooq, M.; Hussain, F.; Baloch, N.K.; Raja, F.R.; Yu, H.; Zikria, Y.B. Impact of Feature Selection Algorithm on Speech Emotion Recognition Using Deep Convolutional Neural Network. *Sensors* **2020**, *20*, 6008. [CrossRef] [PubMed]
21. El Ayadi, M.; Kamel, M.S.; Karray, F. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognit.* **2011**, *44*, 572–587. [CrossRef]

22. Anagnostopoulos, C.N.; Iliou, T.; Giannoukos, I. Features and classifiers for emotion recognition from speech: A survey from 2000 to 2011. *Artif. Intell. Rev.* **2015**, *43*, 155–177. [CrossRef]
23. Shaneh, M.; Taheri, A. Voice Command Recognition System Based on MFCC and VQ Algorithms. *Int. J. Comput. Inf. Eng.* **2009**, *3*, 2231–2235.
24. Xu, M.; Zhang, F.; Cui, X.; Zhang, W. Speech Emotion Recognition with Multiscale Area Attention and Data Augmentation. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 6319–6323.
25. Kim, C.; Shin, M.; Garg, A.; Gowda, D. Improved Vocal Tract Length Perturbation for a State-of-the-Art End-to-End Speech Recognition System. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019; pp. 739–743.
26. Venkataramanan, K.; Rajamohan, H.R. Emotion Recognition from Speech. *arXiv* **2019**, arXiv:1912.10458.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
29. Shanahan, T. Everything You Wanted to Know about Repeated Reading. Reading Rockets. 2017. Available online: https://www.readingrockets.org/blogs/shanahan-literacy/everything-you-wanted-know-about-repeated-reading (accessed on 10 December 2021).
30. Team, S. Silero VAD: Pre-Trained Enterprise-Grade Voice Activity Detector (VAD), Number Detector and Language Classifier. 2021. Available online: https://github.com/snakers4/silero-vad (accessed on 2 March 2022).
31. Jaitly, N.; Hinton, G.E. Vocal tract length perturbation (VTLP) improves speech recognition. In Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language, Atlanta, GA, USA, 16 June 2013; Volume 117, p. 21.
32. Sefara, T.J. The effects of normalisation methods on speech emotion recognition. In Proceedings of the 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Vanderbijlpark, South Africa, 21–22 November 2019; pp. 1–8.
33. Markitantov, M. Transfer Learning in Speaker's Age and Gender Recognition. In *Speech and Computer*; Karpov, A., Potapova, R., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 326–335.
34. Limkonchotiwat, P.; Phatthiyaphaibun, W.; Sarwar, R.; Chuangsuwanich, E.; Nutanong, S. Domain Adaptation of Thai Word Segmentation Models using Stacked Ensemble. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; Association for Computational Linguistics: Online, 2020; pp. 3841–3847. [CrossRef]
35. Ardila, R.; Branson, M.; Davis, K.; Kohler, M.; Meyer, J.; Henretty, M.; Morais, R.; Saunders, L.; Tyers, F.; Weber, G. Common Voice: A Massively-Multilingual Speech Corpus. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 4218–4222.
36. Lee, J.; Tashev, I. High-level feature representation using recurrent neural network for speech emotion recognition. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.K.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
38. Schober, P.; Boer, C.; Schwarte, L.A. Correlation coefficients: Appropriate use and interpretation. *Anesth. Analg.* **2018**, *126*, 1763–1768. [CrossRef] [PubMed]