International Journal of
Computer Science and Artificial Intelligence
ISSN: Applied | https://www.ioro.org/ijcsa

TinyML-Based Speech Emotion Recognition on Microcontrollers Using Quantized CNNs and Mel-Filterbank Energy Features

Mir Muhammad Abidul Haq Ahnaf [1,*]

[1] *Singapore International School, Dhaka, Bangladesh*

* **Corresponding author:** abidulhaqahnaf@gmail.com • ORCID: 0000-0001-8032-8265

Abstract

This paper presents a real-time, resource-efficient Speech Emotion Recognition (SER) system trained on a modified version of the Toronto Emotional Speech Set (TESS) that includes both male and female voice samples for five emotional labels—Angry, Disgust, Fear, Happy, and Neutral—plus a custom Noise class added to improve robustness. The final model was deployed on the M5Stack C Plus, an ESP32-based microcontroller, using TinyML techniques, performing fully offline inference rather than relying on cloud-based processing. Edge Impulse was used to design, train, and optimize a quantized 2D Convolutional Neural Network, leveraging Mel-filterbank energy features extracted from 16 kHz audio captured via a PDM digital microphone interfaced over I²S. The model classifies six emotional states with a validation accuracy of 99.5% and a total inference latency of approximately 571–572 ms, including both DSP and classification phases. The final quantized int8 model occupies less than 2 MB of flash and consumes under 50 KB of RAM. This work demonstrates that accurate, low-latency emotion recognition is feasible on ultra-low-power microcontrollers, making it suitable for privacy-preserving, always-on, embedded human-computer interaction systems.

Keywords: Speech Emotion Recognition (SER); Machine Learning; Real-Time Inference; Quantized CNN; Embedded Systems; TinyML

1. Introduction

Speech emotion recognition aims to identify affective states from vocal signals. Traditional approaches rely on cloud computing, which introduces latency and raises privacy concerns. In contrast, this work explores deploying SER models directly on microcontrollers using TinyML, enabling local, secure, and fast inference.

Let x(t) denote the discrete-time audio signal. The objective is to learn a function $f_\theta(x(t))$ that maps the signal to one of six emotion classes y_1, ..., y_6, where θ are the learnable parameters of a convolutional neural network (CNN).

2. Related Work

Prior work in speech emotion recognition has predominantly used spectrograms and Mel-frequency cepstral coefficients (MFCCs) as input features, coupled with deep learning architectures such as convolutional or recurrent neural networks. However, most of these approaches target high-resource environments, with limited focus on deployment to resource-constrained edge devices. This study builds on recent advances in

embedded artificial intelligence, particularly platforms such as Edge Impulse and TensorFlow Lite for Microcontrollers, to enable real-time, on-device emotion recognition in a low-power, memory-limited setting.

3. Methodology

This section describes the dataset preparation, feature extraction pipeline, and model architecture used to build the SER system.

3.1. Data Preprocessing

An 80–20 split was used to divide the dataset into training and test sets. A Noise class was additionally created using non-verbal environmental sounds to improve model robustness under real-world conditions.

Table 1. Class distribution and duration of the training dataset.

Class	Total samples	Training samples	Test samples	Total duration
Angry	400	322	77	9m 51s
Happy	400	321	79	10m 33s
Disgust	400	320	80	13m 4s
Fear	400	320	79	8m 50s
Neutral	400	320	80	10m 57s
Noise	720	578	142	9m 47s

3.2. Audio Signal Overview

To illustrate the variability of speech patterns across emotional states, representative waveforms were plotted for each of the six target classes. These time-domain plots show the amplitude envelope of 1-second clips sampled at 16 kHz.

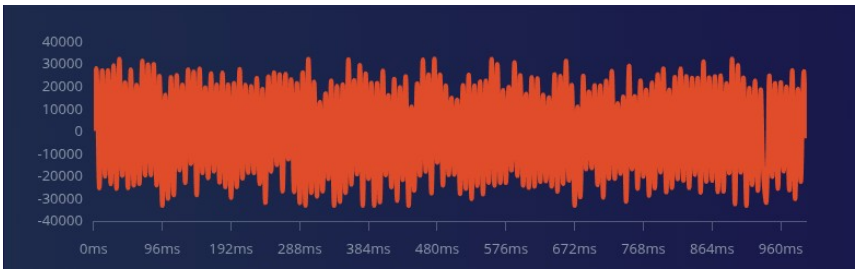

Fig. 1. Representative waveform for the Noise class.

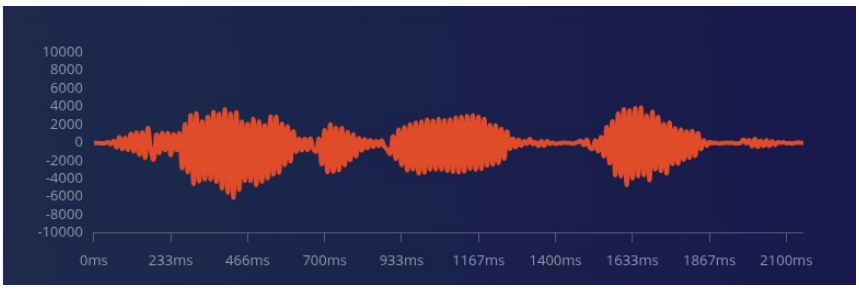

Fig. 2. Representative waveform for the Neutral class.

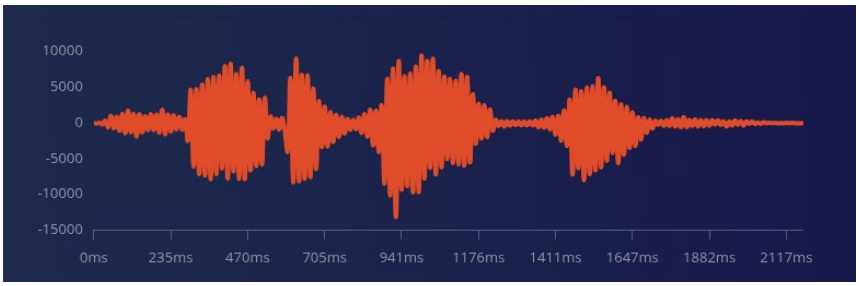

Fig. 3. Representative waveform for the Angry class.

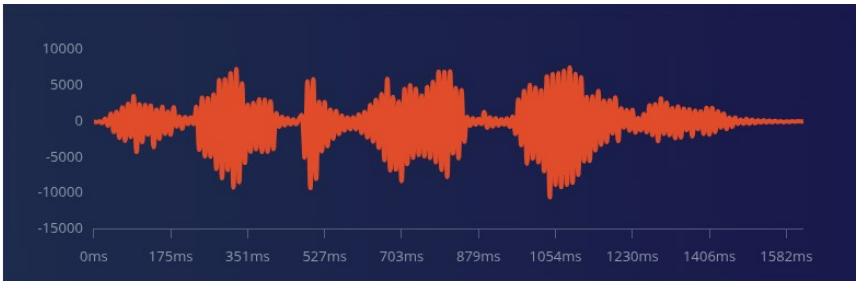

Fig. 4. Representative waveform for the Disgust class.

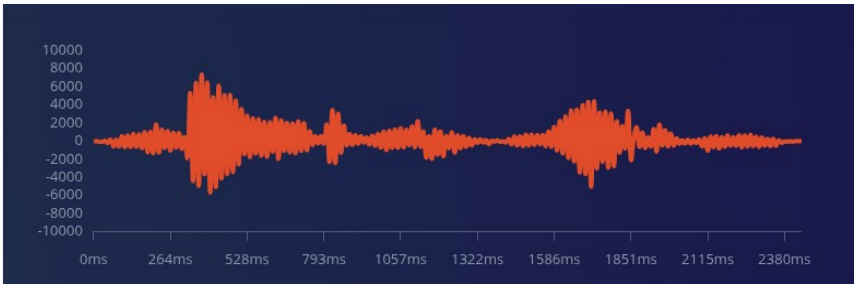

Fig. 5. Representative waveform for the Fear class.

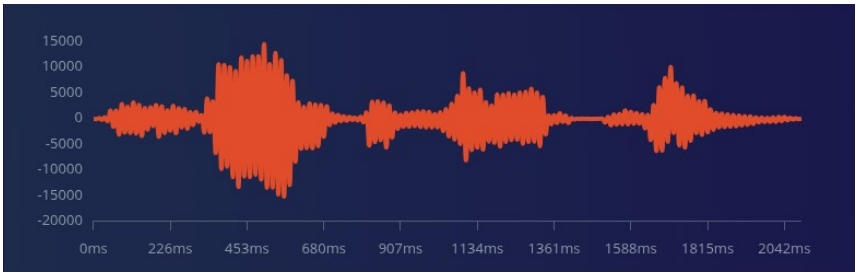

Fig. 6. Representative waveform for the Happy class.

3.3. Signal Processing and Feature Extraction

The input audio x(t) is segmented into frames of length T = 32 ms with a stride S = 32 ms. Each frame is transformed using the Short-Time Fourier Transform (STFT) followed by a Mel-filterbank operation, as shown in Eq. (1) and Eq. (2).

$$X(\mathrm{k},\ \mathrm{n}) = \sum_{m=0}^{N-1} \square\, x(\mathrm{n}S+\mathrm{m})\mathrm{w}(\mathrm{m})\mathrm{e}^{\frac{-j\,2\,\pi km}{N}} \tag{1}$$

(2)

$$MFE(\mathrm{n},\ \mathrm{b}) = \log\left(\sum_{k=k\square_b^{min}}^{k\square_b^{max}} \square \left| X\left(k, n\right)\right|^2 \cdot H_b\left(k\right) + \epsilon \right)$$

where w(m) is the Hamming window, $H_b(k)$ is the b-th triangular Mel filter, $\varepsilon = 10^{-6}$ is added for numerical stability, n is the frame index, and $b \in \{1, \ldots, 32\}$ indexes the 32 filterbanks. Each 1-second clip yields a 32×31 time-frequency matrix of Mel energy features.

Fig. 7 shows the weighting profile applied to FFT bins by the 32 Mel filters, and Fig. 8 shows the resulting Mel energy spectrogram generated from a single 1-second audio sample.

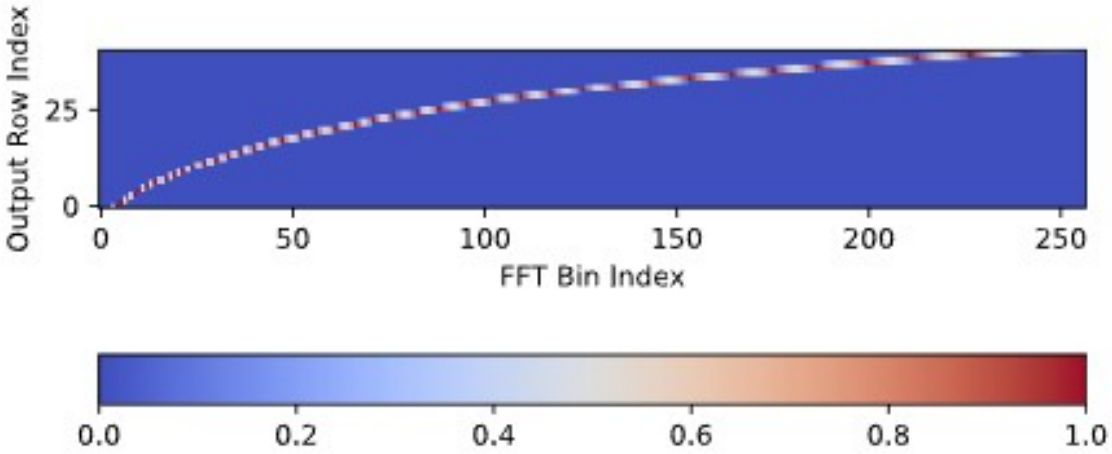

Fig. 7. Weighting profile applied to FFT bins by the 32 Mel filters.

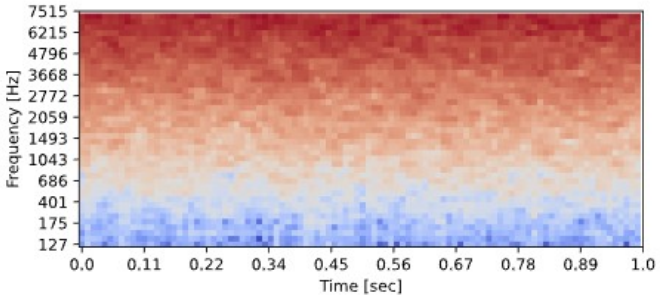

Fig. 8. Mel energy spectrogram generated from a single 1-second audio sample.

3.4. Model Architecture

The CNN is defined as a function f_θ mapping a 32×31 input to a 6-dimensional output. The layers are: a Conv2D layer with weights $W_1 \in R^{3\times3\times1\times8}$ with ReLU activation; a MaxPool2D layer with stride 2; a second Conv2D layer with weights $W_2 \in R^{3\times3\times8\times16}$, repeated up to 64 filters; and a final flatten and dense layer computing a softmax output.

Fig. 9. CNN architecture used for emotion classification.

The loss function is categorical cross-entropy, as shown in Eq. (3). Training uses the Adam optimizer with learning rate η = 0.005, batch size B = 32, and E = 80 epochs.

$$L(y,\hat{y}) = -\sum_{i=1}^{6} \square\, y_i \log(\hat{y}_i) \tag{3}$$

4. Results

4.1. Performance Evaluation

- Validation accuracy: 99.5%
- Validation loss: 0.07
- F1-score (weighted): 0.99
- AUC: 1.00

	ANGRY	DISGUST	FEAR	HAPPY	NEUTRAL	NOISE
ANGRY	100%	0%	0%	0%	0%	0%
DISGUST	0%	99.2%	0%	0.8%	0%	0%
FEAR	1.3%	0%	98.7%	0%	0%	0%
HAPPY	0%	0%	0%	100%	0%	0%
NEUTRAL	0%	0%	0%	0%	100%	0%
NOISE	0.8%	0%	0%	0%	0%	99.2%
F1 SCORE	0.99	1.00	0.99	0.99	1.00	1.00

Fig. 10. Confusion matrix on the validation set.

Let TP_i, FP_i, and FN_i denote the true positive, false positive, and false negative counts for class i. Precision, recall, and F1-score are computed as shown in Eq. (4) and Eq. (5).

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{4}$$

$$F1_i = \frac{2 \times (Precision_i \times Recall_i)}{Precision_i + Recall_i} \tag{5}$$

Average precision, recall, and F1-score across all six classes exceed 98%.

4.1.1. *Training Dynamics and Model Convergence*

To evaluate model convergence and potential overfitting, accuracy and loss were tracked across 80 training epochs. The model converged quickly, reaching over 95% validation accuracy within 20 epochs. Loss curves also stabilized early, with validation loss remaining consistently low across later epochs, suggesting minimal overfitting.

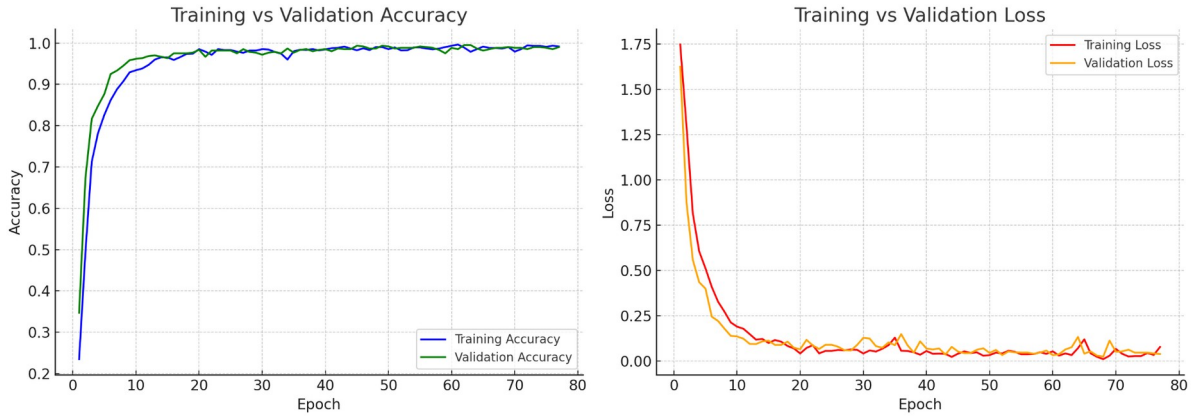

Fig. 11. Progression of training and validation accuracy and loss across 80 epochs.

4.2. Resource Utilization and Performance

- Model size (quantized): ≈ 42.8 KB
- Flash size: ≈ 1.17 MB
- RAM usage: ≈ 29.6 KB
- DSP time: 539 ms

- Classification time: 32–33 ms
- Total inference latency: 571–572 ms

Despite the significant DSP time, the system operates comfortably under a 1-second window, making it viable for real-time inference in low-latency applications.

4.3. Feature Space Visualization

To better understand class separability in the feature space, Edge Impulse's Data Explorer tool was used to project the high-dimensional Mel-filterbank energy feature vectors into a 2D space using Principal Component Analysis (PCA). Fig. 12 shows the distribution of training data by class in this projected space. Clear clustering indicates that the extracted features encode emotional categories effectively, facilitating accurate classification.

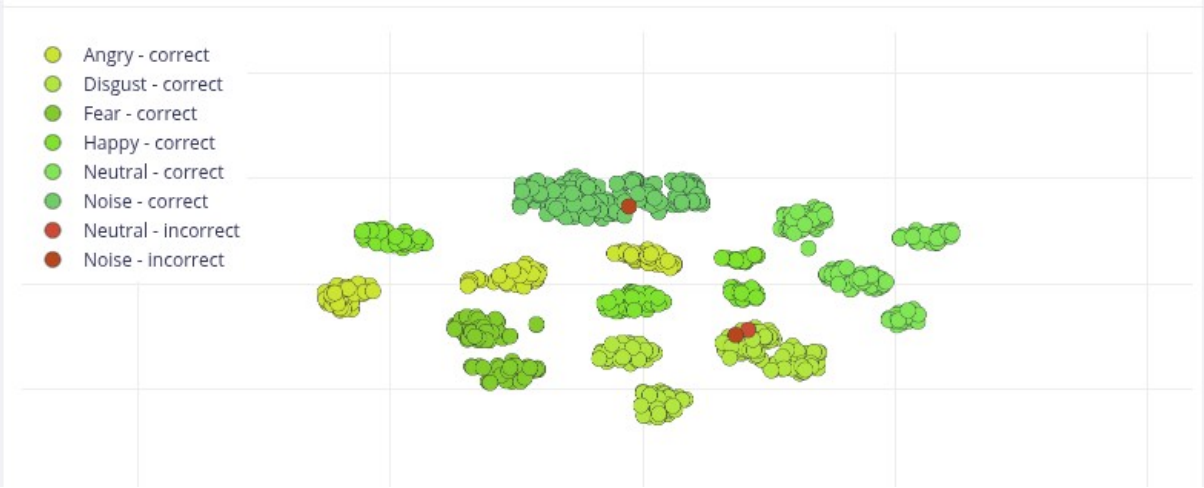

Fig. 12. PCA projection of training data by class in feature space.

5. Deployment

The quantized TensorFlow Lite (int8) model was deployed to an ESP32-PICO-D4 microcontroller (dual-core, 240 MHz, 520 KB SRAM, 4 MB Flash) interfaced with an SPM1423 digital PDM microphone for real-time audio capture. Audio signals were streamed through the ESP32's I²S peripheral configured in PDM receive mode, processed by onboard Mel-filterbank DSP blocks, and classified by the CNN.

Fig. 13. ESP32-based deployment hardware with PDM microphone.

The I²S-PDM integration required configuring the I²S driver with the following parameters:

- Mode: I2S_MODE_PDM
- Sample rate: 16,000 Hz
- Bits per sample: 16-bit packed

- Buffer size: 2048 samples per channel

This configuration enables streaming inference with minimal latency and CPU load. Using the EON compiler, the quantized int8 model was compiled into a binary for the ESP32, and inference is executed entirely offline: audio is acquired via the PDM I2S microphone and passed through the DSP pipeline in real time to produce a classification output.

6. Discussion

This experiment validates that Mel-filterbank energy features and CNN-based classifiers can be efficiently deployed on microcontrollers with limited memory and computation. Despite hardware constraints, the quantized model generalizes well. Limitations include dataset size, noise robustness, and absence of speaker variation. The training and validation metrics in Section 4.1.1 confirm that the model fits well with minimal overfitting under the given dataset. However, the drop in real-world accuracy suggests that overfitting may occur at the feature level due to limited speaker and noise diversity.

6.1. Challenges and Limitations

Although the proposed SER model achieved a validation accuracy of 99.5% under controlled conditions, its performance in real-world scenarios showed noticeable degradation. In field testing with naturally spoken utterances recorded in less controlled environments, the model often misclassified emotions, particularly among acoustically similar categories such as Angry and Disgust. In several cases, only one or two out of five utterances per class were correctly identified, indicating a significant drop in generalization performance. This discrepancy is likely due to a combination of the following factors:

- Overfitting to clean audio: the model was trained on clean, studio-quality samples, whereas real-world environments introduce unpredictable background noise, reverberation, and ambient interference not seen during training.
- Limited speaker diversity: the training dataset included a relatively narrow set of speakers, so the model struggles to generalize across unfamiliar voices, especially those with varying accents, tonal qualities, or pitch ranges.
- Environmental noise robustness: although a custom Noise class was included during training, real-world acoustic interference often differs significantly from the synthetic or recorded noise samples used, leading to misclassification.
- Microphone sensitivity and orientation: the onboard SPM1423 microphone, while adequate for controlled recordings, may suffer from inconsistent sensitivity depending on distance, direction, and environmental conditions, further degrading input quality.

These findings emphasize the importance of incorporating robust data augmentation techniques, collecting diverse real-world samples, and applying adaptive noise filtering to enhance the model's resilience and reliability in deployment environments.

7. Conclusion

A high-accuracy Speech Emotion Recognition (SER) model was implemented on the ESP32 using Mel-filterbank energy features and a quantized 2D Convolutional Neural Network. Achieving near real-time performance and a compact model size, this TinyML-based solution enables efficient, privacy-preserving, and fully offline emotion recognition on ultra-low-power embedded devices.

Acknowledgements

None.

Funding

This research received no external funding.

Conflict of Interest

The author declares no conflict of interest.

Data Availability Statement

The modified Toronto Emotional Speech Set (TESS) used in this study is available on Kaggle (see Reference 6). Trained model files and Edge Impulse project data are available from the author upon reasonable request.

AI Usage Disclosure

Claude (Anthropic) was used to reformat this manuscript into the journal's required template and to check consistency of headings, figure/table numbering, and reference formatting. It was not used to design the experiment, train the model, or generate results. All technical content, analysis, and conclusions are the author's own and were reviewed and verified by the author.

Author Contributions

Conceptualization, methodology, software, formal analysis, investigation, and writing — original draft, M.M.A.H. Ahnaf. The author has read and agreed to the published version of the manuscript.

References

[1] C. R. Banbury, V. J. Reddi, P. Torelli, and J. Holleman, “Micronets: Neural network architectures for deploying tinyML applications on commodity microcontrollers,” arXiv preprint arXiv:2108.04273, 2021.

[2] K. Drossos, F. Ringeval, E. Marchi, and B. W. Schuller, “Automated speech-based emotion recognition: A deep learning approach,” arXiv preprint arXiv:1708.03811, 2017.

[3] A. Aftab, A. Morsali, S. Ghaemmaghami, and B. Champagne, “Light-SERNet: A lightweight fully convolutional neural network for speech emotion recognition,” arXiv preprint arXiv:2110.03435, 2021.

[4] Mustaqeem and S. Kwon, “A CNN-assisted enhanced audio signal processing for speech emotion recognition,” Sensors, vol. 20, no. 1, p. 183, 2020, doi: 10.3390/s20010183.

[5] Edge Impulse, “TinyML platform for embedded machine learning.” [Online]. Available: https://docs.edgeimpulse.com/

[6] “Toronto Emotional Speech Set (TESS),” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess