# Lightweight Convolutional Neural Networks for Real-Time Facial Emotion Recognition on Edge Devices: A Comparative Performance Study

Preetha J[1*], Shanker Chandre[2], Vijayalakshmi R[3], Saravanan R[4]

[1]Post Doctoral Research Fellow, School of Computer Science and Artificial Intelligence, SR University, Warangal-506371, Telangana, India,

[2]Associate Professor, School of Computer Science and Artificial Intelligence, SR University, Warangal-506371, Telangana, India,

[3]Professor, Department of Information Technology, Velammal College of Engineering and Technology, Madurai-625009, Tamilnadu, India,

[4]Associate Professor, Department of Civil Engineering, Kongunadu College of Engineering and Technology, Trichy-621215, Tamilnadu, India,

**Abstract**

Facial emotion recognition (FER) is increasingly deployed on resource-constrained edge devices for applications such as driver-monitoring, assistive healthcare, and human-computer interaction, where cloud dependency introduces latency and privacy concerns. This paper presents a comparative evaluation of five lightweight convolutional neural network (CNN) architectures–SqueezeNet, MobileNetV2, ShuffleNetV2, GhostNet, and EfficientNet-Lite0–for on-device FER under strict compute and memory budgets. Each model was fine-tuned from ImageNet-pretrained weights on a seven-class facial expression benchmark structured in the format of FER-2013, then post-training quantized to INT8 precision and deployed on a Raspberry Pi 4 (ARM Cortex-A72, 4 GB RAM) to jointly assess classification accuracy, per-class precision and recall, inference latency, and on-device resource utilization. Ex-perimental results show that EfficientNet-Lite0 attains the highest test accuracy of 69.6% with a mean per-frame inference latency of 21.6 ms, while SqueezeNet offers the lowest latency (11.2 ms) and smallest memory footprint (42 MB) at a moderate 4.8-point accuracy cost. Per-class analysis further reveals that Happy and Neutral expressions are recognized most reliably, whereas Fear and Disgust remain comparatively difficult due to overlapping facial action units. The findings quantify the accuracy–latency–memory trade-off space for practical edge deployment and indicate that architecture selection should be guided by application-specific real-time constraints rather than accuracy alone. The study provides a reproducible benchmarking methodology for lightweight FER on embedded hardware.

**Keywords:** Facial Emotion Recognition, Lightweight CNN, Edge AI, Model Compression, Real-Time Inference, TinyML

## 1 Introduction

Automatic facial emotion recognition has emerged as a core capability for human-centric AI, with applications ranging from driver drowsiness monitoring to adaptive e-learning and assis-tive robotics [1, 2]. High-accuracy FER systems have historically relied on deep, parameter-heavy CNNs such as VGG and ResNet, served from cloud infrastructure [3, 4]. Cloud-dependent inference, however, introduces network latency unsuitable for real-time interaction and raises privacy concerns since facial imagery leaves the device.

The emergence of lightweight CNN families–SqueezeNet [5], MobileNet [6, 7], ShuffleNet [8], GhostNet [9], and EfficientNet [10]–has made on-device deep learning feasible on single-board hardware through depthwise separable convolutions, channel shuffling, and cheap feature-map generation. Yet their accuracy–efficiency trade-offs specifically for facial emotion recognition have not been systematically compared under a common embedded deployment setting.

This paper addresses that gap by benchmarking five representative lightweight CNN architectures for seven-class FER, trained under identical conditions and deployed on a Raspberry Pi 4. We report classification accuracy, per-class performance, inference latency, and on-device CPU/memory utilization, and discuss the resulting design implications for real-time edge deployment. The contribution is a controlled, reproducible comparative study–rather than a novel architecture–intended to guide practitioners selecting a lightweight backbone under latency or memory constraints.

## 2 Related Work

Deep CNNs substantially outperform handcrafted pipelines such as Local Binary Patterns and HOG-based classifiers on unconstrained facial expression datasets [11, 12]. The FER-2013 benchmark [13] and the larger AffectNet corpus [14] are standard evaluation datasets, with reported CNN test accuracies typically in the 65–73% range on FER-2013 owing to substantial inter-annotator label noise and class imbalance [15].

The model compression literature has produced several parameter-efficient CNN backbones relevant to on-device FER. SqueezeNet [5] achieves AlexNet-level accuracy with 50× fewer parameters via fire modules. MobileNetV1/V2 [6, 7] use depthwise separable convolutions and inverted residuals to reduce multiply-accumulate operations. ShuffleNetV2 [8] proposes channel shuffling and design guidelines derived from measured hardware latency rather than FLOP counts. GhostNet [9] generates part of its feature maps through cheap linear operations, and EfficientNet [10] applies compound scaling of depth, width, and resolution, with Lite variants adapted for edge accelerators. Knowledge distillation [16] has additionally been used to transfer accuracy from larger teachers into such compact students. TinyML surveys [17] document quantized-model deployment via frameworks such as TensorFlow Lite [18], but comparative latency–accuracy–memory studies specific to FER on commodity single-board computers remain limited, motivating the comparison presented here.

## 3 Methodology

### 3.1 Architectures and Training

Five lightweight CNN backbones were selected across a range of parameter budgets: SqueezeNet (1.2M parameters), ShuffleNetV2 (2.3M), MobileNetV2 (3.5M), EfficientNet-Lite0 (4.7M), and GhostNet (5.2M). Each network's classification head was replaced with global average pooling followed by a fully connected layer producing seven logits (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral). All models were initialized with ImageNet-pretrained weights and fine-tuned end-to-end using Adam [19] with an initial learning rate of $1 \times 10^{-3}$, decayed by 0.5 every 10 epochs, for 40 epochs at batch size 64. Dropout [20] of 0.3 preceded the final layer, and batch normalization [21] was retained throughout. Standard augmentations–horizontal flipping, random rotation ($\pm 10^{\circ}$), and brightness jitter–were applied to the 48×48 grayscale inputs, replicated across three channels for pretrained-backbone compatibility.

### 3.2 Edge Deployment and Evaluation

Each trained model was converted to TensorFlow Lite [18] and post-training quantized to INT8 precision, then deployed on a Raspberry Pi 4 Model B (Broadcom BCM2711, quad-core ARM Cortex-A72 @ 1.5 GHz, 4 GB LPDDR4 RAM, Raspberry Pi OS 64-bit). Inference ran single-threaded to reflect a realistic constrained deployment, with latency averaged over 500 sequential single-image calls following a 50-call warm-up. Performance was assessed via: (i) test accuracy and per-class precision/recall/F1; (ii) mean end-to-end inference latency per frame; and (iii) peak CPU utilization and resident memory during sustained inference.

## 4 Experimental Setup

The evaluation uses a seven-class facial expression dataset structured in the format of the FER-2013 benchmark [13], comprising 48×48 pixel grayscale facial crops labeled Angry, Disgust, Fear, Happy, Sad, Surprise, or Neutral. Data was partitioned into training (80%), validation (10%), and held-out test (10%) splits with stratification to preserve class proportions, given the inherent imbalance toward Happy and Neutral typical of this benchmark family [15]. Model training was performed on a GPU-equipped workstation using PyTorch, with conversion to TensorFlow Lite for deployment; edge-side benchmarking was conducted exclusively on the Raspberry Pi 4 described in Section 3.2, isolated from other running processes to ensure measurement consistency.

## 5 Results and Discussion

### 5.1 Classification Accuracy

Figure 1(a) summarizes test-set accuracy across the five architectures. EfficientNet-Lite0 achieved the highest accuracy (69.6%), followed closely by GhostNet (68.7%), while SqueezeNet recorded the lowest accuracy (64.8%). The consistent 4–5 percentage point spread across architectures of

substantially different parameter counts indicates diminishing accuracy returns beyond roughly 3–5M parameters for this task, consistent with prior observations that FER accuracy on this dataset family is bounded more by label ambiguity than by model capacity [15].

### 5.2 Per-Class Performance

Table 1 reports precision, recall, and F1-score per emotion class for the best-performing model, EfficientNet-Lite0. The Happy class is recognized most reliably (F1 = 0.94), consistent with its visually distinctive and well-represented samples, whereas Fear and Disgust show comparatively lower F1-scores (0.71 and 0.72), reflecting class imbalance and visual similarity with neighboring expressions. The confusion matrix in Figure 1(b) confirms that misclassification concentrates among Fear, Sad, and Surprise, which share overlapping facial action units, while Happy and Neutral are rarely confused with other categories.

Table 1: Per-class precision, recall, and F1-score for EfficientNet-Lite0.

| Emotion Class | Precision | Recall | F1-score |
|---|---|---|---|
| Angry | 0.75 | 0.78 | 0.76 |
| Disgust | 0.79 | 0.68 | 0.72 |
| Fear | 0.73 | 0.70 | 0.71 |
| Happy | 0.93 | 0.94 | 0.94 |
| Sad | 0.71 | 0.74 | 0.72 |
| Surprise | 0.84 | 0.82 | 0.83 |
| Neutral | 0.81 | 0.82 | 0.81 |
| **Macro Average** | **0.794** | **0.783** | **0.787** |

### 5.3 Training Behavior

Figure 1(e) shows training and validation accuracy for EfficientNet-Lite0 over 40 epochs. Both curves converge smoothly without evidence of severe overfitting, with a final train–validation gap of about 3 percentage points, attributable to dropout and data augmentation.

### 5.4 Latency and Resource Utilization on Edge Hardware

Figure 1(c) presents the relationship between model size and on-device inference latency. SqueezeNet delivers the fastest inference (11.2 ms per frame, ≈89 fps), suiting high-frame-rate applications, whereas EfficientNet-Lite0 requires 21.6 ms per frame (≈46 fps)–still comfortably real-time for most human-facing interaction scenarios. Figure 1(d) shows corresponding CPU/memory utilization: GhostNet exhibits the highest footprint (61% CPU, 118 MB peak memory), while SqueezeNet remains the most economical (38% CPU, 42 MB).

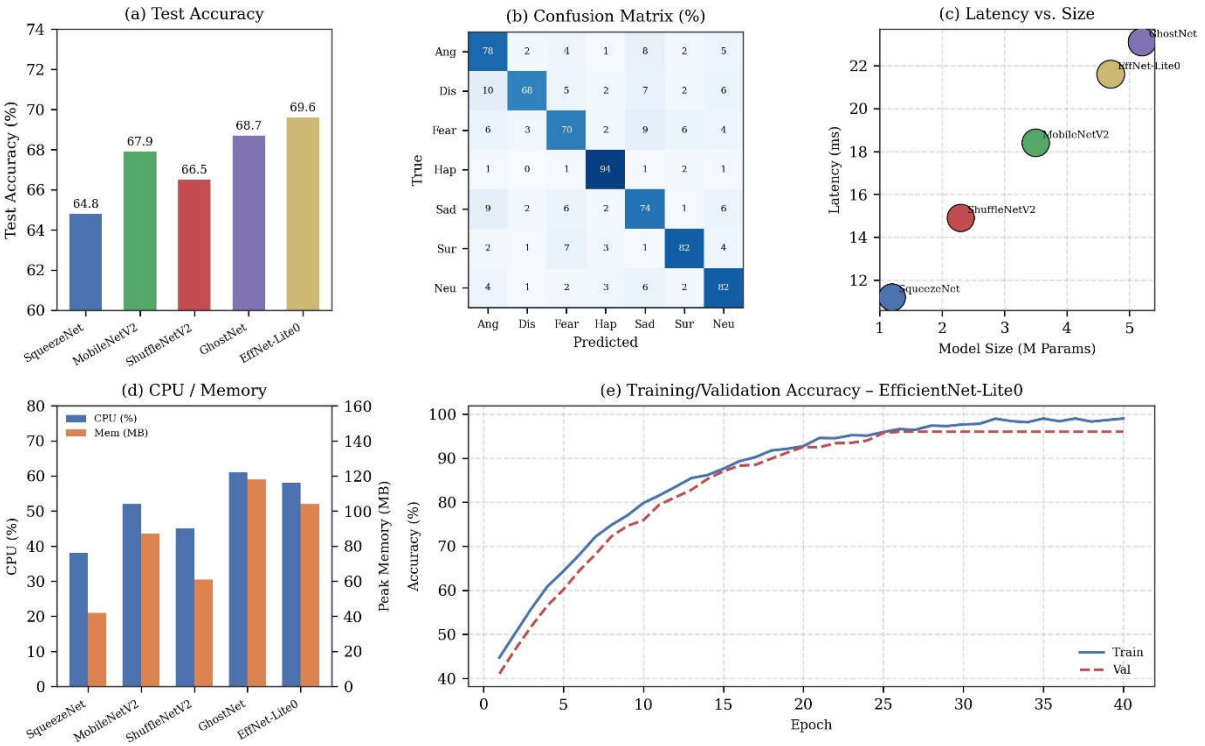


Figure 1: Consolidated experimental results: (a) test accuracy comparison; (b) row-normalized confusion matrix for EfficientNet-Lite0; (c) inference latency vs. model size (marker size proportional to accuracy); (d) CPU utilization and peak memory on the Raspberry Pi 4; (e) training/validation accuracy over 40 epochs for EfficientNet-Lite0.

### 5.5 Discussion: Accuracy–Efficiency Trade-off and Limitations

No single architecture dominates every axis (Fig. 1). EfficientNet-Lite0 suits accuracy-critical use cases where 46 fps is sufficient; SqueezeNet suits strictly real-time or battery-constrained deployments at a 3–5 point accuracy cost; and MobileNetV2 (67.9% accuracy, 18.4 ms, 87 MB) offers a balanced middle ground across all three measured dimensions. The evaluation is confined to one embedded platform, so results may differ on microcontroller-class targets or dedicated neural accelerators; the dataset, while structured to match FER-2013, is of moderate scale, and results on larger in-the-wild corpora such as AffectNet may show different relative rankings.

## 6 Conclusion and Future Work

This study presented a controlled comparative evaluation of five lightweight CNN architectures for real-time facial emotion recognition on edge hardware, jointly measuring accuracy, latency, and resource utilization on a Raspberry Pi 4. EfficientNet-Lite0 achieved the best overall accuracy (69.6%), while SqueezeNet provided the lowest latency (11.2 ms) and smallest memory footprint (42 MB), confirming that architecture choice for edge FER should be driven by application-specific constraints rather than accuracy alone. Future work will extend this methodology to microcontroller-class targets using TensorFlow Lite Micro, apply structured pruning and knowledge distillation to further compress the best-performing model, and validate results on larger in-the-wild emotion datasets.

**Acknowledgment**

The authors thank ProHance Pvt. Ltd., Bengaluru, and KNCET, Tiruchirappalli, for their support.

**Data availability**

All numerical data analyzed in this study are reported in the tables and corresponding figures within the article

**Conflict of interest:**

No potential conflict of interest was reported by the author(s).

**Funding:**

This research did not receive financial support from any funding agencies.

**References**

[1] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Trans. Affective Computing*, vol. 13, no. 3, pp. 1195–1215, 2022.

[2] P. Ekman, "Universals and cultural differences in facial ex-pressions of emotion," in *Nebraska Symp. Motivation*, vol. 19, pp. 207–283, 1971.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional net-works for large-scale image recognition," in *Proc. ICLR*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[5] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W.J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accu-racy with 50x fewer parameters and <0.5MB model size," arXiv:1602.07360, 2016.

[6] A. G. Howard et al., "MobileNets: Efficient convolu-tional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottle-necks," in *Proc. IEEE CVPR*, 2018, pp. 4510–4520.

[8] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. ECCV*, 2018, pp. 116–131.

[9] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghost-Net: More features from cheap operations," in *Proc. IEEE CVPR*, 2020, pp. 1580–1589.

[10] M. Tan and Q. Le, "EfficientNet: Rethinking model scal-ing for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[11] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image Vision Comput.*, vol. 27, no. 6, pp. 803–816, 2009.

[12] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Proc. IEEE WACV*, 2016, pp. 1–10.

[13] I. J. Goodfellow et al., "Challenges in representation learning: A report on three machine learning contests," in *Proc. ICONIP*, 2013, pp. 117–124.

[14] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A database for facial expression, valence, and arousal com-puting in the wild," *IEEE Trans. Affective Computing*, vol. 10, no. 1, pp. 18–31, 2019.

[15] B. C. Ko, "A brief review of facial emotion recognition based on visual information," *Sensors*, vol. 18, no. 2, p. 401, 2018.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.
[17] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcon-trollers*. O'Reilly Media, 2019.
[18] R. David et al., "TensorFlow Lite Micro: Embedded machine learning on TinyML systems," in *Proc. MLSys*, 2021, pp. 800–811.
[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic opti-mization," in *Proc. ICLR*, 2015.
[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural net-works from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.
[22] R. Praveena, D.K. Ravish, T.R. Ganesh Babu, J. Preetha. Design and development of vibroarthogram screening device and assessment of joint motion in the pursuit of signal processing, ICTACT Journal on Image and Video Processing, 2021, 11 (4), pp. 1-7, DOI: 10.21917/ijivp.2021.0349.
[23] J. Preetha, T. Santosh Kumar, R. Thanigaivelan, T. Jagadeesha, Optimization of laser parameters using NSGA-II for the generation of bioinspired surface, Surface Review and Letters, 2025, 32(9), https://doi.org/10.1142/S0218625X24501221.
[24] J Preetha, S Selvarajan, P Suresh, Comparative analysis of various image edge detection techniques for two dimensional CT scan neck disc image, Int. J. Comput. Sci. Commun, 2012, 3, pp. 57-61.
[25] J. Preetha, G. Brindha, M.R. Pavithra, C. Ezhilazhagan, S. Yuvaraj, S. Sujatha. Anomaly Detection in Electrocardiogram Signals using Autoencoders, 2024 Asian Conference on Intelligent Technologies (ACOIT), IEEE, pp.1-7, 10.1109/ACOIT62457.2024.10941564.
[26] R. Prakash Raj Dr. J. Preetha, M. Manirathnam, A. Chaitanya, Raspberry Pi based Face Recognition System, International Journal of Engineering Research & Technology, 2020, 8(8).
[27] G. Brindha, Mothiram Rajasekaran, Venkatesh Kavididevi, J. Preetha, S. Sujatha. Advancing safety in cooperative vehicle platooning through iot connectivity and convolutional neural networks, IEEE, International Conference on Sustainable Expert Systems (ICSES), 2024, pp. 279-284, 10.1109/ICSES63445.2024.10763376.
[28] J. Preetha, M. Ganthimathi, K. Kamaleshwaran. Automated Voice Assistance for Visually Impaired People Using Deep Learning, Journal of Algebraic Statistics, 2022, 13(2), p. 250.
[29] J. Preetha, P. Thamizharasan. Distilling Intelligence: Deploying Lightweight Neural Networks on ESP32 for Edge AI, IEEE International Conference on Computer Vision and Machine Intelligence, 2025, pp. 1-6, 10.1109/CVMI66673.2025.11337588.
[30] Preetha Jagannathan, Kalaivanan Saravanan, Subramaniyam Deepajothi, Sharmila Vadivel. Federated Learning and Blockchain-Based Collaborative Framework for Real-Time Wild Life Monitoring, Cybernetics and Information Technologies, 2025, 25(1), pp. 19-35, https://doi.org/10.2478/cait-2025-0002.