Improved Spoken Emotion Recognition With Combined Segment-Based Processing And Triplet Loss

Dejan Porjazovski and **Tamás Grósz** and **Mikko Kurimo**
Department of lnformation and Communications Engineering
Aalto University, Espoo, Finland
dejan.porjazovski@aalto.fi

Abstract

Traditional spoken emotion recognition solutions often process entire utterances all at once, ignoring the emotional variability within the speech. This shortcoming, especially plaguing end-to-end models, prompted us to investigate a segment-based technique processing only short parts of the audio, improving the recognition accuracy across three diverse emotion datasets. Furthermore, we employed a triplet loss to increase inter-class separability, demonstrating that combining it effectively with segment-based processing within our multi-task learning framework leads to improvements on both English and Finnish datasets. Our proposed method achieves 8.1% unweighted average recall improvement over the baseline on the IEMOCAP, 12% on the RAVDESS, and 7.2% on the FESC dataset. The results also indicate that vocalised emotions are strongly concentrated in short segments of speech, and new methods are needed to exploit this fact.

1 Introduction

In the age of digital transformation, the significance of human-computer interaction (HCI) systems becomes crucial. However, current HCI solutions struggle to comprehend emotions, a critical aspect of tasks like automated analysis of customer feedback. Incorrectly categorising emotions in such analyses could lead to misunderstandings, where complaints might be mistaken for positive feedback and vice versa. Therefore, the integration of an accurate spoken emotion recognition (SER) system within HCI applications holds vital importance in enhancing user experiences (Brave and Nass, 2007).

With the emergence of the Transformer architecture (Vaswani et al., 2017), pre-trained self-supervised models have gained popularity, particularly for tasks with limited data (Grósz et al., 2022). One popular audio-based foundation model is wav2vec2 (Baevski et al., 2020), which has already proven successful in SER applications. In a previous study, the authors utilised a pre-trained wav2vec2 model to extract embeddings from multiple layers, subsequently employing these embeddings as input for a neural network classifier (Pepino et al., 2021). Besides serving as feature extractors, these pre-trained models can also be fine-tuned for the specific task at hand. A fine-tuned wav2vec2 approach was successfully applied for predicting emotional intensities (Porjazovski et al., 2023). In addition to fine-tuning, the researchers incorporated a pre-training stage for the wav2vec2 model, outperforming the other approaches (Chen and Rudnicky, 2023) on the IEMOCAP dataset (Busso et al., 2008).

Despite their popularity, the majority of SER solutions process the whole utterance at once to produce emotion labels. Processing long sequences can cause the model to learn unwanted correlations (Arjovsky et al., 2019). A common way to deal with lengthy audios is to process them in segments (Schuller and Rigoll, 2006; Chen and Rudnicky, 2023; Xia et al., 2021; Tzinis and Potamianos, 2017). We hypothesise that segment-based processing ensures that the model is aware of the varying emotional intensities within the sample, thus improving its accuracy. Moreover, by seeing short segments during training, the model can become more robust to variance in duration. As exact labels for each segment are unavailable, we assigned the same utterance label to all corresponding segments. While not perfect and acknowledging potential label variation across segments, this approach has still proven advantageous (Mao et al., 2020).

The second issue of SER is the limited nature of available data, often addressed by employing unsupervised learning. In a previous study, Trigeorgis et al. (2016) used contrastive predictive coding to learn audio representation in an unsuper-

vised way (Li et al., 2021). Similarly, contrastive loss was used to train a Siamese network (Bromley et al., 1993), which learned to extract discriminative audio features (Lian et al., 2018). Pre-trained transformer models such as wav2vec2 can also benefit from task-specific contrastive learning. In another study, the authors showed the benefits of the wav2vec2 model in combination with contrastive learning and data augmentation (Alaparthi et al., 2022). Closely related to contrastive learning is the triplet loss function (Schroff et al., 2015), which was shown to be beneficial in increasing the inter-class separability of emotions (Huang et al., 2018).

In contrast to prior methodologies that employ the contrastive or triplet loss function across entire utterances, our study introduces a multi-task framework. Here, the model concurrently learns to separate segments with different emotions while optimising the parameters for the SER task using negative log-likelihood loss. By simultaneously applying the loss function at both utterance and segment levels, our approach enables the model to concentrate on both local and global features within the utterances, helping the model understand how emotions change over time. To evaluate the effectiveness of our proposed methods, we conduct empirical experiments using the wav2vec2 model on three distinct datasets in both English and Finnish languages.

2 Datasets

The IEMOCAP dataset contains 12 hours of English speech, annotated with nine discrete emotions. To ensure consistency with prior research, we focused on neutral, sadness, happiness, and anger, omitting the unbalanced classes. To evaluate our solutions, we employed a five-fold cross-validation based on the five recording sessions, as used in other studies (Chen and Rudnicky, 2023; wen Yang et al., 2021).

The second English corpus, called RAVDESS (Livingstone and Russo, 2018), features 12 male and 12 female speakers expressing eight emotions: neutral, calm, happy, sad, angry, fearful, disgust, and surprise, through spoken and sung sentences. As official dataset splits are unavailable, we adopted the splits from Pepino et al. (2021). We merged the calm and neutral emotions and allocated speakers 1-20 for training, 21-22 for development, and 23-24 for testing.

The FESC dataset (Airas and Alku, 2006) comprises Finnish prose passages narrated by five male and four female actors, spanning five hours. The dataset contains annotations for neutral, sadness, joy, affection, and anger emotions. In our experiments, we prepared the data the same way as done by Vaaras et al. (2022) employing a leave-one-speaker-out cross-validation approach, with each fold featuring one speaker for testing, one for validation, and the rest for training. The monotonic character of Finnish, primarily resulting from minimal pitch variation and placement of stress on the first syllable of the words, poses unique difficulties for emotion recognition.

3 Methods

Our proposed model utilises a multi-task setting, optimising two objectives. The negative log-likelihood helps the model learn to successfully classify the emotions, while the triplet loss separates the utterances with different labels farther in the latent space. Moreover, to reduce the variance attributed to varying lengths, we additionally use segment-based processing within the triplet and negative log-likelihood losses. The proposed model is illustrated in Figure 1.

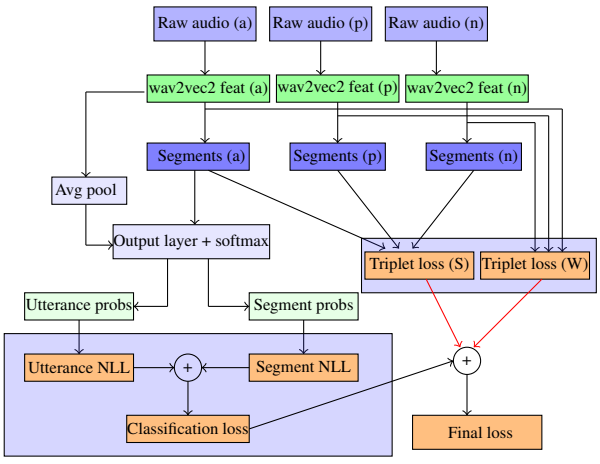


Figure 1: Architecture of the proposed model. "a" refers to the anchor, "p" to the positive, and "n" to the negative element. During training, we either use the triplet loss on the segments (S) or on the whole utterance (W)

3.1 Segment-based processing

To extract features from the raw audio sequence, we used a wav2vec2 model. The embedded data are of shape $X = (N, T, H)$, where N is the batch size, T is the temporal dimension, i.e. the timesteps, and H is the hidden size. As discussed earlier, the goal is to process the utterance in small segments.

We split each embedding vector X into segments with overlapping windows. Then, we average each segment along its temporal dimensions and pass it to a Linear layer, followed by a Softmax function, which produces class probabilities. This way, the model will generate label probabilities for each segment. In case the temporal dimension of the embedding vector X is smaller than the segment size, we process the whole sequence at once without splitting it. During training, we compute the loss over the whole sequence, as well as over each of the segments. In the inference stage, we obtain the label prediction by selecting the segment containing the highest probability.

3.2 Triplet loss

Our multi-task loss function is defined as:

$$L = L_{nll} + L_{tri} \quad (1)$$

where L_{nll} is the negative log-likelihood loss. The triplet loss function L_{tri} is calculated as:

$$L_{tri} = max(d(X_a, X_p) - d(X_a, X_n) + \lambda, 0) \quad (2)$$

where X_a is the anchor element, X_p is the positive element from the same class as X_a, and X_n is the negative element from a different class. The goal of the triplet loss function is to make the distance d between the elements of the same classes smaller than the distance between the elements of different classes. The distance d, in our study, is the L2 norm. λ is a margin determining the minimum distance between the positive (X_a, X_p) and the negative pairs (X_a, X_n). For choosing the negative sample, we ordered the samples by length and chose them to have a similar duration as the positive ones. This was done so that we would not compute the distance between a whole utterance that can not be split and a segment. There are other viable methods for selecting the negative sample, for instance, by picking one with a different valence or arousal; however, that is application-specific and we do not consider it in this study.

As discussed earlier, processing the utterance in segments can reduce the variance attributed to varying lengths. The triplet loss, on the other hand, helps with pulling the latent representation of samples with different classes farther from each other, ensuring easier separability. Therefore, we combined segment-based processing and triplet loss to utilise the benefits of both.

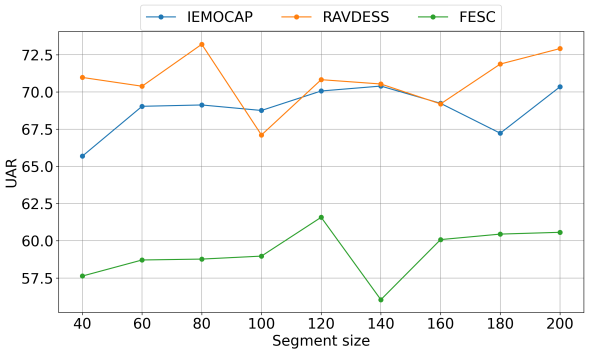


Figure 2: The effect of the segment size. The stride is half of the segment size. A segment size of 100 refers to roughly 200ms.

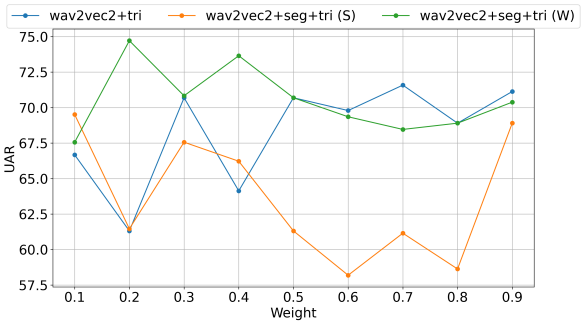


Figure 3: The effect of the weight α when combining the negative log-likelihood and triplet loss for the RAVDESS dataset.

To learn the SER task, we used the negative log-likelihood loss function on each segment, as well as the whole utterance:

$$L_{SER} = \sum_{i=1}^{N} \sum_{j=1}^{S} L_{nll}(i,j) + L_{nll}(i) \quad (3)$$

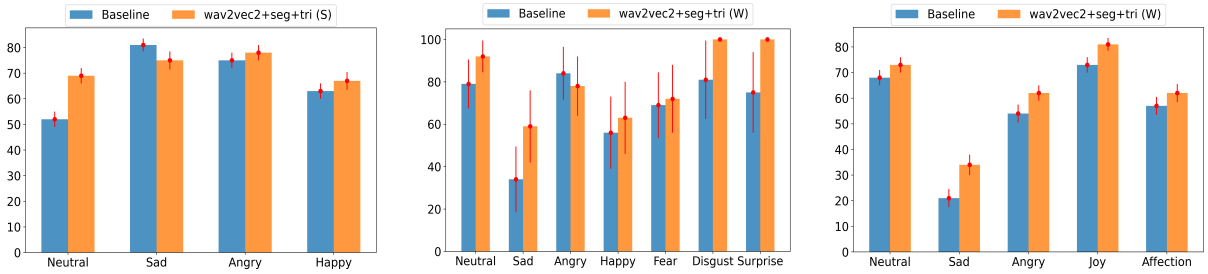
where N is the number of samples, i is the sample, S is the number of segments in sample i, j is the segment, and (i, j) represents the j-th segment in sample i.

In the experiments, we used either the segmented or the whole utterance in the triplet loss. In the end, we interpolated both loss functions.

4 Experiments

To extract features from the FESC utterances, we employed the multilingual pre-trained wav2vec2 model (Conneau et al., 2021), fine-tuned for ASR on Finnish data[1] (∼311M trainable parameters). In this study, we did not consider other self-supervised models, like HuBERT (Hsu et al., 2021) or WavLM (Chen et al., 2022), since they do not

[1] jonatasgrosman/wav2vec2-large-xlsr-53-finnish



(a) IEMOCAP (b) RAVDESS (c) FESC

Figure 4: UAR per class and the 95% confidence intervals (in red) for the baseline and the best model on each dataset.

have a Finnish variant. For the English experiments, we utilised the base wav2vec2 version[2] (∼90.2M trainable parameters), which is not fine-tuned on any specific task. The feature dimensions were set at 1024 for Finnish and 768 for English. We extracted English wav2vec2 features from the last Transformer layer, while for the Finnish version, we utilised layer 23 (out of 24), given that the final layer is typically optimised for the ASR task (Pepino et al., 2021). Even though performing a layer analysis and choosing the best-performing one can potentially improve the results, in this study, we focus on the architecture instead of specific hyperparameters.

To select the optimal segment and stride sizes, we tested the performance of the models with different values. The results of this experiment are shown in Figure 2. For the datasets where we employ cross-validation, we determined the best segment size on one fold. Based on the figure, on the IEMOCAP dataset, a segment size of 140 with a stride of 70 was chosen as the most optimal. For RAVDESS, smaller segment and stride sizes of 80 and 40 gave the best results, whereas, for FESC, a segment size of 120 with a stride of 60 performed the best.

The margin value λ in Equation 1 was set to 1 in all the experiments. When combining the negative log-likelihood and triplet loss functions, we did not use a weighting factor for IEMOCAP and FESC datasets. This decision was based on the high-performance variability between folds; the most optimal value for some splits resulted in poor outcomes for others. To address this, we attempted to set the weight as a learnable parameter, but this did not yield better results compared to not using any weighting.

[2]facebook/wav2vec2-base

For the RAVDESS dataset, since we did not use cross-validation, we conducted a weight analysis to determine the optimal value, as shown in Figure 3. We performed the weight analysis on a subset of the training set and determined the best weight based on the development set. To factor the weight when combining the loss functions as in Equation 1, we used:

$$L = (1 - \alpha) * L_{nll} + \alpha * L_{tri} \quad (4)$$

The weighting analysis revealed that the most optimal α value was 0.7 for the wav2vec2 model utilising the triplet loss, 0.1 for the model combining segmented processing and triplet loss on the segments (S), and 0.2 for the model using the triplet loss on the whole utterance (W). These findings revealed that when using segmented processing, it is better to give more weight to the negative log-likelihood loss, while when processing the whole utterance, it is better to give more priority to the triplet loss.

For optimisation, we used the Adam optimiser and trained the models for 30 epochs using a single V100 GPU. For the most complex model that uses segmented processing and triplet loss during training, the training time for one epoch with a batch size of 12 took around 34 minutes. During training, we kept the Convolutional Feature Encoder frozen while fine-tuning the Transformer layers. The complete code, along with a detailed list of hyperparameters, is publicly available[3].

5 Results

In this section, we compare the proposed techniques against the standard wav2vec2 pipeline, processing the whole utterance at once. We used unweighted average recall (UAR) as an evaluation

[3]Removed due to anonymity

Model	IEMOCAP	RAVDESS	FESC
wav2vec2 P-TAPT (Chen and Rudnicky, 2023)	(74.3)	/	/
wav2vec2+layer avg (Pepino et al., 2021)	67.2	84.3	/
wav2vec2 baseline	66.5 (65.6)	68.5 (67.8)	57.1 (60.5)
wav2vec2+seg	67.6 (66.7)	73.2 (72.1)	60.1 (62.1)
wav2vec2+tri(W)	73.6 (72.6)	80.4 (78.4)	61.0 (62.1)
wav2vec2+seg+tri(S)	**74.6 (73.9)**	79.0 (76.9))	63.6 (64.9)
wav2vec2+seg+tri(W)	73.9 (72.9)	**80.5 (78.4)**	**64.3 (65.0)**

Table 1: UAR and UA (given in the brackets) scores for the IEMOCAP, RAVDESS, and FESC test sets. (S) indicates that the triplet loss was calculated on the segments, while(W) indicates that it was done on the whole utterance.

metric. The UAR metric is calculated as a sum of the class-wise recall divided by the number of classes. For comparison with the previous state-of-the-art (SOTA) method on IEMOCAP, we additionally provide unweighted accuracy (UA) scores.

Table 1 presents the UAR and UA scores achieved on the IEMOCAP, RAVDESS, and FESC datasets. Looking at the IEMOCAP test results, we can notice that fine-tuning the wav2vec2 model with a classification layer already yields good performance. Segment-based processing, which involves splitting the utterances into segments and processing them individually, slightly improves the results over the wav2vec2 baseline.

To explore the impact of segment-based processing on recognising sequences of varying lengths, we divided the test set into segments below and over 10 seconds in duration. For short utterances, the baseline achieved a recognition rate of 65.9%, whereas segment-based processing notably improved the performance to 67.2% UAR, underscoring its efficacy for shorter utterances. Conversely, for utterances longer than 10 seconds, segment-based processing got slightly inferior results with 69.7% UAR, compared to the baseline's 70.6%. These findings highlight the nuanced effect of segment-based processing, demonstrating its effectiveness for short sequences while indicating the need for further optimisation or alternative approaches for longer ones.

Introducing the triplet loss, combined with negative-log-likelihood, gives a further improvement of 7.1% UAR score over the wav2vec2 baseline. Furthermore, we observed additional improvement by combining segment-based processing and triplet loss. When using the segments in the triplet loss, the model got a 74.6% UAR score.

On the RAVDESS test set, the segment-based processing achieved a UAR score of 73.2%, considerably better than the baseline of 68.5%. Using the triplet loss further enhances the results over solely using the segmented processing. The multi-task learning approach produces the best UAR score of 80.5%, this time by using the whole utterance in the triplet loss. Since the lengths of the utterances in this dataset are short, we could not assess the performance of the segment-based model on short and long samples.

The Finnish experiments follow a similar trend, where the segment-based processing outperforms the baseline. To examine the impact of segment-based processing on utterance length, we partitioned the test set into segments shorter and longer than 10 seconds, mirroring our approach in the IEMOCAP dataset. Notably, this analysis revealed enhancements in recognition performance for both short and long utterances through segmented processing. Specifically, for short utterances, the baseline wav2vec2 model achieved a UAR score of 57%, while segment-based processing improved it to 59.3%. For longer segments, the difference is more pronounced, with the baseline yielding a UAR score of 69%, contrasted with 72.7% for segment-based processing.

Adding the triplet loss further improves the results, achieving a UAR score of 61%. In the multi-task scenario, employing the triplet loss on entire utterances rather than segments gives better results, as seen from Table 1.

The superior performance of our proposed multi-task model comes at a cost of increased computational time. For instance, to evaluate one split of the IEMOCAP test set, the baseline model took 30 seconds using a batch size of 1, whereas the wav2vec2+seg+tri(S) took 41 seconds.

Compared to SOTA results on the IEMOCAP dataset, our multi-task model using triplet loss, in combination with segmented processing, achieves

Model	IEMOCAP	RAVDESS	FESC
wav2vec2 baseline	75.1	78.3	71.5
wav2vec2+seg	74.5	78.2	72.8
wav2vec2+tri(W)	82.8	86.8	80.2
wav2vec2+seg+tri(S)	/	89.7	79.7
wav2vec2+seg+tri(W)	83.0	/	/

Table 2: Model agreement in terms of UAR, where the best model's predictions for each dataset are treated as ground truth.

a slightly worse UA score than the P-TAPT, which modifies the pre-training stage of the wav2vec2 model to generate emotion-specific features. On the RAVDESS dataset, the SOTA results incorporate a weighted average of all wav2vec2 layers, whereas we only utilise the output of the last Transformer layer. Exploring multiple layers or selecting the best layer for the task could potentially improve the results, but this falls beyond the scope of our study. Additionally, that approach performs the best on the RAVDESS dataset, but its performance drops on IEMOCAP, indicating that it is not robust enough. For the Finnish FESC dataset, we could not find a suitable benchmark.

To get a better understanding of the improvements gained from the multi-task model, we plotted the UAR per class for the baseline and the best-performing model on each dataset, shown in Figure 4. Upon examining the class-specific performances of both models across various emotions, it becomes evident that the multi-task approach almost always achieves superior recognition rates. Notably, exceptions include the recognition of sadness in the IEMOCAP dataset and anger in the RAVDESS dataset. A plausible explanation for the diminished performance in recognising the sad emotion might stem from its extended average duration. As previously discussed, the model demonstrates a slight decline in performance when processing long utterances within the IEMOCAP dataset.

To test the stability of the models, we calculated the 95% confidence intervals for the best models on each dataset, using the bootstrapping method. The confidence intervals are presented in Figure 4. For the RAVDESS dataset, we observed a large interval which contains the real performance with a 95% chance. These findings indicate that there is a high variability between the utterances for some of the emotions. Moreover, our model tends to be more stable with less variability for disgust, surprise and neutral emotions in comparison to the baseline. Nevertheless, the performance per class is in the middle of the confidence intervals, meaning that the overall performance is not distorted by some extremely easy or difficult test samples.

In the last set of experiments, we test how much the models differ in the predictions. To achieve that, we calculated the model agreement, where we treated the best-performing model's predictions as ground truth and evaluated it against the other models. The results of this experiment are presented in Table 2.

On the IEMOCAP and RAVDESS datasets, the best-performing model has the biggest agreement with its similar counterpart (wav2vec2+seg+tri(W) for IEMOCAP and wav2vec2+seg+tri(S) for RAVDESS), followed by the model just utilising the triplet loss. On the FESC dataset there is a higher agreement between the best-performing model and the one that only incorporates triplet loss, even though that model falls behind in terms of UAR, compared to both multi-task approaches. These results indicate that the mistakes that both the wav2vec2+seg+tri(S) and wav2vec2+seg+tri(W) models make differ from each other, suggesting that they learn different things when using segments or whole utterances in the triplet loss.

6 Conclusion

In this work, we investigated segment-based processing, triplet loss, and a multi-task combination of both techniques for SER. The results from our English and Finnish experiments demonstrated the effectiveness of segment-based processing compared to the conventional approach of processing the entire utterance at once. Moreover, we showed that the segment size plays an important role and should be chosen carefully. By integrating the triplet loss into the learning framework, we observed considerable performance improvements across all datasets, surpassing the segment-based processing and showing the benefits of separating

the different classes in the latent space. On all three corpora, the multi-task approach of combining the segmented processing and triplet loss gave the best results. Furthermore, we showed that segment-based processing improves the model's robustness on short utterances, whereas for long ones, there is a performance drop on the IEMOCAP, but an improvement on FESC. By comparing the model agreement, we found that using segments or whole utterances in the triplet loss can lead to the models learning different things, making their predictions differ.

Limitations

While the proposed multi-task approach significantly outperforms the baseline, it comes with increased computational demands. Further enhancements can be achieved by combining multiple Transformer layers, as demonstrated in (Pepino et al., 2021). However, this study omits layer experimentation to prioritise architectural analysis over hyperparameter tuning. Additionally, not incorporating a weighting factor for IEMOCAP and FESC when combining the loss functions adds a limitation, which remains an important future task.

Ethics Statement

The use of emotion recognition in certain applications can lead to human rights violations. The EU AI act (EU, 2024) has classified emotion recognition systems as a high-risk application, meaning that the users need to be informed if such a system is being put into place. Moreover, relying on automatic emotion recognition systems to determine the state of a person can be dangerous, especially when that person is in shock and may not express the actual emotions. Furthermore, emotional expression varies significantly from person to person. For instance the emotional expression in children with autism differs from typically developing children (Chaidi and Drigas, 2020). While the development of emotion recognition technology may offer societal benefits, it is essential to carefully consider who the primary users are and how they will be affected.

Acknowledgements

The computational resources were provided by Aalto ScienceIT. We are grateful for the Business Finland project LAREINA under Grant 7817/31/2022.

References

Matti Airas and Paavo Alku. 2006. Emotions in vowel segments of continuous speech: analysis of the glottal flow using the normalised amplitude quotient. *Phonetica*, 63(1):26–46.

Varun Sai Alaparthi, Tejeswara Reddy Pasam, Deepak Abhiram Inagandla, Jay Prakash, and Pramod Kumar Singh. 2022. Scser: Supervised contrastive learning for speech emotion recognition using transformers. In *2022 15th international conference on human system interaction (HSI)*, pages 1–7. IEEE.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, et al. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, et al. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Scott Brave and Cliff Nass. 2007. Emotion in human-computer interaction. In *The human-computer interaction handbook*, pages 103–118. CRC Press.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, et al. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Irene Chaidi and Athanasios Drigas. 2020. Autism, expression, and understanding of emotions: literature review.

Li-Wei Chen and Alexander Rudnicky. 2023. Exploring wav2vec 2.0 fine tuning for improved speech emotion recognition. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised Cross-Lingual Representation Learning for Speech Recognition. In *Proc. Interspeech 2021*, pages 2426–2430.

EU. 2024. Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts.

https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CONSIL%3AST_7536_2024_INIT&qid=1716543737061.

Tamás Grósz, Dejan Porjazovski, Yaroslav Getman, et al. 2022. Wav2vec2-based paralinguistic systems to recognise vocalised emotions and stuttering. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 7026–7029.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, et al. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Jian Huang, Ya Li, Jianhua Tao, et al. 2018. Speech emotion recognition from variable-length inputs with triplet loss function. In *Interspeech*, pages 3673–3677.

Mao Li, Bo Yang, Joshua Levy, et al. 2021. Contrastive unsupervised learning for speech emotion recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6329–6333. IEEE.

Zheng Lian, Ya Li, Jianhua Tao, et al. 2018. Speech emotion recognition via contrastive loss under siamese networks. In *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and First Multi-Modal Affective Computing of Large-Scale Multimedia Data*, pages 21–26.

Steven R Livingstone and Frank A Russo. 2018. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391.

Shuiyang Mao, P.C. Ching, C.-C. Jay Kuo, et al. 2020. Advancing Multiple Instance Learning with Attention Modeling for Categorical Speech Emotion Recognition. In *Proc. Interspeech 2020*, pages 2357–2361.

Leonardo Pepino, Pablo Riera, and Luciana Ferrer. 2021. Emotion Recognition from Speech Using wav2vec 2.0 Embeddings. In *Proc. Interspeech 2021*, pages 3400–3404.

Dejan Porjazovski, Yaroslav Getman, Tamás Grósz, and Mikko Kurimo. 2023. Advancing audio emotion and intent recognition with large pre-trained models and bayesian inference. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9477–9481.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Björn Schuller and Gerhard Rigoll. 2006. Timing levels in segment-based speech emotion recognition. In *Proc. Interspeech 2006*, pages paper 1695–Wed2BuP.8.

George Trigeorgis, Fabien Ringeval, Raymond Brueckner, et al. 2016. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5200–5204. IEEE.

Efthymios Tzinis and Alexandras Potamianos. 2017. Segment-based speech emotion recognition using recurrent neural networks. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 190–195. IEEE.

Einari Vaaras, Manu Airaksinen, and Okko Räsänen. 2022. Analysis of Self-Supervised Learning and Dimensionality Reduction Methods in Clustering-Based Active Learning for Speech Emotion Recognition. In *Proc. Interspeech 2022*, pages 1143–1147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, et al. 2021. SUPERB: Speech Processing Universal PERformance Benchmark. In *Proc. Interspeech 2021*, pages 1194–1198.

Yangyang Xia, Li-Wei Chen, Alexander Rudnicky, et al. 2021. Temporal context in speech emotion recognition. In *Interspeech*, volume 2021, pages 3370–3374.