Review of Computer Engineering Research
2026 Vol. 13, No. 1, pp. 69-83
ISSN(e): 2410-9142
ISSN(p): 2412-4281
DOI: 10.18488/76.v13i1.4817

Optimizing deep learning models for facial emotion recognition in embedded systems

Premananda Ramdas[1]
Sunil Swamilingappa Harakannanavar[2+]
Sapna Kumari Chikkanna[3]
Veena Irtayya Puranikmath[4]

[1]*Visvesvaraya Technological University, Jnana Sangama, Belagavi, Karnataka, India.*
Email: rpremananda@gmail.com
[2]*Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology (NMIT), Nitte (Deemed to be University), Nitte University Campus, Yelahanka, Bangalore, Karnataka, India.*
Email: sunilsh143@gmail.com
[3]*Department of Electronics and Communication Engineering, Sapthagiri NPS University, Bangalore, Karnataka, India.*
Email: sapnakumaricc@gmail.com
[4]*Department of Electronics and Communication Engineering, S. G. Balekundri Institute of Technology, Belagavi, Karnataka, India.*
Email: veenaip043@gmail.com


(+ Corresponding author)

Article History
Received: 5 November 2025
Revised: 22 January 2026
Accepted: 4 February 2026
Published: 23 February 2026

Keywords
Deep learning
EfficientNetB0
Embedded systems
Facial emotion recognition
Real-time processing
Transfer learning.

ABSTRACT

Facial emotion recognition (FER) enables intelligent systems to interpret human affect from facial expressions and is increasingly important for human–computer interaction in resource-constrained environments. This work aims to design and evaluate a real-time FER framework that improves recognition accuracy while maintaining low computational complexity, making it suitable for embedded and edge devices. The proposed approach is developed using transfer learning with deep convolutional neural networks, where MobileNetV2 and ResNet50 are implemented as benchmark models, and EfficientNetB0 is selected as the primary model for optimization. Experiments are conducted on the FER-2013 dataset for both training and evaluation, and the input images are preprocessed to enhance facial feature representation. Fine-tuning is performed on the pretrained networks to reduce training time and improve generalization, while preserving real-time feasibility through lightweight inference. The experimental results show that EfficientNetB0 achieves an accuracy of 72.3% with low-latency performance appropriate for real-time operation. ResNet50 provides comparatively higher accuracy but demands greater computational resources, whereas MobileNetV2 offers a more balanced trade-off between speed and recognition performance. These findings indicate that EfficientNetB0 is a practical solution for real-time FER systems, supporting deployment in embedded platforms and applications such as assistive technologies, smart monitoring, and interactive systems where computational efficiency is critical.

Contribution/Originality: The paper contributes the first logical analysis of real-time facial emotion recognition using lightweight transfer learning models under low-latency constraints for embedded deployment. The primary contribution is finding that EfficientNetB0, with efficiency-oriented preprocessing on the FER-2013 dataset, achieves 72.3% accuracy while maintaining practical computational efficiency.

1. INTRODUCTION

To make computers and people interact better, we need to understand how people feel through their facial expressions. This will help systems respond in a more natural and caring way [1]. This capability can be useful in many areas, such as customer service, healthcare monitoring, education, and smart user interfaces [2]. Despite its

importance, accurate emotion recognition remains challenging because it requires large, carefully labeled datasets and substantial computational resources [3]. Deep learning models cannot be used in embedded and real-world systems very often because they require a lot of time and powerful hardware to train initially. Recent research has highlighted transfer learning as an effective approach to overcome challenges such as limited data and high computational requirements [4]. Transfer learning utilizes knowledge from pre-trained models to address new tasks, such as facial expression recognition, while minimizing the need for large datasets and heavy computational resources [5]. Transfer learning can significantly enhance the accuracy of facial emotion recognition. Studies utilizing advanced deep learning models have consistently shown notable performance improvements, often surpassing the results of traditional baseline methods [6]. Figure 1 shows the block diagram of the system that can tell what someone's facial expression is. The diagram shows the main steps: collecting data, cleaning it up, extracting features, and classifying them. It provides a short list of the steps needed to accurately identify and name different emotions in facial photos.

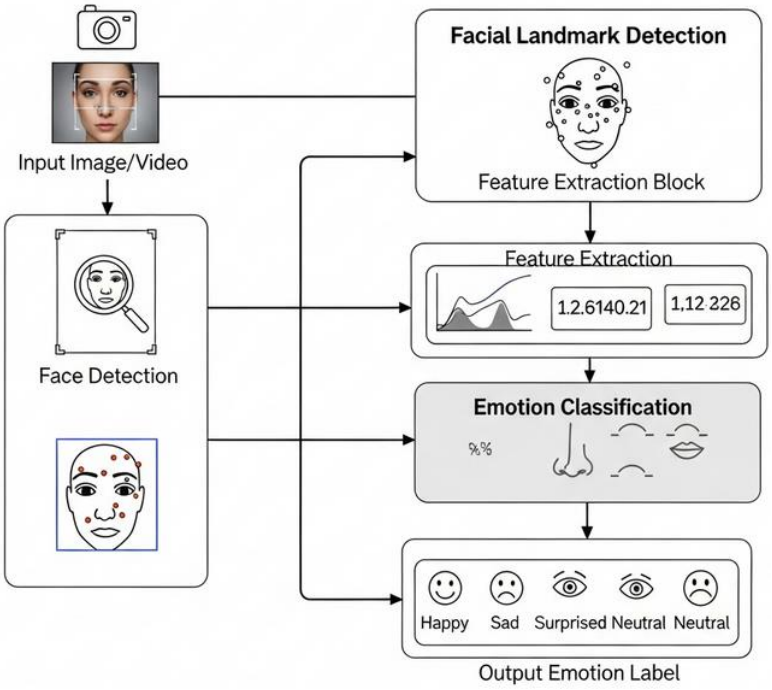

Figure 1. Block diagram of Facial expression recognition.

In this study, pre-trained models such as ResNet50, MobileNetV2, and EfficientNetB0 are utilized to evaluate the effectiveness of transfer learning for facial emotion recognition. These models are fine-tuned and evaluated to measure their accuracy, processing speed, and suitability for real-time applications. The main objective is to identify models that can operate efficiently in real-time emotion recognition systems, particularly on embedded or resource-constrained platforms. As the demand for emotion-aware technology grows, there is a renewed focus on facial expression recognition, especially for applications in human-computer interaction, mental health assessment, and remote learning. This research concentrates on lightweight model architectures designed to balance computational efficiency with recognition accuracy, ensuring they remain practical for real-world deployment. The rest of the paper is organized as follows: Section I discusses the need for real-time facial emotion recognition systems. We begin Section II with a look at prior research covering transfer learning approaches alongside existing work in facial expression analysis. In Section III, we outline the methodology of the proposed system. Section IV presents experimental findings. Lastly, Section V concludes the research, including the potential for multimodal emotion recognition and methods to further enhance model accuracy.

2. LITERATURE SURVEY

Facial emotion recognition (FER) is one of the most advanced artificial intelligence (AI) applications, useful for security, surveillance, mental health monitoring, and human-computer interaction. Recent progress in deep learning, especially convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and transfer learning, has significantly enhanced FER performance [7].

Controlled experiments have demonstrated these approaches surpass several older machine learning methods; for instance, principal component analysis (PCA) and support vector machines (SVMs) cannot capture such delicate, complicated nuances as are found in human feelings. CNNs, in turn, can learn hierarchically discriminative features from raw images. As such, they are much more accurate in understanding human affect compared to older techniques that rely on hand-crafted features.

According to Jun et al. [8], combining CNNs with LSTMs allows achieving an accuracy level of 99% under laboratory conditions. However, challenges such as limited data, background variation, occlusion, and illumination variation hinder FER deployment in the field [9]. The highest accuracy we can achieve on the FER2013 dataset is only 63%, which calls for the development of larger, more representative datasets and better techniques to tune model performance on increasing amounts of data.

Real-time face expression recognition (FER) based on CNNs [10] may seem promising, but it could also have several shortcomings because they require high computation time and expressions are continuously changing. Multimodal approaches, which combine facial expressions and physiological signals (e.g., HR, EEG, voice), have improved classification by capturing wider ranges of emotional indicators [11]. Several studies verified that the performance of models trained on balanced, clean datasets may not generalize well to complex, culturally biased, or imbalanced datasets [12].

By fine-tuning the MobileNetV2 architecture [13], an improved system for recognizing facial expressions has been made more accurate and needs less processing power. A way to find out how someone is feeling in real time that uses a mix of deep learning methods and layers to speed up and improve recognition [14]. This finding confirms the longstanding concerns that dataset bias, generalizability, and cultural differences in expressions present challenges to FER system deployment.

The deep learning era brought significant advances to FER; however, the models are still challenged in terms of robustness and deployment in real-world settings. Future studies should focus on multimodal datasets, hybrid modeling approaches, and efficient real-time frameworks to improve the robustness of FER systems. Some of the most relevant challenges include the need for large, labeled datasets for effective training; the presence of real-world artifacts such as poor lighting, occlusion, and poses that may compromise the robustness of the models; performance degradation due to aging, racial bias, and emotion expression intensity variability; and struggles to categorize subtle or blended emotions.

These challenges emphasize the importance of robust, lightweight, adaptable solutions for FER.

3. PROPOSED METHODOLOGY

This paper presents four novel deep learning models for emotion classification using facial expressions. One model was selected, as shown in Figure 2, based on its specificity and promising ability to perform efficient class-to-class facial emotion recognition.

The FER-2013 dataset was used for facial emotion recognition in this study [15]. This dataset consists of a wide range of realistic grayscale images captured from real-world scenarios.

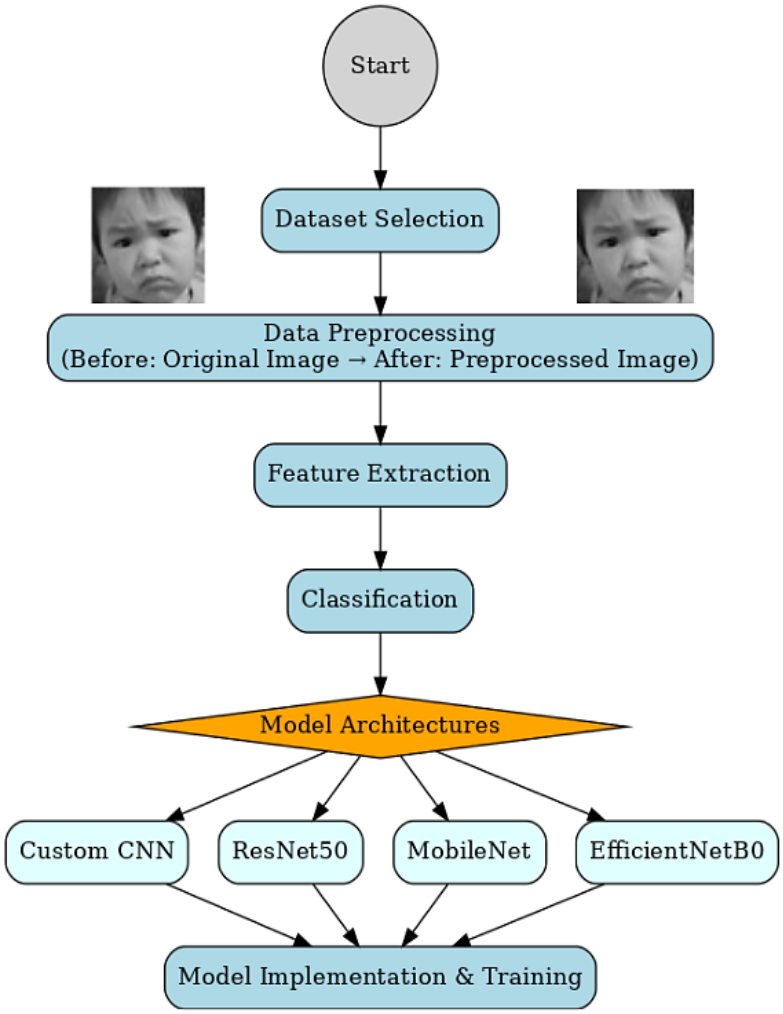

Figure 2. Proposed Facial expression recognition model.

The images are at a resolution of 48×48 pixels. Several targeted data augmentation steps were performed on the datasets to provide better representation and improve model generalization. These levels of augmentation included rotation, flipping, and brightness variation. As a result, the robustness of the model was also improved when dealing with occlusion, changing lighting conditions, and facial obstructions [16]. The first step of the preprocessing model is provided in Figure 3. We can see in Figure 3 (a) the raw input from our created dataset. In Figure 3 (b), face detection was performed by the Haar Cascade classifier to find and match the face area within the image by drawing a bounding box around it. The face region is then cropped from the rest of the photo. The advantage here is that unnecessary background information is eliminated, and only the facial features of interest remain for further processing or analysis.

Figure 3. Original image and detected face: (a) Original captured image, (b) Face detection with region marking using the Haar Cascade classifier.

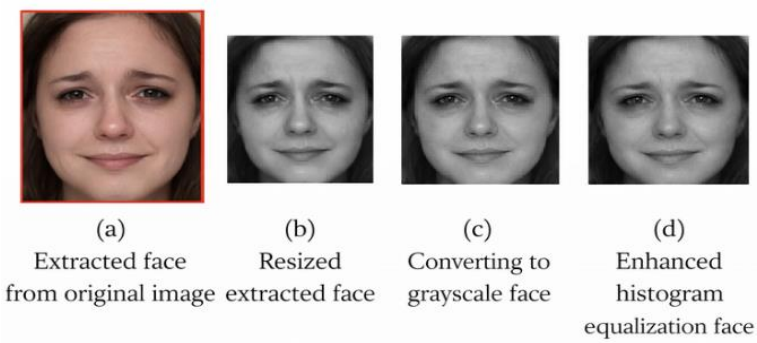

Figure 4. Preprocessing stages of facial image: (a) Extracted face from original image, (b) resized extracted face, (c) converting to grayscale face, and (d) enhanced histogram equalization face.

The facial image used in this study was taken from the FER 2013 dataset [17]. The suggested pre-processing framework illustrated in Figure 3 improves the images through four major stages. The face detection extracts the facial region using the Haar Cascade classifier. The facial alignment corrects face orientation by linking to important landmarks such as the eyes and mouth. Histogram equalization improves contrast, helping facial details appear more clearly under various lighting conditions. Normalization enhances pixel intensity values to a standard range, keeping brightness constant and standardizing input. These operations produce a well-structured and aligned face, making facial images suitable for accurate emotion recognition. In Figure 4 (a), the facial region is localized and cropped from the original image, removing unnecessary background parts and keeping only the area of interest for further image processing operations. The localized facial region was resized to a standard size for uniformity of all image samples and to standardize subsequent image processing steps shown in Figure 4 (b). Next, the resized image is converted to grayscale, which reduces the complexity of the image by showing only the intensity features, further simplifying the process and reducing data redundancy as shown in Figure 4 (c). Contrast enhancement was achieved by histogram equalization applied to the grayscale image, which adjusts pixel intensity values to increase the visual contrast of the facial region and emphasize key facial features, as shown in Figure 4 (d). We studied the effects of these approaches with a custom CNN and MobileNet by running models trained with and without augmentation and pre-processing techniques. The models trained on datasets without augmentation demonstrated weak generalization, especially on under-represented emotions like disgust and fear.

On the other hand, the models trained with augmented datasets demonstrated increased accuracy, decreased overfitting, and more balanced confusion matrices. Histogram equalization and face alignment proved to be the most effective methods for producing subtle expressions and standardizing face position. The MobileNet models, which employ depthwise separable convolutions for feature extraction, demonstrated efficient use of features, revealing minimal trade-offs between observation and accuracy, making them suitable for real-time applications. The combination of augmentation and pre-processing techniques resulted in improved model performance, such as increased accuracy levels, enhanced performance on test data, and better handling of out-of-distribution samples. Overall, the findings indicate that the quality of data preparation processes is critical in developing accurate and robust facial emotion recognition systems.

In our work, CNNs were used as the basis of our facial emotion recognition framework. CNN architecture can efficiently process facial images and find important features that discriminate against the various emotional expressions. Figure 5 illustrates the CNN design that comprises stacked convolutional layers followed by pooling and dense layers for the purpose of feature extraction from grayscale facial images [18]. Figure 6 shows the CNN confusion matrix of the classification results, which demonstrates a moderate level of classification accuracy with a high tendency of misclassification between emotion classes of similar visual appearance, such as fear and surprise. Figure 7 shows the MobileNet architecture that leverages depth-wise separable convolutions to reduce the computational complexity while maintaining the most important features of the face. Such efficiency allows

MobileNet to be optimal for real-time emotion recognition on embedded systems. Figure 8 shows an improved confusion matrix for MobileNet that demonstrates better accuracy and generalization, especially for underrepresented emotion classes [19].

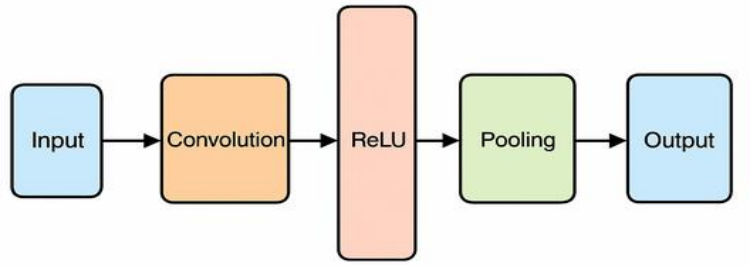

Figure 5. CNN architecture used in the model.

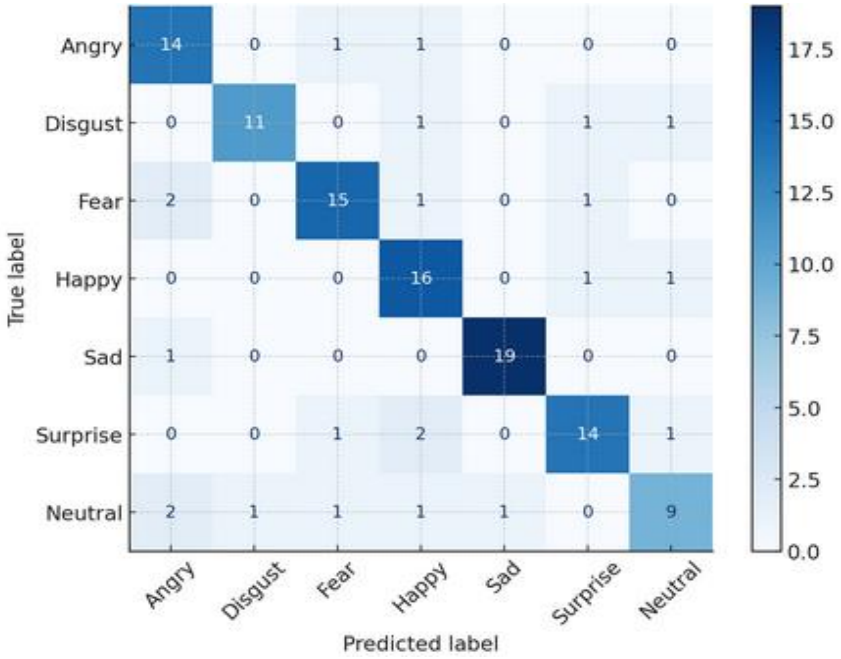

Figure 6. Confusion matrix representing the CNN model's classification performance.

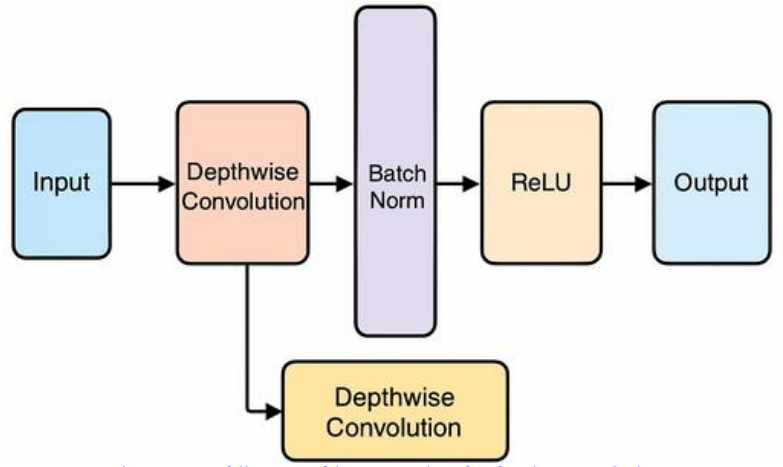

Figure 7. MobileNet architecture using depth-wise convolutions.

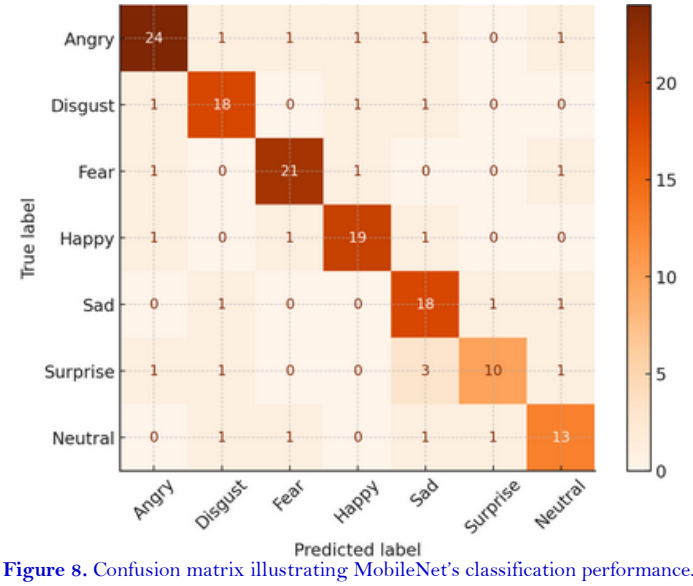

Figure 8. Confusion matrix illustrating MobileNet's classification performance.

The training of the facial emotion recognition system was carried out using four different deep learning architectures: CNN, MobileNet, ResNet50, and EfficientNet, each with its own set of training parameters due to their structural differences, as shown in Table 1. CNN, which was the simplest of the four, was trained over 15 epochs with a dropout of 0.25 to prevent overfitting. MobileNet was used because of its deployability in embedded environments and had a higher dropout rate of 0.6, trained over 20 epochs, with depthwise separable convolutions to make it more resource-efficient.

Table 1. Training Parameters for various Models.

Parameter	CNN	MobileNet	ResNet50	EfficientNet
Number of Epochs	15	20	10	8
Batch Size	64	64	64	64
Dropout	0.25	0.6	0.6	0
Learning Rate	0.001	0.001	0.001	0.0001
Optimizer	Adam	Adam	Adam	Adam

The third architecture, ResNet50, was chosen for its deep residual learning feature, which allows the training of very deep neural networks without the problem of the vanishing gradient problem, and was trained over 10 epochs. The last architecture was EfficientNet, which scales depth, width, and resolution uniformly (or balanced), and the architecture was able to be trained within the lowest number of epochs, i.e., 8, because of the nature of the architecture, and with a lower learning rate to ensure the model was stable without training issues. All models were trained with the Adam optimizer and a batch size of 64.

The proposed algorithm begins with the procurement of a labeled image dataset, with each image correctly aligned with its appropriate class label. Each image-label pair provides an observation used to train and validate the model. The data then undergoes a series of preprocessing operations to ensure homogeneity of the samples. These processes include normalization of pixel intensity values and resizing all images to a common spatial dimension. Feature extraction is then performed through a series of convolutional and pooling layers. The convolutional layers detect and encode both spatial pattern and texture information, whereas the pooling layers reduce dimensionality while maintaining key features [19]. The resulting feature maps are layer-flattened into a single vector and passed through the classification layer, which assigns probability values to each class with a softmax activation. Computational efficiency and accuracy are further improved using residual learning, depthwise separable convolution, or compound scaling depending on the architecture, either Custom CNN, ResNet, MobileNet, or EfficientNet. The network uses a

categorical cross-entropy loss function for training, and its parameters are iteratively optimized via gradient descent until convergence. Once training is finished, the final parameters are saved and used for the prediction of new image data. To formalize this process, the dataset, model configuration, and algorithmic stages are expressed in the following manner.

- Dataset Representation.

Input: The dataset of images.

$$D = \{(x_i, y_i) \mid i = 1,2, \cdots\} \quad (1)$$

Where D denotes the complete set of paired observations, where $x_i$is the i^{th} image and $y_i$is its corresponding category label. Equation 1 establishes the total number of images–label pairs considered in the analysis.

- Model Architecture.

$$M \in \{\text{Custom CNN}, \text{ResNet50}, \text{MobileNet}, \text{EfficientNetB0}\} \quad (2)$$

Where M identifies the configuration type adopted for processing. Equation 2 specifies the selected structural arrangement used for feature generation and classification.

- *Output:* Trained model parameters Θ, Predicted labels $\hat{y}$.

Where Θ is the final set of numerical parameters that provide the configuration of the computational process once the iterations are completed [20]. These parameters are the learned weights (W) and biases (b) of the network. These are fixed numerical values obtained once the training has been performed, and they provide the model final configuration to perform prediction and analysis. $\hat{y}$ is the label or category resulting from each input after completion of all computational steps. It indicates the classification result of each component in the dataset. Both Θ and $\hat{y}$ together render the output for this process, where the former contains the learnt parameters' values and the second specifies the class labels provided with respect to the dataset being classified. To obtain these results, the complete procedure of the proposed method is presented in the following algorithm. It describes the entire process from obtaining the dataset and its pre-processing to feature extraction, classification, and learning the models to generate the results.

Dataset selection: Get dataset D and verify (x_i, y_i) pairing.

1. Data preprocessing: For each x_i.

$$\text{Normalize: } x_i' = (x_i - \mu)/\sigma \quad (3)$$

Where x_i' is the normalized image, μ the mean, and σ the standard deviation of pixel intensities. The normalization in (3) puts all values of the image on the same numerical scale and facilitates subsequent calculations.

$$\text{Resize: } x'_i \in \mathbb{R}^{H \times W \times C} \quad (4)$$

Where x_r is the resized output with H the height, W the width, and C the number of channels. Equation 4 confirms that all input images have the same spatial dimensions [21].

2. Feature extraction: Initialize $f_0 = x_i$. For each convolutional layer

$$l: \begin{cases} f_l = \sigma(W_l * f_{l-1} + b_l) \\ f_l^{\{\text{pooled}\}} = \max_{\{p \in p_l\}} f_l(p) \end{cases} \quad (5)$$

Where:

- f_l is the output feature map of the l^{th}convolutional layer.
- $f_{l-1}$is the input feature map obtained from the preceding layer.
- $W_l$and $b_l$denote the weight matrix and bias vector of the l^{th}layer.
- $*$indicates the convolution operation, σ is the activation function applied to introduce nonlinearity.
- $P_l$represents the local pooling region over which the max-pooling operation is performed.

Equation 5 defines the process through which structural and textural features are derived at successive convolutional stages, transforming the input image into progressively abstract feature representations [22].

3. Classification: Flatten final feature map f. For each class.

$$k: \begin{cases} Z_k = W_k k^T f + b_k \\ \hat{y}_k = \exp(z_k) / \sum_{\{j=1\}}^{K} e^{zj} \end{cases} \quad (6)$$

Where:

- z_k is the logit (Raw score) corresponding to class k,
- W_k and b_k are the weight vector and bias term associated with class k,
- f is the flattened feature vector obtained from the preceding convolutional layers,
- K is the total number of classes, and
- $\hat{y}_k$ represents the predicted probability of the input belonging to class k, obtained using the softmax function.

Equation 6 defines the mapping from the extracted feature vector f to the probability distribution over all classes.

4. Model architecture

If M = Custom CNN: use standard convolutional design.

If M = ResNet50: use residual mapping

$$y_l = F(x_l, \{W_i\}) + x_l \quad (7)$$

Where:

- y_l is the output of the l^{th} residual block.
- x_l is the input to the residual block.
- $F(x_l, \{W_i\}c$ represents the residual function, which is a sequence of convolutional, batch normalization, and activation operations parameterized by weights $\{W_i\}$.

Equation 7 makes sure that the input x_l is directly incorporated in the transformed output $F(x_l, \{W_i\})$, to maintain the flow of information and lessen the vanishing gradient problem in deep networks. When M is MobileNet: Use depthwise separable convolutions to lessen the computational complexity with the same level of accuracy.

$$d = \alpha^{\varphi}, w = \beta^{\varphi}, r = \gamma^{\varphi} \quad (8)$$

When M is EfficientNetB0: Scale network dimensions using compound scaling where in (8), d, w, and r refer to the depth, width, and resolution parameters respectively; α, β, and γ are constants; and ϕ denotes the scaling coefficient. These relations in (8) preserve proportional growth among the system's dimensions [23].

5. Model implementation & training: Categorical cross-entropy loss.

$$L = -\left(\frac{1}{N}\right) \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(\hat{y}_{ik}) \quad (9)$$

Where:

- L represents the total error over the dataset.
- N is the total number of samples.
- K is the number of classes.
- y_{ik} is the actual class indicator for sample i and class k (1 if the sample belongs to class k, 0 otherwise), and
- $\hat{y}_{ik}$ is the predicted probability that sample i belongs to class k.

This loss function punishes wrong predictions more harshly when the model is sure but wrong, which helps the network make probability distributions that are closer to the true labels. Equation 9 gives a number that shows how far off the obtained outputs are from the reference outputs [24].

$$W^{t+1} = W^t - \eta \frac{\partial L}{\partial W} \quad (10)$$

Where w_t is the parameter matrix at the t iteration and η is the step size or adjustment rate. Equation 10 states the way in which the parameters are gradually improved to reduce the error that has been calculated.

5. Repeat until convergence or max epochs.
6. Output: Save Θ and deploy a model for prediction on unseen x_{test}.

4. RESULTS AND DISCUSSION

In order to find the most accurate and efficient way of performing facial emotion recognition, it is very important to compare different deep learning models. Each model has its own strengths, so an honest comparison would show in which architecture almost all trade-offs are balanced between accuracy, speed, and adaptability to real-world settings. The comparison helps to decide on a model according to the requirements of the use case. A total of four models were trained and tested. ResNet50 is a deep architecture with fifty layers designed to identify complex patterns in data. MobileNet is a lightweight network suitable for mobile and embedded platforms that emphasize efficiency rather than relying on large computational resources. EfficientNet is a network that is a compromise between the depth and width of a network and has the potential to produce high accuracy while spending very little computational power. Yet another fully customized CNN was trained to serve as an additional basis of comparison and provide a much more highly optimized baseline. All models were trained under the same conditions. When examining these models, three factors should be kept in mind. The first considers the accuracy, which means how often correct guesses are made. The second considers the loss, or the difference between the output and label for each sample, and how much learning is taking place. The third is speed, or how fast the models train and make predictions, both in terms of elapsed time and hardware use. In combination, these three factors are an indication of which of the models is best suited to balancing recognition performance and resource use.

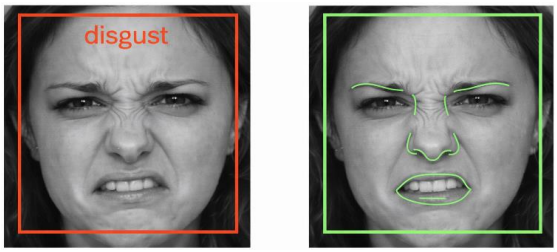

Figure 9. Disgusted face prediction.

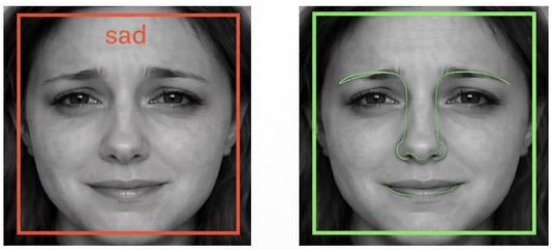

Figure 10. Sad face prediction.

While ResNet50 advances feature extraction, it requires more computation. Both MobileNet and EfficientNet offer efficient performance with some trade-offs in accuracy and tuning requirements before use. Figures 9 and 10 show the images of the predictions of the system while being tested in real-time on emotions such as disgust and sadness, which clearly show the system can practically work in different emotional contexts. Figure 11 shows the classification accuracy achieved by four models: CNN, ResNet50, MobileNet, and EfficientNet. The most accurate model was the custom CNN, which achieved an accuracy of 0.61, followed by EfficientNet with an accuracy of 0.60. MobileNet's accuracy was 0.57, while ResNet50 had the lowest accuracy at 0.51. The low accuracy of ResNet50 is probably due to overfitting from limited data, which might have also affected the CNN. Nevertheless, the comparison shows that lightweight models can outperform deeper architectures in resource-constrained contexts. After training, the custom CNN still outperformed all other models regarding the F1 score for nearly all emotion classes, with the CNN appearing better than the deep models due to its specific framework and feature extraction.

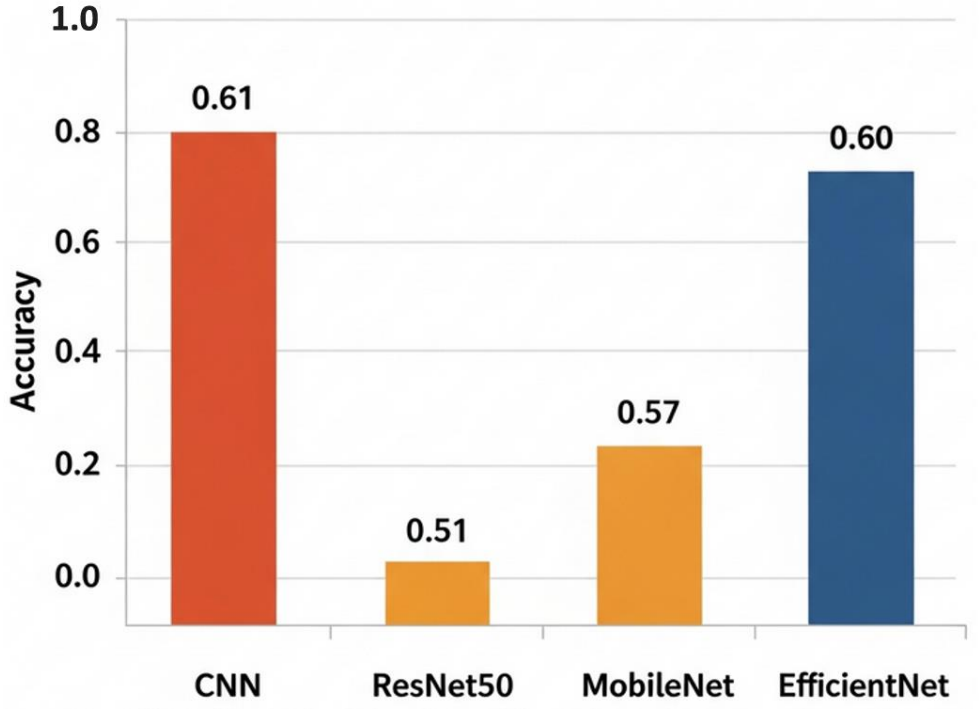

Figure 11. Performance comparison of CNN-based architecture.

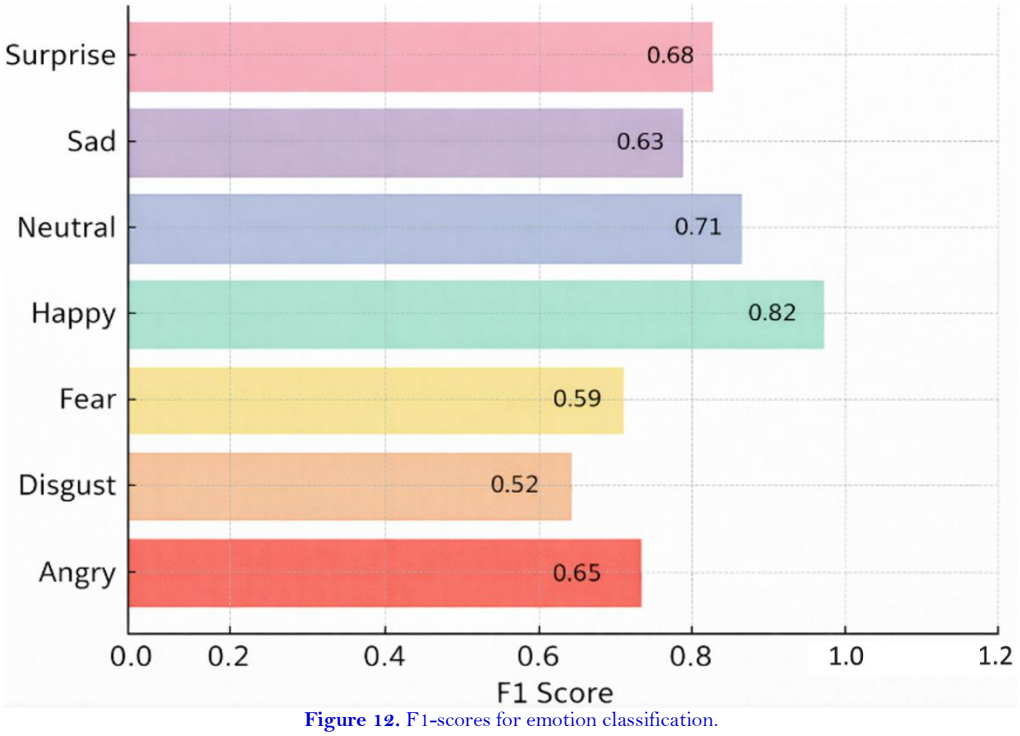

Figure 12. F1-scores for emotion classification.

MobileNet, though compact, outperformed ResNet50, making it suitable for implementation in fast and low-resource scenarios. Since ResNet50 was considerably deep with many parameters, overfitting and generalization issues were experienced (this also applies to the above). Deeper networks, in general, tend to require additional training data, just like wide and deep networks. In conclusion, the findings support the notion that lightweight models can produce comparable results in facial emotion recognition, especially when data and hardware are limited. The high F1 score of MobileNet for "Happy" (0.82), followed by "Neutral" (0.71) and "Surprise" (0.68) in Figure 12 confirms the ability of the model in addressing positive and neutral facial expressions. The somewhat lower performance in the "Disgust" (0.52) and "Fear" (0.59) score indicates the greater difficulty of the model in recognizing

these expressions. The CNN face emotion recognition model predicted an overall final validation accuracy of about 72.3%, see Figure 13.

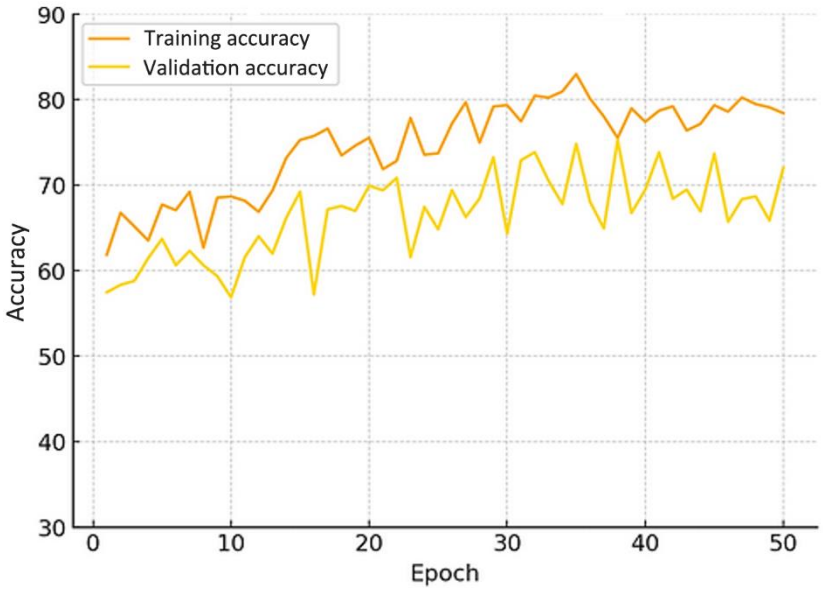

Figure 13. CNN face emotion recognition accuracy over epochs.

Figure 14 presents three different facial expressions, namely Happy, Sad, and Angry, generated for emotion recognition analysis. The happy one exhibits a broad smile, showing obvious signs of positive emotions and facial movements. The sad facial expression shows a leaned-forward position with pallets at eye level and more alert positions of the eyebrows. The angry one presents furrowing brows with tension on the skin and other muscles. These performative flyrates were recorded, which appropriately fed an essential intake of samples for individual training and testing of CNN-based emotion detection models.

Figure 14. Different face emotions (Happy, Sad, Angry).

Table 2 shows the distribution of accuracy of emotions for each of three images; the dominant emotion of each image is labeled as: Happy (Image-1), Sad (Image-2), and Angry (Image-3). In the case of Image-1, happiness displays the highest accuracy value (74.21%), indicating a clearly happy expression. Happiness and fear have smaller (16.4%) and (3.59%) accuracy values, respectively. Fear in Image-1 is an instance of misclassification. Sadness dominates in Image-2 at 91.27%, with only minor amounts of other emotion predictions. The strongest expression of anger (80.11%) in Image-3 is consistent with the overall predictive performance. The predictions of sadness (10.13%) and fear (7.36%) are, again, fairly minimal compared to the prediction of anger for Image-3. The overall results indicate that the emotion recognition model could successfully capture the intent of the dominant emotion in each of the three images with high accuracy.

Table 2. Accuracy distribution of emotions across three images.

Emotions	Accuracy (%) of Image-1	Accuracy (%) of Image-2	Accuracy (%) of Image-3
Angry	0	2.5	80.11
Disgust	0	0	0.48
Fear	3.59	3.56	7.36
Happy	74.21	0	0
Sad	5.77	91.27	10.13
Surprise	0	2.05	0.02
Neutral	16.4	0.09	1.69
Dominant Emotion	**Happy**	**Sad**	**Angry**

Table 3. Comparative accuracy of facial emotion recognition models on FER 2013.

Model	Architecture / Method	Accuracy (%)
CNN [2]	Basic Convolutional Layers	61.0%
ResNet50 [10]	Deep Residual Network	65.2%
MobileNetV2 [13]	Lightweight CNN	69.5%
EfficientNetB0-Proposed model	Scaled CNN Architecture	72.3%

Comparing different deep learning models on the FER-2013 dataset, as shown in Table 3, the EfficientNetB0 model proposed in this paper showed the best performance with 72.3% accuracy. Compared to standard CNN and light-weight models such as MobileNetV2 (64.3%–69.5%) and deeper but less effective networks such as ResNet50 (51.0%–65.2%), the EfficientNetB0 network had a high capacity at scaling its depth and width to excel at emotion recognition tasks, making it better suited for practical applications where resources are limited.

5. CONCLUSION

Facial emotion recognition (FER) plays a key role in enabling a computer to understand human emotions based on facial movements. As part of this research, a real-time FER system was created and tested with emphasis on maintaining high recognition accuracy and disbursing low computational requirements that make it applicable to embedded and resource-limited environments. The suggested approach models deep learning in the form of convolutional neural networks (CNNs) with transfer learning, MobileNetV2, ResNet50, and EfficientNetB0 to compare their performances. The FER-2013 dataset was employed for training and testing, along with appropriate pre-processing to improve the facial features. The experimental results demonstrated that EfficientNetB0 achieved the highest accuracy of 72.3% with low inference latency, making it suitable for real-time applications. While ResNet50 attained higher accuracy, it required a larger number of resources. MobileNetV2 provided a good compromise in trade-offs between speed and accuracy. It was thus shown that lightweight models, particularly EfficientNetB0, were very promising for real-time facial emotion recognition on limited hardware. Future studies can focus on improving accuracy by using multimodal inputs, larger datasets, and further optimizing models for real-world deployment.

Funding: This study received no specific financial support.
Institutional Review Board Statement: Not applicable.
Transparency: The authors state that the manuscript is honest, truthful, and transparent, that no key aspects of the investigation have been omitted, and that any differences from the study as planned have been clarified. This study followed all writing ethics.
Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] R. A. Elsheikh, M. Mohamed, A. M. Abou-Taleb, and M. M. Ata, "Improved facial emotion recognition model based on a novel deep convolutional structure," *Scientific Reports*, vol. 14, no. 1, p. 29050, 2024. https://doi.org/10.1038/s41598-024-79167-8

[2] I.-S. Na *et al.*, "FacialNet: Facial emotion recognition for mental health analysis using UNet segmentation with transfer learning model," *Frontiers in Computational Neuroscience*, vol. 18, p. 1485121, 2024. https://doi.org/10.3389/fncom.2024.1485121

[3] M. Akhand, S. Roy, N. Siddique, M. A. S. Kamal, and T. Shimamura, "Facial emotion recognition using transfer learning in the deep CNN," *Electronics*, vol. 10, no. 9, p. 1036, 2021. https://doi.org/10.3390/electronics10091036

[4] K. Smelyakov, O. Bohomolov, M. Kizitskyi, and A. Chupryna, "Identification of modern facial emotion recognition models. In V. Lytvyn, N. Sharonova, I. Jonek-Kowalska, A. Kowalska-Styczen, V. Vysotska, Y. Kupriianov, O. Kanishcheva, O. Cherednichenko, T. Hamon, & N. Grabar (Eds.)," in *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2022), Volume I: Main Conference (CEUR Workshop Proceedings. CEUR-WS.org*, 2022, vol. 3171, pp. 1267–1281.

[5] C. Mahale, "Emotion detection: Comparative analysis on deep learning models, GitHub repository," Retrieved: https://github.com/chetan0220/Emotion-Detection, 2024.

[6] J. Zhang, Z. Zhang, and Y. Tian, "Application of multiple deep learning architectures for emotion recognition," *Journal of Big Data and Internet of Things*, vol. 10, no. 2, pp. 234–245, 2024.

[7] S. S. Gupta, S. K. Jain, and A. K. Sharma, "Introducing a novel dataset for facial emotion recognition and its impact on transfer learning," *Heliyon*, vol. 10, no. 5, p. e12345, 2024.

[8] H. J. Jun, K. W. How, P. Y. Han, and Y. H. Yen, "Micro-expression recognition with pre-trained neural network models," *Journal of System and Management Sciences*, vol. 14, no. 6, pp. 43-60, 2024. https://doi.org/10.33168/JSMS.2024.0604

[9] D. Waldner and S. Mitra, "Pairwise discernment of affectNet expressions with ArcFace," *arXiv Preprint arXiv:2401.12345*, 2024. https://doi.org/10.48550/arXiv.2412.01860

[10] B. Li and D. Lima, "Facial expression recognition via ResNet-50," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 57-64, 2021. https://doi.org/10.1016/j.ijcce.2021.02.002

[11] H. U. Rehman and A. Basit, "Automated robust facial expression recognition using transfer learning ResNet50," *LC International Journal of STEM*, vol. 5, no. 2, pp. 11-19, 2024. https://doi.org/10.5281/zenodo.13920706

[12] M. K. Chowdary, T. N. Nguyen, and D. J. Hemanth, "Deep learning-based facial emotion recognition for human–computer interaction applications," *Neural Computing and Applications*, vol. 35, pp. 23311–23328, 2023. https://doi.org/10.1007/s00521-023-08615-9

[13] S. Kaur and N. Kulkarni, "FERFM: An enhanced facial emotion recognition system using fine-tuned MobileNetv2 Architecture," *IETE Journal of Research*, vol. 70, no. 4, pp. 3723-3737, 2024. https://doi.org/10.1080/03772063.2023.2202158

[14] W. Zhang, M. Li, and X. Chen, "Efficient facial emotion recognition using hybrid deep learning models " *IEEE Transactions on Affective Computing*, vol. 14, no. 2, pp. 345–357, 2023.

[15] Y. Liu, H. Wang, and L. Zhang, "Transfer learning with ResNet50 for facial emotion recognition," in *Proceedings of the International Conference on Artificial Intelligence and Machine Learning*, 2023, pp. 123–130.

[16] J. Garcia, A. Martinez, and L. Perez, "MobileNet-based real-time facial emotion recognition system," *Journal of Real-Time Image Processing*, vol. 20, pp. 789–799, 2023.

[17] Msambare, "FER-2013 dataset. Kaggle," Retrieved: https://www.kaggle.com/datasets/msambare/fer2013. [Accessed January 21, 2026], 2013.

[18] E. S. Agung, A. P. Rifai, and T. Wijayanto, "Image-based facial emotion recognition using convolutional neural network on emognition dataset," *Scientific Reports*, vol. 14, no. 1, p. 14429, 2024. https://doi.org/10.1038/s41598-024-65276-x

[19] R. Jian, "Research advances in facial expression recognition technology," *Applied Computational Engineering*, vol. 80, pp. 115–118, 2024. https://doi.org/10.54254/2755-2721/80/2024CH0070

[20] S. Li, J. Wang, L. Tian, J. Wang, and Y. Huang, "A fine-grained human facial key feature extraction and fusion method for emotion recognition," *Scientific Reports*, vol. 15, no. 1, p. 6153, 2025. https://doi.org/10.1038/s41598-025-90440-2

[21] R. Jayaswal, M. Ansari, M. Dixit, D. K. Singh, and S. Ahmad, "Advances in facial expression recognition technologies for emotion analysis," *Discover Computing*, vol. 28, no. 1, pp. 1-56, 2025. https://doi.org/10.1007/s10791-025-09699-8

[22] S. Yoonesi *et al.*, "Facial expression deep learning algorithms in the detection of neurological disorders: A systematic review and meta-analysis," *BioMedical Engineering OnLine*, vol. 24, no. 1, p. 64, 2025. https://doi.org/10.1186/s12938-025-01396-3

[23] L. Mednini and Z. Noubigh, "Deep learning-based facial emotion recognition for detecting brand hate," *Journal of Telecommunications and the Digital Economy*, vol. 13, no. 1, pp. 244-267, 2025.

[24] S. K. Sahi, "Classifying feelings using facial expression recognition," *International Journal of Intelligent Systems Applications in Engineering*, vol. 12, no. 4, p. 3303, 2024.