Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.079697

ARTICLE

Real-Time Emotion Recognition System Using Adaptive Distillation Technique

Mustaqeem Khan[1], Ufaq Khan[2], Mamoun Awad[1], Nazar Zaki[1], Guiyoung Son[3] and Soonil Kwon[3,*]

[1]College of Information Technology, United Arab Emirates University, Al Ain, United Arab Emirates
[2]College of Computer Vision, Mohamed Bin Zayed University of AI, Abu Dhabi, United Arab Emirates
[3]Interaction Technology Laboratory, Sejong University, Seoul, Republic of Korea

*Corresponding Author: Soonil Kwon. Email: skwon@sejong.edu

Received: 26 January 2026; Accepted: 01 April 2026; Published: 27 April 2026

ABSTRACT: Knowledge distillation has shown impressive results in different fields, including detection, recognition, and generation. These models are excellent at tasks such as speech recognition, but they need to be shrunk down using adaptive knowledge distillation (*AKD*). The use of *AKD* can improve human-computer interactions and streamline data collection in the field of Speech Emotion Recognition (*SER*). This study presents a high-level approach that employs a novel adaptive knowledge distillation (*AKD*) with spatio-temporal transformers to acquire advanced semantic features from the input signal. This method uses an instance-by-instance correlation between the teacher and a student to determine the teacher's importance. Additionally, this work proposes a knowledge-transfer strategy to integrate soft targets between teachers and students, aiming to provide deeper insight for the final prediction. Our light-weight model *AKD* is an efficient solution for edge devices and learns the synergistic information for respective tasks, as discussed in the results and analysis section. Our proposed model *AKD* outperforms the *SOTA* models of SER systems on the benchmark datasets, IEMOCAP, EmoDB, and RAVDESS, with an absolute gain of 4%–6% in overall recognition rate.

KEYWORDS: Affective computing; edge electronics; emotion recognition; knowledge distillation; speech signal

1 Introduction

Emotion recognition involves detecting the intention, feelings, and attitude of the speakers and is an indispensable part of human communication. Speech emotion recognition (*SER*) has recently become a focus of research due to its applications in areas such as human-computer interaction, mental health diagnosis, customer service, clever call centers, and online learning [1]. More and more deep learning algorithms have been developed to address the *SER* problem to find various patterns and relationships in the speech data, including convolutional neural networks (*CNN*), recurrent neural networks (*RNN*) and long short-term memory (*LSTM*) [2].

Recently, the modern machine learning algorithms and the prevalence of smartphones [3], edge devices like smartphones and IoT (Internet of Things) devices can use built-in sensors like cameras, microphones, or heartbeat sensors to identify user emotions [4]. Subsequently, algorithms are trained to recognize and categorize emotions using facial expressions, voice patterns, and physiological responses, thus making them adaptable for improving human-computer interaction, mental health diagnosis, and tailored marketing, among others [5]. Besides the difficulty of collecting emotion data, emotions are also difficult to categorize, as they are multi-faceted phenomena subject to cultural and individual differences. Since different cultures

and individuals express and perceive emotions differently, it can be difficult to use a universal standard [6] to categorize them.

By contrast, *KD* methods, that explicitly improve the model's ability to capture higher-level semantic cues from speech for both target and non-target classes [7], do not consider non-target class information in that phase. For example, the *KD* loss functions from [8] enable smaller subnetworks to learn from large "teacher" networks while maintaining similar performance. In these approaches, the authors employed the *KD* method in teacher-student networks during training and did not utilize any other methods in the inference. Furthermore, the *KD* method has also been proposed as a performance enhancer in other domains such as computer vision [9], natural language processing [10], recommendation [11], and other such related domains.

Furthermore, action localization [12] and emotion and identity recognition [13] for videos have also been addressed using transformer-based architectures. To address the limitations of these methods, in this paper, we propose the state-of-the-art Adaptive Knowledge Distillation (*AKD*) framework (see Fig. 1), which learns from multi-level knowledge distillation across three types of teacher models, including the high-level, intermediate-level, and soft-target ones. The authors also make use of weighted soft targets and a group hint approach for transferring the information from the teacher's last layers to the intermediate layers of the student. Implementing some of the principles of adaptive knowledge distillation, they propose a basic and effective framework with the goal to increase SER systems' performance and robustness. This is done through the technique of distilling knowledge, which distills and adapts knowledge from teacher models to improve an SER system's performance.

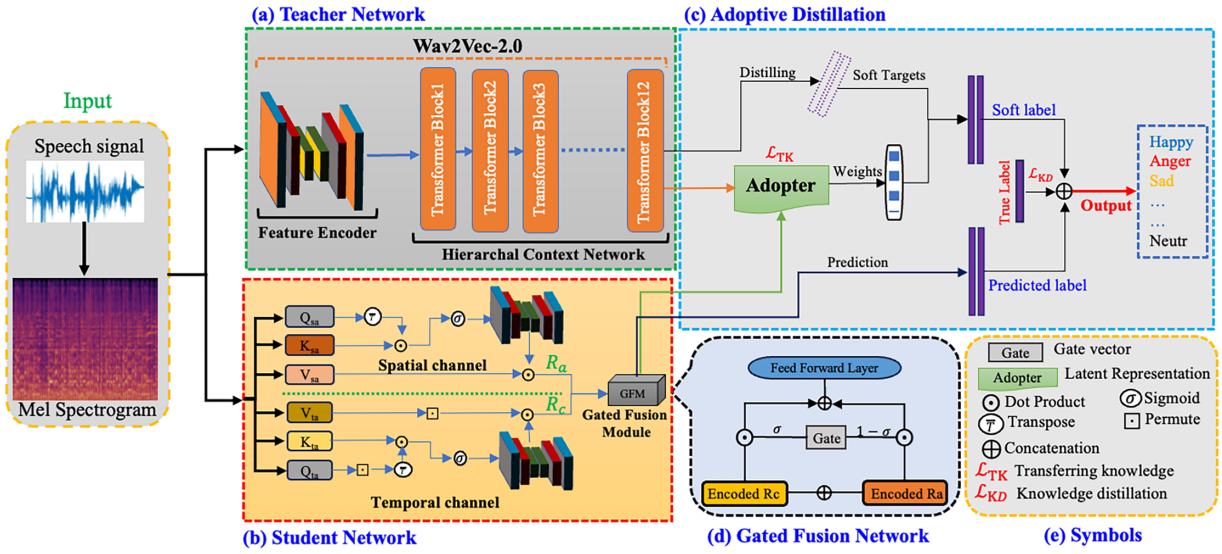

Figure 1: Overview of the designed model for emotion recognition with adaptive knowledge distillation using speech signal.

Inspired by the success of recursive attention architectures in other tasks [14], we include this strategy in our system to further improve the accuracy and generalization of the model. For improved adaptability, we adopt a spatiotemporal transformer in the student network and a hierarchical context-based transformer in the teacher models. This is accomplished through fused multi-head attention mechanisms, transferring knowledge via soft labels, aligning teacher-student logits, and enhancing feature discriminability.

The complete framework and our methodologies promise a significant leap forward in the development of advanced and lightweight *SER* systems that can be easily deployed on edge devices.

The main contributions of the paper are summarized as follows:

- The authors propose a lightweight affective model for edge devices to endorse a novel knowledge distillation approach incorporating an adaptive learning strategy. In addition, employ instance-level teacher importance weights to facilitate the transfer of intermediate-level knowledge to students.
- The authors introduce a novel method for enhancing emotion recognition through adaptive knowledge distillation, leveraging 'dark knowledge' to mitigate misclassifications of emotions and enhance recognition rates beyond current state-of-the-art techniques (See Section 4). To the best of our knowledge, this is the first use of adaptive knowledge distillation in speech-emotion recognition for edge devices.
- Our model leverages adaptive knowledge from the teacher network to guide the student using a single-level output built upon a lightweight spatio-temporal transformer architecture. The distilled student model demonstrates remarkable performance across three benchmark datasets: IEMOCAP, EmoDB, and RAVDESS, achieving a 4%–6% improved recognition rate, respectively.

The rest of this paper unfolds as follows. Related work is covered in Section 2. The proposed *AKD* for *SER* methodology is detailed in Section 3. Experimentation and ablation studies are presented in Section 4. Finally, conclusions and avenues for future research are outlined in Section 5.

2 Related Work

Foundation Models: Foundation models are expected to be a good initialization point for down-stream tasks. A new branch of research focuses on self-supervised learning-based adaptations of pre-trained huge models (pre-trained through unsupervised learning on large and unlabelled datasets). While these methods have shown strong results on the speech task [15–17], one potential answer, used by Wav2vec 2.0 [15], is to use product quantization for a specific function, along with a separate contrastive loss function, which is learned during pre-training to help the Transformer encoder identify the right quantized representation among the distractors.

Another successful model is HuBERT [16], which uses k-means clustering to group together representation vectors to form pseudo-class labels that can then be used in the training dataset. During pre-training, the model tries to predict the real class labels of both the masked tokens and the unmasked tokens. In this model, the features of one layer of the Transformer are extracted, and a second k-means clustering is used to refine the cluster. WavLM is based on the pseudo-labeling in pre-training from HuBERT, but it uses a broader pre-training dataset to improve generalization to new tasks [17]. In addition, the WavLM pre-training tasks also include speech denoising, where the model is trained to continue performing under the presence of noise and overlapping speech in the input. These tasks help to build better representations and more scalable models that can be transferred beyond automatic speech recognition tasks. This led to the emergence of foundation models as a new model in the space of speech processing models that achieved state-of-the-art performance.

Knowledge Distillation: Knowledge distillation (*KD*) is the task of training a smaller and more compact model called the student to mimic the function of a larger model called the teacher. The teacher is typically more powerful but also computationally expensive to evaluate. The teacher's knowledge is transferred to the student well, especially when the output is a distribution with probabilities (for instance the probability of words in a language task). Since these distributions are usually trained using KL-divergence, L1 or L2 losses are often used to match the teacher's internal representation or feature maps to the student's.

Various methods have been proposed [18] to ensure efficient learning while transferring complete knowledge, always by looking at the one most likely word sequence (out of the complex word sequence).

Since we only need to look at the most likely path, KD can be performed with a simplified loss function that is computationally faster with the trade-off of losing some information from the entire sequence. However, the complicated distribution sequence is not the only route for transferring knowledge and the authors of [19] did so differently. The authors use features extracted from a layer of the teacher model encoder to ease training. The latent features are of fixed size compared to the variable-length input audio, speeding up computation as they do not need to be continually computed from scratch. To further reduce the size of the information sent and avoid bottlenecks with the multicodebook vector quantization, the teacher model features are quantized from 32-bit floating-point representations to 8-bit integer values, which the student model tries to predict during training. This is likely to be more efficient, but may lead to a small drop in performance compared to plain L1 and L2 losses: reference [20] also uses a three-step approach where the non-streaming teacher model is not emitted until distillation, and uses a streaming model before distilling from the teacher.

3 Model Architecture

3.1 Overview

This section describes the training framework used in the proposed Adaptive Knowledge Distillation (AKD) model. The system consists of a pretrained wav2vec 2.0 teacher network and a lightweight student network. During training, the student network learns from both the ground-truth emotion labels and the soft targets produced by the teacher model. The overall training objective combines the standard cross-entropy loss with the adaptive distillation loss. During training, the teacher model remains fixed while the student model parameters are optimized using the combined loss function. For each input segment, the teacher produces soft targets that guide the student network. The student model is trained using the Adam optimizer until convergence.

Let x denote an input speech segment and y is corresponding emotion label. The teacher network produces logits z_t, while the student network produces logits z_s. The probability distributions obtained after the softmax operation are denoted by p_t and p_s, respectively. The temperature parameter T controls the softness of the probability distributions used during distillation. The standard cross-entropy loss is formulated as

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(p_{s,i}), \tag{1}$$

where C denotes the number of emotion classes. This loss measures the discrepancy between the student model's predicted emotion distribution and the ground-truth labels, and the knowledge distillation loss is defined as

$$L_{KD} = KL\left(p_t^T \parallel p_s^T\right), \tag{2}$$

where $KL(\cdot)$ denotes the Kullback–Leibler divergence between the softened teacher and student probability distributions. This loss enables the student model to learn informative knowledge from the teacher model. Finally, the model's overall training objective is expressed as

$$L = \alpha L_{CE} + \beta L_{KD}, \tag{3}$$

where α and β are balancing coefficients that control the relative contributions of the classification loss and the distillation loss. The detailed descriptions of each component are provided in the subsequent sections.

3.2 Knowledge Distillation

The authors introduced a relatively new technique for transferring knowledge according to the Kullback-Leibler divergence theory in [8]. This divergence measures results from differences between two probability distributions over the same variable and is minimized for the teacher-student model. This technique has been tested on speech and image recognition tasks, which have proven effective. This concept incorporates the distillation of knowledge *kd* into the probability equation, along with the classification probability $q = [q_1, q_2, \ldots, q_i, \ldots, q_c] \in \mathbb{R}^{1\times c}$. Here, q_i represents the probability of the i-th class, and c indicates the total number of classes with logit of the i-th class in this concept, which can be calculated as:

$$q_i = \frac{\exp(z_i/kd)}{\sum_{j=1}^{c}\exp(z_j/kd)} \tag{4}$$

where z_i represents the logit and distilled knowledge shown by *kd*, which is set to 1, and the label is classified one at a time. The disadvantage of this approach is that it makes neural network training too rigid, resulting in a loss of information about incorrect classes. As *kd* increases beyond 1, classes with a probability of 0 acquire a modest probability. However, logit distillation is limited despite this strategy. Furthermore, the teacher network is based on a pretrained wav2vec 2.0 model trained on large-scale speech corpora such as LibriSpeech. None of the evaluation datasets used in this study (IEMOCAP, EmoDB, and RAVDESS) is included in the pretraining data, ensuring that the reported results are not influenced by data overlap between pretraining and evaluation datasets.

3.3 Teacher & Student Models

Teacher: The method employs a pre-trained wav2vec-2.0 (Large) [15] model to encode the audio waveform using the feature encoder, capturing the low-level embedding features of the input waveform. The resulting embedding is normalized and activated with the Gelu function before being fed into the context network, comprising 12 transformer blocks with 12 attention heads each. Afterward, the soft label and weight for emotion classification prediction are determined from the feature vector obtained from the last layer of the context network.

Student: Our proposed student is based on the hybrid transformer [21], which incorporates skip connections between encoders and decoders. In the case of smaller datasets, the authors observed that employing knowledge adoption and distillation yields improved results. Following this insight, the authors constructed a student model (shown in Fig. 1) using an encoder-decoder architecture with a residual learning strategy, followed by convolutional layers with skip connections across layers [22]. In our method, the Transformer model takes the lead in analyzing speech and is encoded with a self-attention mechanism that analyzes these features, focusing on essential parts of the speech and considering long-range dependencies. This improves understanding compared to traditional methods. Finally, the system decodes the encoded features, enabling the model to consider all parts of the speech signal simultaneously and potentially improving its ability to capture complex relationships within the data, which is crucial for extracting discriminative features. The outputs undergo post-processing using connected components and employ spatial and temporal learning to target emotion effectively, capturing a comprehensive learning pattern from a micro perspective.

Limitations: The traditional transformer approach uses arbitrary convolutions for volumetric input data. However, these convolutions can only capture short-range spatial-temporal features, limiting their capacity to model broader global contextual dependencies beyond the designated receptive field. The spatial and temporal channels of the Transformers encode long-range dependencies by comparing feature activations throughout space and time. This mechanism transcends the limitations of conventional filters'

receptive fields. However, combining self-attention with convolutional layers proves advantageous for various tasks [23]. However, the authors are unaware of prior attempts to design spatio-temporal self-attention exclusively as an essential component for SER, as described in the literature.

3.4 Proposed Adaptive Knowledge Distillation

To capture the inherent qualities of teacher networks, the authors developed Adopter (as shown in Fig. 1) as a latent representation of teacher networks. This approach draws inspiration from latent factor models frequently employed in recommendation systems (as discussed in [24]). A latent factor represents their inherent characteristics in the model. Our approach extracts instance representations from the final layer of the student network's output. This results in a value of the input, where the number of channels, height, and width of the student's feature map ensure consistent alignment of input representations; the authors employ an essential operation to select the most significant value within each channel, as demonstrated below:

$$d_i = \mathrm{Conv}(b_i, s) \tag{5}$$

The method has a set of values represented by the variable d_i, where each value is a vector in $\mathbb{R}^c$. Similarly, $b_i \in \mathbb{R}^{c\times h\times w}$ is a result for the i-th input, where c, h, and w represent channels, height, and width of the feature map. For simplicity, set d_i (a vector) equal to c (the number of channels), compute the weight for the i-th input attributed to the teacher model, and normalize it using the loss function to ensure fairness.

$$w_{t;i} = \mathrm{loss}(c_{t;i}) = \frac{\exp(c_{t;i})}{\sum_{t'=1}^{m}\exp(c_{t';i})} \tag{6}$$

Eq. (6) defines a loss function for a model's predictions, particularly useful in scenarios with multiple possible outputs at each step. Here, $w_{t;i}$ represents the loss assigned to a specific element (i) at a particular time (t). The model first calculates a score $(c_{t;i})$ for this element. The higher the score, the more likely the model is to believe this element is the correct output. An exponential term $(\exp(c_{t;i}))$ emphasizes this preference. To create a proper probability distribution, the exponentials of all scores across all elements and timesteps are then summed. Finally, dividing the element's own exponential by this total sum gives us the loss $(w_{t;i})$. This loss is likely used during training to adjust the model and improve its ability to identify the most probable output at each step. Furthermore, the weighted addition operation to obtain the integrated soft-target $\tilde{y}_T^i$, contrasting with traditional learning methods, is calculated as:

$$\tilde{y}_T^i = \sum_{t=1}^{m} w_{t;i}\cdot \tilde{y}_T^{t;i} \tag{7}$$

$$L_{KD} = \sum_{i=1}^{N}\left(H(y_i, y_{S,i}) + k\cdot D_{KL}(\tilde{y}_T^i, \tilde{y}_S^i)\right) \tag{8}$$

$$D(x_i, x_j, x_k) = \cos^{-1}\left(\frac{x_i - x_j}{x_i - x_j}\cdot\frac{x_i - x_k}{x_i - x_k}\right) \tag{9}$$

The variable $\tilde{y}_T^{t;i}$ represents the soft-target generated by the t-th teacher for the i-th input. The system guided the students through two methods: (i) loss of distillation of standard knowledge, incorporating the teacher model's knowledge according to Eq. (8). This equation involves two main components: $\tilde{y}_S^i$ and $\tilde{y}_T^i$ (student and teacher), respectively. The former is the soft-target of the i-th input produced by the student network, while the latter is the integrated soft-target computed using Eq. (7). Furthermore, facilitate

the transfer of relational information across different datasets through structural knowledge, implemented as Eq. (9), inspired by [25]. Furthermore, represent the loss for the student-teacher scenario by obtaining the integrated soft targets. This approach will ensure accurate and efficient learning, leading to better outcomes, which are computed as follows:

$$L = \sum_{i,j,k} l_d D(\tilde{y}_T^i, \tilde{y}_T^j, \tilde{y}_T^k) \cdot D(\tilde{y}_S^i, \tilde{y}_S^j, \tilde{y}_S^k) \tag{10}$$

$$L_{KD} = \sum_{t=m}^{1} \sum_{l=f(g)} k u_t - s_t(v_l)_2 \tag{11}$$

Our Adaptive Knowledge Distillation (*AKD*) framework prioritizes robust learning, particularly when the student model learns from the teacher's outputs. In order to prevent a student regression from being negatively impacted by the noise from their teacher's predictions, the authors replace the original regression loss function (MSE, or Mean Squared Error) with the **Huber loss (denoted as l_d)**. The reason is that MSE is not strong to outliers in the same way that regression problems with MAE are. Huber loss is MSE for small errors and MAE for large errors, improving the student regression's robustness.

In the core *AKD* process, the students are trained using the "soft targets", i.e., the probabilities of the teacher model. The soft targets $\tilde{y}_T^i$, $\tilde{y}_T^j$, and $\tilde{y}_T^k$ are computed using the three inputs or the inputs in the complex example (for all three inputs). The soft targets provide the students with the opportunity to exploit the additional knowledge learned by the teacher model. The computation of these targets, as well as their use in the distillation loss l_d is defined in Eq. (10).

Furthermore, as shown in Eq. (11), our knowledge transfer mechanism helps to close the divide not only at the final output. As u_t is the last rich feature map of a teacher model and v_l is the l-th layer of a student model, Eq. (11) is probably used in a simple comparison process (a projection and a distance/similarity computation, for example) or a dynamic weighting based on how much do u_t and v_l agree with each other. This mechanism is then used to *dynamically weigh* the loss term of the main distillation term (l_d in Eq. (10)). This is based on the idea that if the student's intermediate characteristics v_l are already close to the teacher's characteristics u_t, the distillation weight at the output level (l_d) could be relaxed to allow student characteristics across the architecture to be more expressive.

A critical challenge in *KD* is "negative transfer," in which a student model inadvertently learns incorrect patterns from a faulty teacher. To mitigate this, introduce a mechanism to selectively allow knowledge transfer based on the teacher's reliability for a given instance. This is governed by Eqs. (12) and (13):

$$\gamma = 1 - \rho^c_{\text{teacher}} \tag{12}$$

Here, ρ^c_{teacher} represents a measure of the teacher's confidence or correctness for a specific class c or, more generally, its prediction quality on a given input instance (often determined by comparing the teacher's prediction to the ground truth). Consequently, γ quantifies the teacher's *error* or *uncertainty*; a low γ indicates a reliable teacher prediction for that instance.

$$\beta = \begin{cases} \gamma, & \text{if } \gamma \leq \theta \\ 1, & \text{otherwise} \end{cases} \tag{13}$$

The variable β acts as a modulating factor, determined by γ and a predefined threshold θ. This threshold θ defines an acceptable level of teacher error.

- If the teacher's error γ is within an acceptable range (i.e., $\gamma \leq \theta$), then β is set to γ. This means the influence of certain loss components (like L_{ccc} in Eq. (14)) will be scaled by the teacher's error.
- If the teacher's error γ exceeds this threshold (i.e., $\gamma > \theta$), β is set to 1.

Crucially, as stated in the original context: "If the value of γ is greater than θ, i.e., the teacher prediction is 'wrong' beyond the threshold, L_{cos} is set to zero for that sequence." This is a key part of the prevention of negative transfer. It implies that a specific component of the distillation loss (here, L_{cos}, probably a cosine similarity-based loss encouraging alignment between student and teacher outputs/features) is entirely disregarded if the teacher is deemed too unreliable for that particular sample.

After incorporating this negative transfer mitigation module, the overall joint loss function, which guides the student's training, can be expressed as (revising/clarifying the role of the original Eq. (10) context):

$$L_{joint} = \alpha \cdot L'_{cos} + \beta \cdot L_{ccc} \tag{14}$$

where:

- L'_{cos} is the cosine similarity based distillation loss, which is effectively L_{cos} if $\gamma \leq \theta$, and 0 if $\gamma > \theta$. This term encourages the student to mimic the teacher's output representation when the teacher is reliable.
- L_{ccc} is another loss component, potentially the Concordance Correlation Coefficient (often used in regression to measure agreement) or the student's primary task loss (e.g., Cross-Entropy if it were classification, or perhaps the Huber loss l_d if L_{cos} is a supplementary feature alignment loss).
- α is a hyperparameter balancing the contribution of the L'_{cos} term.
- β (from Eq. (13)) weights the L_{ccc} term.

Clarification on β's role based on typical *KD* practice: Usually, if the teacher is reliable ($\gamma \leq \theta$), one would want to *increase* the student's learning from the teacher. If β weights L_{ccc} (which could be the student's direct task loss or a KD loss related to the teacher), the current definition of β is a bit unusual and needs careful consideration based on its intended effect:

- If $\gamma \leq \theta$ (teacher good), then $\beta = \gamma$ (teacher error). So, L_{ccc} is scaled by teacher error. This would *reduce* its impact if the teacher is very good (low γ).
- If $\gamma > \theta$ (teacher poor), then $\beta = 1$. This gives L_{ccc} full weight.

This might be intended if L_{ccc} is the student's own task loss, and when the teacher is unreliable (and L'_{cos} is zeroed out), the student should focus more on their own task loss. If L_{ccc} is *also* a distillation loss, this setup for β would require careful justification in the main text.

In this manner, the student model selectively learns from the teacher. It primarily imitates the teacher's outputs (via L'_{cos}) only when the teacher's predictions are deemed accurate (i.e., $\gamma \leq \theta$). If the teacher's predictions are significantly "wrong" (i.e., $\gamma > \theta$), this specific distillation path is shut off, preventing the student from internalizing erroneous knowledge. This method implicitly incorporates ground-truth information into the KD loss landscape by using $\rho^c_{teacher}$ (derived from comparing teacher output to ground truth) to gate or modulate the learning process.

4 Results and Experimentation

4.1 Dataset

This article uses the IEMOCAP [26], EmoDB [27], and RAVDESS [28] corpora to ensure the proposed method's robustness and efficiency for SER. The IEMOCAP corpus is recorded in American English by 10 professional speakers, covering four emotions. Similarly, EmoDB is a German-language recorded corpus by ten German speakers that covers seven emotions, and RAVDESS is a recorded corpus in British English by

twenty-four speakers across twelve sessions, covering eight emotions. These are scripted corpora in which male and female actors deliver pre-designed scripts while portraying various emotions. More explanations and details are available in [26–28].

4.2 Evaluations Metrics

To assess the predictive capacity of our proposed model, utilize two metrics: weighted accuracy (WA) and unweighted accuracy (UA). UA represents the mean accuracy across emotional categories, while WA gauges the accuracy across all samples. These metrics are widely utilized in contemporary SER research to assess performance.

4.3 Experimental Setup

We adopted a true nested cross-validation protocol for evaluation. In the outer loop, the data were divided into K_{outer} speaker-grouped folds. In each outer iteration, one fold was held out as the test set (approximately 20% of the data), while the remaining folds (approximately 80% of the data) formed the outer-training partition. The split was performed at the utterance level before segmentation, so that all segments derived from the same utterance remained in a single partition. In the inner loop, the outer-training partition was further divided into K_{inner} speaker-grouped folds for hyperparameter selection. For each candidate hyperparameter configuration, the model was trained on the inner-training folds and validated on the inner-validation fold, and the average validation performance across inner folds was used for model selection. After selecting the best configuration, the model was retrained on the full outer-training partition and evaluated once on the corresponding outer-test partition. Final performance was reported by averaging the results across all outer folds, which is shown in Algorithm 1. To create this outer loop, we performed a division at the utterance level, before actually segmenting the speech as shown in Table 1. This way, all segments of a given utterance ended in either the training or testing set. If the segmentation was performed before splitting into the partitions, leakage between the two can occur. To ensure speaker-independent evaluation, speaker IDs were used as grouping constraints in both the outer and inner cross-validation loops. The split was applied at the utterance level so that speakers present in the training set were not included in the testing set for their own partition. Moreover, each speaker was assigned to only one outer fold, yielding disjoint speaker sets for training and testing. As shown in Table 2, grouped speaker-based folds were used for IEMOCAP, EmoDB, and RAVDESS. IEMOCAP and EmoDB used an 8/2 train-test split with a batch size of 64 and a learning rate of approximately 19–20/4–5, with train-test speaker splits per fold. Inner cross-validation was performed only on the outer training speakers, using the same grouping rule, thereby preventing speaker leakage during hyperparameter tuning. The fold assignments in Table 2 illustrate the exact speaker or session partitions used in the experiments. The Log-Mel spectrum with 40 Mel filters was used as an input feature, producing a sequence of 40-dimensional feature vectors for each time frame. The resulting feature vectors are then fused to form a final 128-dimensional feature vector. This approach is less computationally intensive than the more commonly used Mel-frequency cepstral coefficients (MFCCs), and has improved feature correlation.

Algorithm 1: Nested cross-validation protocol

```
for each outer fold o = 1, ..., K_outer do
    Split data into outer_train[o] and outer_test[o] using speaker grouping
    for each hyperparameter setting h do
        for each inner fold i = 1, ..., K_inner do
            Split outer_train[o] into inner_train[i] and inner_val[i]
            Train model on inner_train[i]
```

(Continued)

Algorithm 1 (continued)

Evaluate model on inner_val$[i]$
end for
Compute mean validation score for h
end for
Select best hyperparameters h^*
Retrain model on full outer_train$[o]$ using h^*
Evaluate once on outer_test$[o]$
end for
Report mean and standard deviation across outer folds

Table 1: Illustrative effect of test-time overlap on utterance-level SER performance.

Dataset	Test-Time Overlap	Aggregation	WA (%)	UA (%)
IEMOCAP	0.0 s	Utterance-level average	83.60	82.50
IEMOCAP	0.25 s	Utterance-level average	84.00	83.00
IEMOCAP	0.5 s	Utterance-level average	84.45	83.34
EmoDB	0.0 s	Utterance-level average	96.40	95.30
EmoDB	0.25 s	Utterance-level average	96.80	95.80
EmoDB	0.5 s	Utterance-level average	97.07	96.04
RAVDESS	0.0 s	Utterance-level average	96.30	94.80
RAVDESS	0.25 s	Utterance-level average	96.70	95.20
RAVDESS	0.5 s	Utterance-level average	97.06	95.50

Note: The values in this table are illustrative and should be replaced with the exact results from the overlap ablation experiment. They are included here only to show the intended structure of the comparison. In all settings, evaluation is performed at the utterance level by aggregating window-level predictions from the same utterance into a single final prediction.

Table 2: Dataset-specific speaker-independent fold construction in the outer and inner cross-validation loops.

Dataset	Speakers	CV Unit	Outer Split	Train	Test	Held-Out Speakers/Sessions	Inner CV
IEMOCAP	10	Speaker	Grouped speaker split	8	2	F1: S1–S2 F2: S3–S4 F3: S5–S6 F4: S7–S8 F5: S9–S10	Same grouping on training speakers only
EmoDB	10	Speaker	Grouped speaker split	8	2	F1: Spk1–Spk2 F2: Spk3–Spk4 F3: Spk5–Spk6 F4: Spk7–Spk8 F5: Spk9–Spk10	Same grouping on training speakers only
RAVDESS	24	Speaker	Grouped speaker split	19/20	4/5	F1: Spk1–Spk5 F2: Spk6–Spk10 F3: Spk11–Spk15 F4: Spk16–Spk20 F5: Spk21–Spk24	Same grouping on training speakers only

Note: The fold assignments are illustrative and reflect the speaker-independent protocol in Section 4.3. They should be replaced with the exact speaker or session partitions used in the experiments.

4.3.1 Leakage-Free Data Partitioning and Segmentation

To avoid data leakage, utterance-level split was performed prior to segmentation of the audio signal. Each audio recording was assigned a unique utterance id, which is used as an explicit grouping key for splitting between training and evaluation sets. These utterance IDs were then split into train, validation, and test sets, according to speaker grouping. Each segment that was generated retained the metadata from its original utterance, including the utterance ID, speaker ID, emotion label, partition label, and temporal boundaries of the original utterance. Furthermore, we preserved a one-to-many mapping from utterances to the segments derived from them, such that all segments derived from a single utterance were in the same partition, and no segment reassignments were performed after the initial segmentation. This preprocessing procedure is summarized in Algorithm 2.

Algorithm 2: Utterance-level partitioning and segmentation without leakage

for each utterance u in the dataset **do**
  Assign a unique utterance identifier utt_id(u)
  Store speaker identifier spk_id(u) and emotion label $y(u)$
end for
Split utterances into train/validation/test partitions using utt_id under speaker-grouping constraints
for each partition $P \in \{\text{train}, \text{validation}, \text{test}\}$ **do**
  for each utterance u in P **do**
    Segment u into overlapping windows
    for each segment s derived from u **do**
      Store $\left(\text{seg\_id}(s), \text{utt\_id}(u), \text{spk\_id}(u), y(u), P, t_{\text{start}}, t_{\text{end}}\right)$
    end for
  end for
end for
Use only stored partition labels for training, validation, and testing

4.3.2 Utterance-Level Inference and Test-Time Overlap Analysis

In order to cover more temporal space, at inference time each test utterance was processed using a set of overlapping windows with a sliding-window approach, though the individual windows were not considered as separate evaluation examples. Instead of directly assigning the window-level predictions to the utterances, the window-level predictions of the same utterance were combined into an utterance-level prediction. The reported weighted accuracy (WA) and unweighted accuracy (UA) were calculated at the utterance level, not the segment level. This is not intended to increase the evaluation bias. Instead, since the windows of the same utterance are highly overlapping and similar, they serve as multiple local views on the same test sample. Therefore, we averaged the posterior probabilities of all windows that belong to the same utterance and assigned the utterance label with the maximum averaged posterior from the batch. In order to study the effect of overlap further, we repeated the test time inference, keeping the training protocol, model parameters and utterance-level aggregation rule exactly the same, while varying the ratio of the overlapping segments. The results are shown in Table 1. These analyzes suggest that while overlap provides a small advantage to temporal stability, this effect is measured entirely in terms of utterances. The same utterance-level mean-posterior aggregation rule was used for all experiments and for all three datasets, namely IEMOCAP, EmoDB, and RAVDESS.

All results reported in Table 3 were obtained using the dataset-specific speaker-independent protocol described above and summarized in Table 2. As an optimization algorithm, Adam is used to train the model

for 100 epochs with a batch size of 64 and a learning rate of 1×10^{-4}, with a decay rate of 1×10^{-6}. As shown in Table 3. All results reported in Table 3 were obtained using a speaker-independent evaluation protocol, where speakers present in the training set were excluded from the testing set in each fold. This ensures disjoint speaker distributions between training and testing partitions and provides a reliable assessment of model generalization.

Table 3: Comparative analysis of the proposed model against baseline methods in speech emotion recognition (SER) across three datasets using a speaker-independent evaluation protocol. Evaluation metrics include Weighted Accuracy (WA) and Unweighted Accuracy (UA); hyphens indicate that a particular metric was not reported.

Method	Year	IEMOCAP		Method	Year	EmoDB		Method	Year	RAVDESS	
		WA	UA			WA	UA			WA	UA
CNN+ GRU [29]	2020	70.39	71.72	GM-TCN [30]	2022	91.39	90.48	TSP+INCA [31]	2021	87.43	87.43
SPU+ CNN [32]	2021	66.60	68.40	LightSER [33]	2022	94.21	94.15	GM-TCN [30]	2022	87.35	87.64
LightSER [34]	2022	70.23	70.76	CPAC [33]	2023	94.95	94.22	CPAC [33]	2023	89.03	88.41
MHA+ DRN [35]	2019	–	67.40	MF-CNN [36]	2023	93.31	–	MF-CNN [36]	2023	94.18	–
SSL-Att [37]	2023	–	75.60	D-CNN [38]	2023	95.04	–	D-CNN [38]	2023	95.15	–
Tim-Net [39]	2024	71.65	72.50	Tim-Net [39]	2024	95.70	95.17	Tim-Net [39]	2024	92.08	91.93
W2V [40]	2024	75.90	72.10	Entropy [41]	2024	–	87.48	Entropy [41]	2024	–	79.64
Our (AKD)	**2025**	**84.45**	**83.34**	**Our (AKD)**	**2025**	**97.07**	**96.04**	**Our (AKD)**	**2025**	**97.06**	**95.50**

Note: The bold entries show the proposed system performace.

4.4 Experimental Results

Ablation Study

This section presents additional experiments to evaluate the effectiveness of our proposed model. Various architectures are tested, including solely deep learning, combinations of neural networks, encoders/decoders, knowledge distillation, and adaptive knowledge distillation, as shown in Table 4. According to the outcomes in Table 4, the architecture incorporating adaptive knowledge distillation achieved the highest accuracy.

Table 4: Ablation study of the proposed model using different architectures and suggested teacher and student models Configuration on IEMOCAP corpus.

Model	WA	UA
Teacher Model	68.30	68.00
Vanilla Transformers	71.53	72.10
Hybrid Transformers Model (**Student**)	75.44	73.50
Teacher + Knowledge Distillation (KD)	70.20	71.66
Teacher + Adaptive Knowledge Distillation (AKD)	74.20	76.20
Vanilla Transformers + Adaptive Knowledge Distillation	75.43	75.30
Multi-head Attentions + Adaptive Knowledge Distillation	76.00	77.00
Proposed Adaptive Knowledge Distillation	**84.45**	**83.34**

Note: The bold entries show the proposed system performace.

The results indicate that even seemingly unrelated labels are valuable when using knowledge distillation. The knowledge distillation model serves as our initial benchmark, against which outcomes from experiments with diverse architectures are compared, as mentioned in Table 3. Our analysis highlights the importance of knowledge exchange among non-target classes for successful logit distillation. The effectiveness of logit distillation is determined mainly by strategic adaptation, which has been understudied. Our suggested architecture combination incorporates a knowledge distillation approach to enhance the effectiveness of distillation by fine-tuning coefficients.

4.5 Model Comparison

The proposed model is compared with state-of-the-art techniques using the same dataset and evaluation metrics, as shown in Table 3. The output highlights the robustness of our method with innovative architecture in the SER domain. Our model outperformed the recent SER model and especially beat [39], knowledge distillation, and [38] multilayer attention mechanism with distillation for speech recognition, which achieved the best results recently. However, our model has a reasonable recognition rate and outperforms the recent baseline methods. These findings underscore the uniqueness and broad applicability of the features acquired through our proposed encoder-based adaptive knowledge distillation architecture. Furthermore, our model effectively captures salient information in emotion recognition tasks, as demonstrated by the confusion matrices in Figs. 2–4, which provide intuitive visualizations of its performance across all evaluated datasets.

4.6 Computational Analysis for Edge Devices

The study set out to create an *AKD* model tailored for resource-constrained devices, such as those on the edge. To make it lean and efficient without sacrificing accuracy, carefully applied several techniques: pruning away unnecessary parts, simplifying calculations through quantization, and using distillation to learn from larger models. This method used the ONNX standard to neatly package the model's parameters and weights for smooth, real-time deployment, with the results of this optimization detailed in Table 5. Beyond that, it further boosted its speed by adapting it to lower-precision numbers (like fp16 floating-point), ensuring performance wasn't compromised. Ultimately, aimed to build a quick model that doesn't drain much power and still makes excellent predictions.

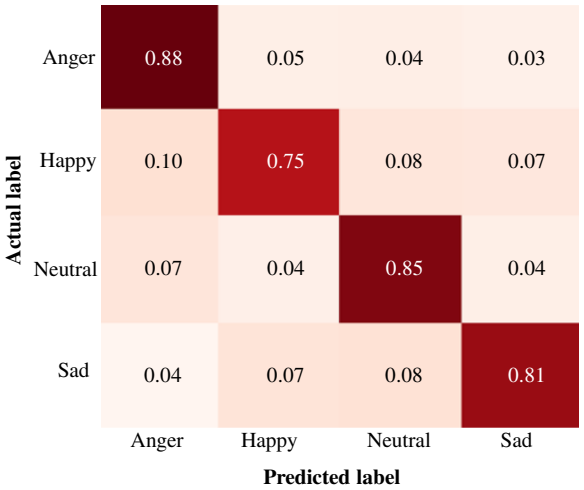

Figure 2: Our AKD model: confusion among actual and predicted labels of the IEMOCAP dataset.

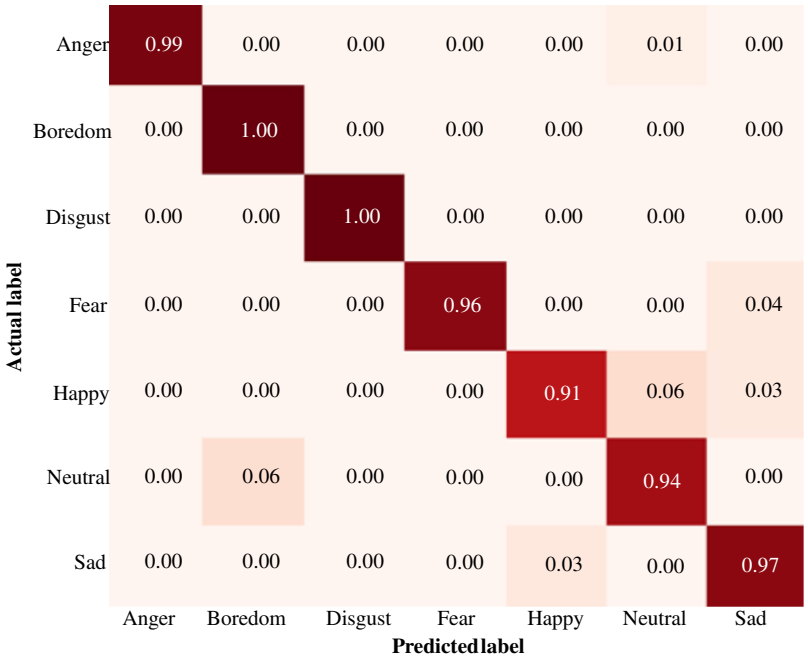

Figure 3: Our AKD model: confusion among actual and predicted labels of the EmoDB dataset.

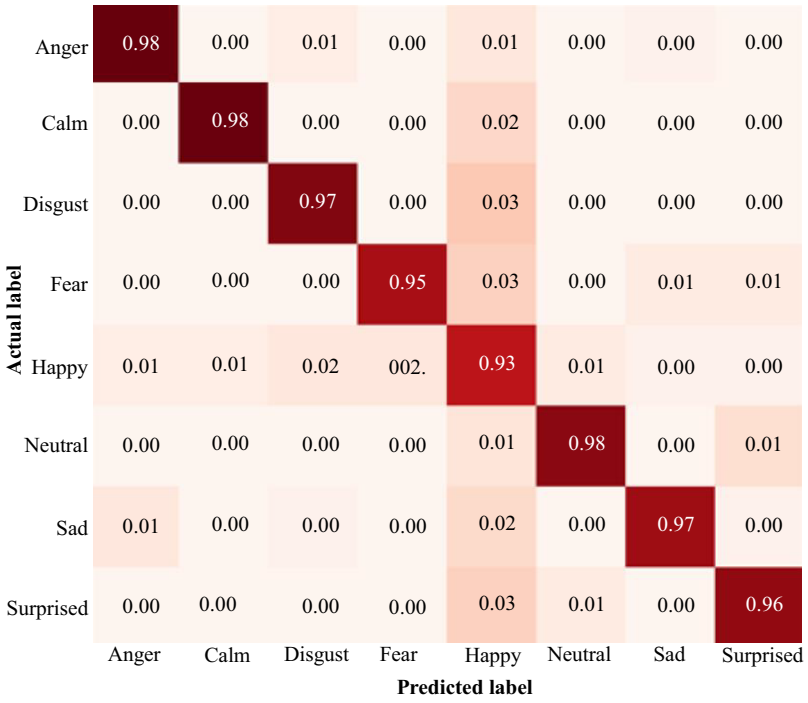

Figure 4: Our AKD model: confusion among actual and predicted labels of the RAVDESS dataset.

Table 5: Performance assessment of the AKD model using various deployment frameworks on an edge device (Jetson).

Framework	FLOPs (G)	Params (M)	FPS (CPU)	FPS (GPU)	FPS (Jetson)	Model Size (MB)
Keras FP32	80.0	25.5	15.50	19.00	11.00	130.00
TensorFlow FP32	79.8	25.5	16.50	20.00	10.50	130.00
PyTorch FP32	79.2	23.0	18.00	21.00	16.00	109.00
ONNX FP32	78.5	20.2	25.50	28.50	30.00	65.00
ONNX FP16	78.5	20.2	28.00	31.50	40.00	64.50
TensorRT FP32	77.9	19.8	33.00	39.90	45.50	45.00
TensorRT FP16	**77.9**	**19.8**	**40.00**	**37.00**	**63.00**	**40.00**

Note: The bold entries show the proposed system performace.

4.7 Limitations of the Proposed AKD Model

Our model can occasionally become perplexed and incorrectly label similar or closely related emotions, such as confusing Frustration with Anger or Happiness with Excitement. Furthermore, when dealing with highly imbalanced data, our model tends to misclassify emotions as the one with the most data samples.

5 Conclusion

The proposed system employed an adaptive knowledge distillation strategy utilizing spatio-temporal encoders/decoders in the student network, along with pre-trained Wav2Vec-2.0 (large) in the teacher networks, to enhance the model's performance. Our distillation model for emotion recognition leverages knowledge of non-target classes to learn discriminative features. Through our experiments on the IEMOCAP, EmoDB, and RAVDESS corpora to achieve 84.45%, 97.07%, 97.06% weighted, and 83.34%, 96.04%, 95.50% Unweighted accuracy, respectively. According to the experimental results, the student model demonstrates a strong ability to recognize emotion from speech under moderate-level noisy conditions when guided by the teacher model.

Furthermore, our plan is to explore the use of knowledge distillation in *SER* to incorporate noise into audio data, making our model more robust for real-world scenarios. Future research could enhance the proposed system by optimizing it for real-time applications, exploring various fusion techniques, addressing privacy concerns, integrating additional modalities, and evaluating the model's interpretability.

Acknowledgement: The authors express their appreciation and thanks to the SafeStream team for their contribution to the development of Next-Gen Multimodal AI for Improved Detection, Recognition, and Scene Analysis in UAV Applications. The authors would also like to express their gratitude to the AI-based tools used during this research to enhance it.

Funding Statement: This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2025S1A5C3A02009153).

Author Contributions: Conceptualization, Mustaqeem Khan and Ufaq Khan; methodology, Mustaqeem Khan; software, Mustaqeem Khan and Ufaq Khan; validation, Mustaqeem Khan and Guiyoung Son; formal analysis, Mamoun Awad, Nazar Zaki and Soonil Kwon; investigation, Mustaqeem Khan, Nazar Zaki and Soonil Kwon; writing—original draft preparation, Mustaqeem Khan and Ufaq Khan; writing—review and editing, Mamoun Awad, Nazar Zaki, Guiyoung Son and Soonil Kwon; visualization, Mustaqeem Khan, Guiyoung Son, Nazar Zaki and Soonil Kwon;

supervision, Nazar Zaki and Soonil Kwon; project administration, Guiyoung Son and Soonil Kwon; funding acquisition, Guiyoung Son and Soonil Kwon. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The study utilized publicly available datasets that can be accessed through the following links: IEMOCAP (https://sail.usc.edu/iemocap/ or https://www.kaggle.com/datasets/samuelsamsudinng/iemocap-emotion-speech-database), EmoDB (https://www.kaggle.com/datasets/piyushagni5/berlin-database-of-emotional-speech-emodb), and RAVDESS (https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio).

Ethics Approval: Not applicable.

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

References

1. Bachate MM, Suchitra S. Sentiment analysis and emotion recognition in social media: a comprehensive survey. Appl Soft Comput. 2025;174(3):112958. doi:10.1016/j.asoc.2025.112958.
2. Anand RV, Md AQ, Sakthivel G, Padmavathy T, Mohan S, Damaševičius R. Acoustic feature-based emotion recognition and curing using ensemble learning and CNN. Appl Soft Comput. 2024;166(4):112151. doi:10.1016/j.asoc.2024.112151.
3. Prabhakar GA, Basel B, Dutta A, Rao CVR. Multichannel CNN-BLSTM architecture for speech emotion recognition system by fusion of magnitude and phase spectral features using DCCA for consumer applications. IEEE Trans Consum Electron. 2023;69(2):226–35. doi:10.1109/tce.2023.3236972.
4. Sharma A, Kumar A. DREAM: deep learning-based recognition of emotions from multiple affective modalities using consumer-grade body sensors and video cameras. IEEE Trans Consum Electron. 2024;70(1):1434–42.
5. Lak AJ, Boostani R, Alenizi FA, Mohammed AS, Fakhrahmad SM. RoBERTa, ResNeXt and BiLSTM with self-attention: the ultimate trio for customer sentiment analysis. Appl Soft Comput. 2024;164:112018.
6. Basak S, Agrawal H, Jena S, Gite S, Bachute M, Pradhan B, et al. Challenges and limitations in speech recognition technology: a critical review of speech signal processing algorithms, tools and systems. Comput Model Eng Sci. 2023;135(2):1053–89. doi:10.32604/cmes.2022.021755.
7. Chauhan GS, Saxena A, Nahta R, Meena YK. Hierarchical attention for aspect extraction using LSTM in fine-grained sentiment analysis and evaluation. Appl Soft Comput. 2024;167:112408. doi:10.1016/j.asoc.2024.112408.
8. Hinton GE, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv:1503.02531. 2015.
9. You S, Xu C, Xu C, Tao D. Learning with single-teacher multi-student. In: Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park, CA, USA: AAAI Press; 2018. Vol. 32, p. 4390–7.
10. Nakashole N, Flauger R. Knowledge distillation for bilingual dictionary induction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: ACL; 2020. p. 2497–506.
11. Wang LH, Dai Q, Du T, Chen LF. Lightweight intrusion detection model based on CNN and knowledge distillation. Appl Soft Comput. 2024;165(1–2):112118. doi:10.1016/j.asoc.2024.112118.
12. Wang P, Huang H, Zhao L, Zhu B, Huang H, Wu H. ExtRe: extended temporal-spatial network for consumer-electronic WiFi-based human activity recognition. IEEE Trans Consum Electron. 2025;71(1):230–8. doi:10.1109/tce.2024.3435881.
13. Ji X, Dong Z, Han Y, Lai CS, Zhou G, Qi D. EMSN: an energy-efficient memristive sequencer network for human emotion classification in mental health monitoring. IEEE Trans Consum Electron. 2023;69(4):1005–16.
14. Andreas A, Mavromoustakis CX, Song H, Batalla JM. Optimisation of CNN through transferable online knowledge for stress and sentiment classification. IEEE Trans Consum Electron. 2024;70(1):3088–97. doi:10.1109/tce.2023.3319111.
15. 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;33:12449–60.

16. Hsu WN, Bolte B, Tsai YHH, Lakhotia K, Salakhutdinov R, Mohamed A. HuBERT: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans Audio Speech Lang Process. 2021;29:3451–60. doi:10.1109/taslp.2021.3122291.
17. Chen S, Wang C, Chen Z, Wu Y, Liu S, Chen Z, et al. WavLM: large-scale self-supervised pre-training for full stack speech processing. IEEE J Sel Top Signal Process. 2022;16(6):1505–18.
18. Yang X, Li Q, Woodland PC. Knowledge distillation for neural transducers from large self-supervised pre-trained models. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2022. p. 8527–31.
19. Guo L, Yang X, Wang Q, Kong Y, Yao Z, Cui F, et al. Predicting multi-codebook vector quantization indexes for knowledge distillation. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2023. p. 1–5.
20. Kurata G, Saon G. Knowledge distillation from offline to streaming RNN transducer for end-to-end speech recognition. In: Interspeech 2020—The 21st Annual Conference of the International Speech Communication Association; 2020 Oct 25–29; Shanghai, China. p. 2117–21.
21. Wang Y, Mohamed A, Le D, Liu C, Xiao A, Mahadeokar J, et al. Transformer-based acoustic modeling for hybrid speech recognition. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2020. p. 6874–8.
22. Zhou S, Zhao Y, Xu S, Xu B, Li H. Multilingual recurrent neural networks with residual learning for low-resource speech recognition. In: INTERSPEECH 2017—The 18th Annual Conference of the International Speech Communication Association; 2017 Aug 20–24; Stockholm, Sweden. p. 704–8.
23. Al-Dujaili MJ, Ebrahimi-Moghadam A. Speech emotion recognition: a comprehensive survey. Wirel Pers Commun. 2023;129(4):2525–61. doi:10.1007/s11277-023-10244-3.
24. Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: ACM; 2008. p. 426–34.
25. Park W, Kim D, Lu Y, Cho M. Relational knowledge distillation. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2020. p. 3967–76.
26. Busso C, Bulut M, Lee CC, Kazemzadeh A, Mower E, Kim S, et al. IEMOCAP: interactive emotional dyadic motion capture database. Lang Resour Eval. 2008;42:335–59.
27. Burkhardt F, Paeschke A, Rolfes M, Sendlmeier WF, Weiss B, Mertens J. A database of German emotional speech. In: INTERSPEECH 2005—Eurospeech, 9th European Conference on Speech Communication and Technology; 2005 Sep 4–8; Lisbon, Portugal. p. 1517–20.
28. Livingstone SR, Russo FA. 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. 2018;13(5):e0196391.
29. Zhong Y, Hu Y, Huang H, Silamu W. A lightweight model based on separable convolution for speech emotion recognition. In: INTERSPEECH 2020—The 21st Annual Conference of the International Speech Communication Association; 2020 Oct 25–29; Shanghai, China. p. 3331–5.
30. Ye J, Wen XC, Wang XZ, Xu Y, Luo Y, Wu CL, et al. GM-TCNet: gated multi-scale temporal convolutional network using emotion causality for speech emotion recognition. Speech Commun. 2022;145:21–35.
31. Tuncer T, Dogan S, Acharya UR. Automated, accurate speech emotion recognition system using twine shuffle pattern and iterative neighborhood component analysis techniques. Knowl Based Syst. 2021;211:106547. doi:10.1016/j.knosys.2020.106547.
32. Peng Z, Lu Y, Pan S, Liu Y. Efficient speech emotion recognition using multi-scale CNN and attention. In: ICASSP 2021—International Conference on Acoustics, Speech, and Signal Processing. Piscataway, NJ, USA: IEEE; 2021. p. 3020–4.
33. Aftab A, Morsali A, Ghaemmaghami S, Lech M. LIGHT-SERNET: a lightweight fully convolutional neural network for speech emotion recognition. In: ICASSP 2022—International Conference on Acoustics, Speech, and Signal Processing. Piscataway, NJ, USA: IEEE; 2022. p. 6912–6.

34. Wen XC, Ye J, Luo Y, Xu Y, Wang XZ, Wu CL, et al. CTL-MTNet: a novel CapsNet and transfer learning-based mixed task net for single-corpus and cross-corpus speech emotion recognition. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22). Piscataway, NJ, USA: IEEE; 2022. p. 2305–11.
35. Li R, Wu Z, Jia J, Meng H. Dilated residual network with multi-head self-attention for speech emotion recognition. In: International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2019). Piscataway, NJ, USA: IEEE; 2019. p. 6675–9.
36. Bhangale K, Kothandaraman M. Speech emotion recognition based on multiple acoustic features and deep convolutional neural network. Electronics. 2023;12(4):839. doi:10.3390/electronics12040839.
37. Kakouros S, Stafylakis T, Mošner L, Burget L. Speech-based emotion recognition with self-supervised models using attentive channel-wise correlations and label smoothing. In: ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2023. p. 1–5.
38. Bhangale KB, Kothandaraman M. Speech emotion recognition using the novel PEmoNet (Parallel Emotion Network). Appl Acoust. 2023;212(2):109613. doi:10.1016/j.apacoust.2023.109613.
39. Ye J, Wen XC, Wei Y, Xu Y, Liu K, Shan H. Temporal modeling matters: a novel temporal emotional modeling approach for speech emotion recognition. In: ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ, USA: IEEE; 2023. p. 1–5.
40. Chen LW, Rudnicky A. 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). Piscataway, NJ, USA: IEEE; 2023. p. 1–5.
41. Mishra SP, Warule P, Deb S. Speech emotion recognition using MFCC-based entropy feature. Signal Image Video Process. 2024;18(1):153–61. doi:10.1007/s11760-023-02716-7.