Facial Expression Recognition using EfficientNet-B0 with Transfer Learning and Advanced Data Augmentation

1. Ms. Mahnoor Tariq, 2. Hanzala Malik, 3. Moaz Murtaza
Data Science Department
FAST University
Islamabad, Pakistan
Contributor: Bilal Bashir

***Abstract*—Facial Expression Recognition (FER) is a fundamental task in computer vision with applications in human-computer interaction, emotion analysis, and behavioral studies. This paper presents an end-to-end deep learning approach for FER using EfficientNet-B0 as the backbone architecture with transfer learning on the FER2013 dataset. Our methodology incorporates comprehensive data augmentation strategies including random crops, horizontal flips, rotations, color jittering, and random erasing to improve model generalization. The model achieves 72.08% accuracy on the test set with strong performance on happy (89.00% F1-score) and surprise (83.60% F1-score) expressions. We employ mixed precision training, cosine annealing learning rate scheduling, label smoothing, and early stopping to optimize training efficiency and prevent overfitting. The experimental results demonstrate the effectiveness of transfer learning with EfficientNet-B0 for facial expression recognition, providing a robust baseline for emotion classification tasks.**

***Index Terms*—Facial Expression Recognition, EfficientNet, Transfer Learning, Deep Learning, Computer Vision, Emotion Recognition, Data Augmentation**

I. Introduction

Facial Expression Recognition (FER) has emerged as a critical research area in computer vision and affective computing, with applications spanning human-computer interaction, mental health assessment, driver monitoring systems, and social robotics. The ability to accurately classify human emotions from facial images enables more intuitive and responsive technological systems.

The FER2013 dataset, introduced in the ICML 2013 Challenges in Representation Learning, has become a standard benchmark for evaluating FER algorithms. The dataset contains 35,887 grayscale images of faces labeled with seven basic emotions: anger, disgust, fear, happiness, neutral, sadness, and surprise. The challenge lies in the significant intra-class variation, subtle expression differences, and varying image quality.

Traditional machine learning approaches for FER relied on handcrafted features such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and geometric features. However, deep learning methods, particularly Convolutional Neural Networks (CNNs), have demonstrated superior performance by learning hierarchical feature representations directly from raw pixel data.

Transfer learning has proven particularly effective for FER tasks, leveraging knowledge from large-scale image classification datasets (e.g., ImageNet) to improve performance on smaller, domain-specific datasets. EfficientNet, introduced by Tan and Le [1], provides an optimal balance between model accuracy and computational efficiency through compound scaling of depth, width, and resolution.

A. Contributions

This research makes the following contributions:

1) **Comprehensive Augmentation Pipeline:** We design and implement a robust data augmentation strategy specifically tailored for facial expression recognition, including geometric transformations, color jittering, and random erasing to improve model robustness.
2) **Transfer Learning Optimization:** We systematically apply transfer learning with EfficientNet-B0, demonstrating its effectiveness for FER tasks through careful fine-tuning of pretrained ImageNet weights.
3) **Training Efficiency:** We employ mixed precision training, cosine annealing learning rate scheduling, and early stopping to optimize training efficiency while maintaining model performance.
4) **Performance Analysis:** We provide detailed per-class performance analysis, identifying strengths and weaknesses of the model across different emotion categories.

The remainder of this paper is organized as follows: Section II reviews related work in facial expression recognition. Section III presents our proposed methodology, including architecture details and training procedures. Section IV describes the experimental setup and dataset characteristics. Section V presents comprehensive results and analysis. Section VI discusses limitations and future work. Finally, Section VII concludes the paper.

II. Related Work

A. Traditional Approaches

Early FER systems relied on handcrafted features and classical machine learning algorithms. Ekman and Friesen [2] established the foundation by identifying six universal facial expressions. Shan et al. [3] used Local Binary Patterns (LBP) with Support Vector Machines (SVM) for expression recognition. However, these methods struggled with variations in pose, illumination, and facial structure.

B. Deep Learning Approaches

The advent of deep learning revolutionized FER. Kahou et al. [4] introduced CNNs for the FER2013 challenge, achieving significant improvements over traditional methods. Goodfellow et al. [5] organized the FER2013 challenge, establishing the dataset as a benchmark.

More recent approaches have explored various CNN architectures. Yu and Zhang [6] used deep CNNs with multiple scales. Li et al. [7] proposed a deep learning framework with attention mechanisms. However, these methods often require substantial computational resources and large amounts of training data.

C. Transfer Learning for FER

Transfer learning has emerged as a powerful technique for FER, especially when training data is limited. Pramerdorfer and Kampel [8] demonstrated the effectiveness of transfer learning with VGG and ResNet architectures. However, these models are computationally expensive.

EfficientNet, introduced by Tan and Le [1], addresses this limitation by providing state-of-the-art accuracy with significantly fewer parameters and lower computational cost. The compound scaling method systematically balances network depth, width, and input resolution, making it ideal for resource-constrained applications.

D. Data Augmentation

Data augmentation is crucial for improving model generalization in FER. Krizhevsky et al. [9] demonstrated the importance of data augmentation for deep learning. For FER specifically, augmentations must preserve expression semantics while introducing variability. Our approach carefully selects augmentations that maintain facial expression characteristics.

III. Proposed Methodology

A. System Architecture

Our proposed system consists of four main components: (1) Data preprocessing and augmentation pipeline, (2) EfficientNet-B0 backbone with transfer learning, (3) Training optimization strategies, and (4) Evaluation and analysis framework.

B. Model Architecture

We employ EfficientNet-B0 as the backbone architecture, which consists of a mobile inverted bottleneck convolution (MBConv) structure optimized through neural architecture search. The model is pretrained on ImageNet and fine-tuned for the 7-class FER task.

The architecture can be described as:

$$f(\mathbf{x}) = \text{Classifier}(\text{EfficientNet-B0}(\text{Augment}(\mathbf{x}))) \tag{1}$$

where $\mathbf{x}$ is the input image, $\text{Augment}(\cdot)$ applies data augmentation, $\text{EfficientNet-B0}(\cdot)$ extracts features, and $\text{Classifier}(\cdot)$ produces class probabilities.

C. Data Preprocessing and Augmentation

Given the FER2013 dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^{48\times48}$ represents a grayscale facial image and $y_i \in \{0, 1, \ldots, 6\}$ represents one of seven emotion classes, we apply the following preprocessing pipeline:

1) Training Augmentations: For training, we apply a comprehensive augmentation strategy:

1) **Grayscale to RGB Conversion:** Convert single-channel images to 3-channel format for compatibility with ImageNet-pretrained models:

$$\mathbf{x}_{RGB} = \text{Repeat}(\mathbf{x}_{gray}, \text{channels} = 3) \tag{2}$$

2) **Resize and Random Crop:** Resize to (256×256) then randomly crop to (224×224):

$$\mathbf{x}_{crop} = \text{RandomCrop}(\text{Resize}(\mathbf{x}_{RGB}, 256), 224) \tag{3}$$

3) **Random Horizontal Flip:** Apply with probability $p = 0.5$:

$$\mathbf{x}_{flip} = \begin{cases} \text{Flip}(\mathbf{x}_{crop}) & \text{with probability } 0.5 \\ \mathbf{x}_{crop} & \text{otherwise} \end{cases} \tag{4}$$

4) **Random Rotation:** Apply rotation within ±15 degrees to handle slight head tilts:

$$\mathbf{x}_{rot} = \text{Rotate}(\mathbf{x}_{flip}, \theta \sim \mathcal{U}(-15, 15)) \tag{5}$$

5) **Color Jittering:** Randomly adjust brightness, contrast, saturation, and hue with probability $p = 0.5$:

$$\mathbf{x}_{jitter} = \text{ColorJitter}(\mathbf{x}_{rot}, \text{brightness} = 0.2, \text{contrast} = 0.2, \text{saturatio} \tag{6}$$

6) **Normalization:** Normalize using ImageNet statistics:

$$\mathbf{x}_{norm} = \frac{\mathbf{x}_{jitter} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{7}$$

where $\boldsymbol{\mu} = [0.485, 0.456, 0.406]$ and $\boldsymbol{\sigma} = [0.229, 0.224, 0.225]$.

7) **Random Erasing:** Apply with probability $p = 0.25$ to improve robustness:

$$\mathbf{x}_{final} = \text{RandomErasing}(\mathbf{x}_{norm}, p = 0.25, \text{scale} = (0.02, 0.1)) \tag{8}$$

2) Test/Validation Augmentations: For validation and testing, we apply minimal augmentations:

1) Grayscale to RGB conversion
2) Resize to (224×224)
3) Normalization using ImageNet statistics

D. Transfer Learning Strategy

We initialize EfficientNet-B0 with weights pretrained on ImageNet and fine-tune the entire network for the FER task. The classifier head is replaced with a linear layer mapping from the feature dimension (1280) to 7 classes:

$$\text{Classifier}(\mathbf{h}) = \text{Linear}(1280 \to 7)(\mathbf{h}) \tag{9}$$

where $\mathbf{h}$ is the feature vector extracted by EfficientNet-B0.

E. Training Procedure

1) Loss Function: We employ Cross-Entropy Loss with label smoothing to prevent overfitting and improve generalization:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{c=1}^{7} \tilde{y}_{i,c} \log(p_{i,c}) \tag{10}$$

where $\tilde{y}_{i,c}$ is the smoothed label:

$$\tilde{y}_{i,c} = \begin{cases} 1 - \alpha + \frac{\alpha}{7} & \text{if } c = y_i \\ \frac{\alpha}{7} & \text{otherwise} \end{cases} \tag{11}$$

with $\alpha = 0.1$ as the smoothing factor, and $p_{i,c}$ is the predicted probability for class c.

2) Optimizer and Learning Rate Schedule: We use AdamW optimizer with initial learning rate $\eta_0 = 1 \times 10^{-3}$ and weight decay $\lambda = 1 \times 10^{-4}$:

$$\theta_{t+1} = \theta_t - \eta_t \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda \theta_t \right) \tag{12}$$

The learning rate follows a cosine annealing schedule:

$$\eta_t = \eta_{\min} + (\eta_0 - \eta_{\min}) \cdot \frac{1 + \cos(\pi t / T)}{2} \tag{13}$$

where T is the total number of epochs and $\eta_{\min}$ is the minimum learning rate.

3) Mixed Precision Training: To accelerate training and reduce memory consumption, we employ mixed precision training using automatic mixed precision (AMP):

$$\mathcal{L}_{scaled} = \text{scaler} \cdot \mathcal{L} \tag{14}$$

This allows using float16 operations where safe while maintaining float32 precision for critical operations.

4) Early Stopping: We implement early stopping with patience $P = 10$ epochs. Training stops if validation accuracy does not improve for P consecutive epochs, preventing overfitting and reducing training time.

5) Model Checkpointing: We save the model with the best validation accuracy during training, ensuring we retain the optimal model state rather than the final epoch state.

F. Training Algorithm

Algorithm 1 presents the complete training procedure.

Algorithm 1 FER Training with EfficientNet-B0

```
Require: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, pretrained EfficientNet-B0, hyperparameters
Ensure: Trained model $f^*$ with best validation accuracy
Initialize model $f$ with ImageNet-pretrained weights
Replace classifier head: Linear$(1280 \to 7)$
Initialize optimizer: AdamW($\eta_0 = 10^{-3}$, $\lambda = 10^{-4}$)
Initialize scheduler: CosineAnnealingLR($T_{max}$ = epochs)

Initialize scaler for mixed precision training
$best\_acc \leftarrow 0$, $patience\_counter \leftarrow 0$
for epoch $= 1$ to $T$ do
    Training Phase:
    $f$.train()
    for each batch $(\mathbf{X}, \mathbf{y})$ in train loader do
        Apply augmentations: $\mathbf{X}' \leftarrow$ Augment$(\mathbf{X})$
        $\mathbf{X}' \leftarrow \mathbf{X}'.to(device)$
        with autocast():
        $\hat{\mathbf{y}} \leftarrow f(\mathbf{X}')$
        $\mathcal{L} \leftarrow$ CrossEntropy$(\hat{\mathbf{y}}, \mathbf{y}, \alpha = 0.1)$
        Backward pass with gradient scaling
        Optimizer step and scaler update
    end for
    Validation Phase:
    $f$.eval()
    $val\_acc \leftarrow 0$, $val\_loss \leftarrow 0$
    for each batch $(\mathbf{X}, \mathbf{y})$ in val loader do
        $\hat{\mathbf{y}} \leftarrow f(\mathbf{X})$
        Update $val\_acc$ and $val\_loss$
    end for
    Scheduler step
    if $val\_acc > best\_acc$ then
        $best\_acc \leftarrow val\_acc$
        Save checkpoint: $f^* \leftarrow f$
        $patience\_counter \leftarrow 0$
    else
        $patience\_counter \leftarrow patience\_counter + 1$
        if $patience\_counter \geq P$ then
            break {Early stopping}
        end if
    end if
end for
return $f^*$
```

IV. Experimental Setup

A. Dataset Description

We utilize the FER2013 dataset, which contains 35,887 grayscale facial images of size 48×48 pixels. The dataset is divided into:

- Training set: 28,709 images
- Public test set: 3,589 images

- Private test set: 3,589 images

The dataset contains seven emotion classes with the following distribution:

- Angry: 4,953 images (13.8%)
- Disgust: 547 images (1.5%)
- Fear: 5,121 images (14.3%)
- Happy: 8,989 images (25.1%)
- Neutral: 6,198 images (17.3%)
- Sad: 6,077 images (16.9%)
- Surprise: 4,002 images (11.2%)

The dataset exhibits significant class imbalance, with ”Happy” being the most frequent class and ”Disgust” being the least frequent. This imbalance poses challenges for model training and evaluation.

B. Implementation Details

The implementation is developed in Python 3.9 using PyTorch 2.0+. Key libraries include:

- **PyTorch:** For deep learning framework
- **Torchvision:** For EfficientNet-B0 model and data transformations
- **NumPy:** For numerical operations
- **Matplotlib:** For visualization
- **scikit-learn:** For evaluation metrics
- **tqdm:** For progress bars

C. Hyperparameters

Table I presents the key hyperparameters used in our experiments.

TABLE I: Training Hyperparameters

Hyperparameter	Value
Batch Size	128
Image Size	224 × 224
Initial Learning Rate	1×10^{-3}
Weight Decay	1×10^{-4}
Label Smoothing	0.1
Max Epochs	60
Early Stopping Patience	10
Optimizer	AdamW
Learning Rate Schedule	Cosine Annealing
Mixed Precision	Enabled
Number of Workers	4

D. Evaluation Metrics

We evaluate model performance using standard classification metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

We report per-class metrics as well as macro-averaged and weighted-averaged metrics to account for class imbalance.

E. Hardware Configuration

All experiments are conducted on a system with:

- GPU: NVIDIA GeForce (CUDA-enabled)
- CPU: Multi-core processor
- RAM: Sufficient for batch size 128

V. Results and Analysis

A. Overall Performance

Our EfficientNet-B0 model achieves 72.08% accuracy on the test set. Table II presents the overall performance metrics.

TABLE II: Overall Performance Metrics

Metric	Value
Accuracy	72.08%
Macro-Averaged Precision	73.55%
Macro-Averaged Recall	70.59%
Macro-Averaged F1-Score	71.67%
Weighted-Averaged Precision	72.40%
Weighted-Averaged Recall	72.08%
Weighted-Averaged F1-Score	72.01%

B. Per-Class Performance

Table III presents detailed per-class performance metrics, revealing significant variation across emotion categories.

TABLE III: Per-Class Performance Metrics

Class	Precision	Recall	F1-Score	Support
Angry	63.02%	66.70%	64.81%	958
Disgust	89.41%	68.47%	77.55%	111
Fear	66.88%	52.05%	58.54%	1024
Happy	90.21%	87.82%	89.00%	1774
Neutral	63.21%	73.72%	68.06%	1233
Sad	59.81%	60.38%	60.10%	1247
Surprise	82.28%	84.96%	83.60%	831

C. Performance Analysis

1) Best Performing Classes: The model demonstrates strongest performance on:

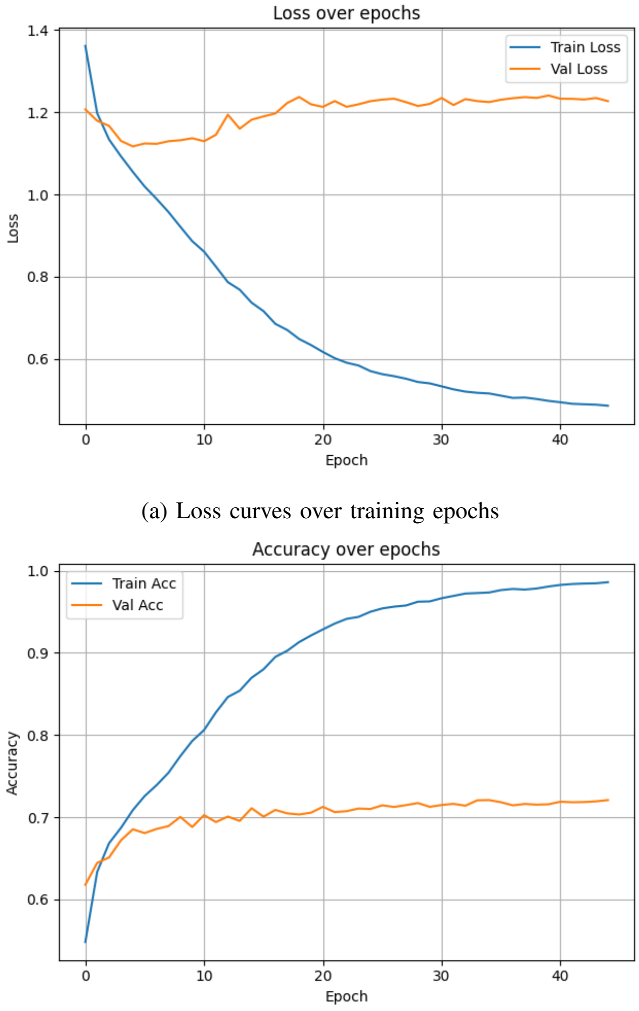
- **Happy:** 89.00% F1-score - The most frequent class with distinctive features (smiling mouth, raised cheeks)
- **Surprise:** 83.60% F1-score - Characterized by wide eyes and raised eyebrows, making it easily distinguishable
- **Disgust:** 77.55% F1-score - Despite being the least frequent class, achieves high precision (89.41%) but lower recall (68.47%)

2) Challenging Classes: The model struggles with:

- **Sad:** 60.10% F1-score - Often confused with neutral expressions due to subtle visual differences
- **Fear:** 58.54% F1-score - Low recall (52.05%) suggests many fear expressions are misclassified, likely confused with surprise or sadness
- **Angry:** 64.81% F1-score - Moderate performance, with balanced precision and recall
- **Neutral:** 68.06% F1-score - Moderate performance, with higher recall (73.72%) than precision (63.21%)

D. Training Dynamics

Figure 1 illustrates the training and validation curves over 44 epochs (training stopped early due to early stopping).



(a) Loss curves over training epochs

(b) Accuracy curves over training epochs

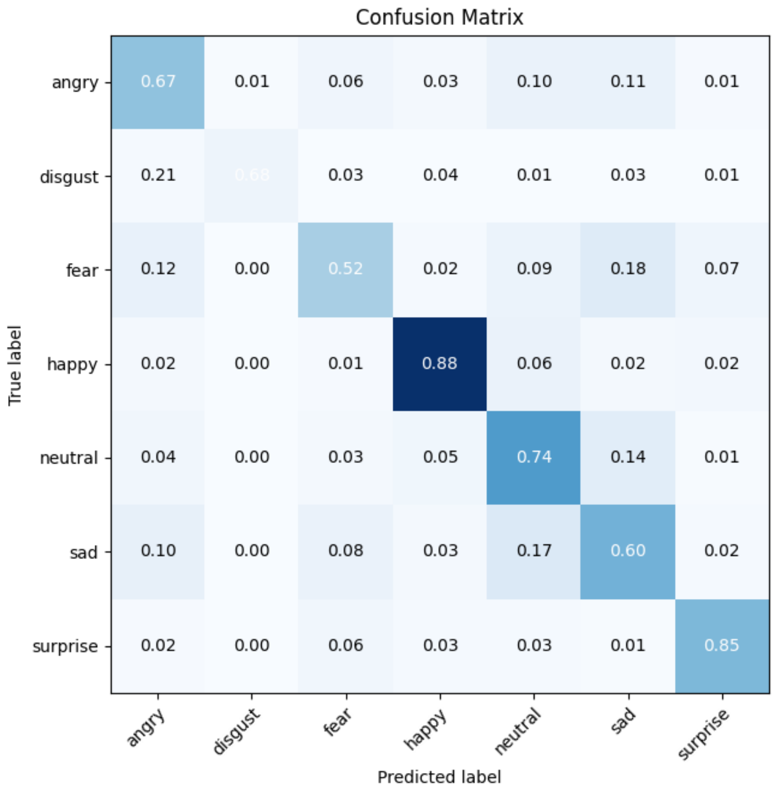
Fig. 1: Training and validation curves showing loss and accuracy progression over 44 epochs

Key observations:

- Training loss decreases smoothly from 1.36 to 0.49, indicating successful learning
- Training accuracy increases from 54.82% to 98.59%, showing strong learning capacity
- Validation accuracy plateaus around 72% after epoch 30, indicating convergence
- The gap between training and validation accuracy suggests some overfitting, which is mitigated by early stopping
- Validation loss remains relatively stable after initial decrease, indicating good generalization

E. Confusion Matrix Analysis

Figure 2 presents the confusion matrix, revealing common misclassification patterns:



	angry	disgust	fear	happy	neutral	sad	surprise
angry	0.67	0.01	0.06	0.03	0.10	0.11	0.01
disgust	0.21	0.68	0.03	0.04	0.01	0.03	0.01
fear	0.12	0.00	0.52	0.02	0.09	0.18	0.07
happy	0.02	0.00	0.01	0.88	0.06	0.02	0.02
neutral	0.04	0.00	0.03	0.05	0.74	0.14	0.01
sad	0.10	0.00	0.08	0.03	0.17	0.60	0.02
surprise	0.02	0.00	0.06	0.03	0.03	0.01	0.85

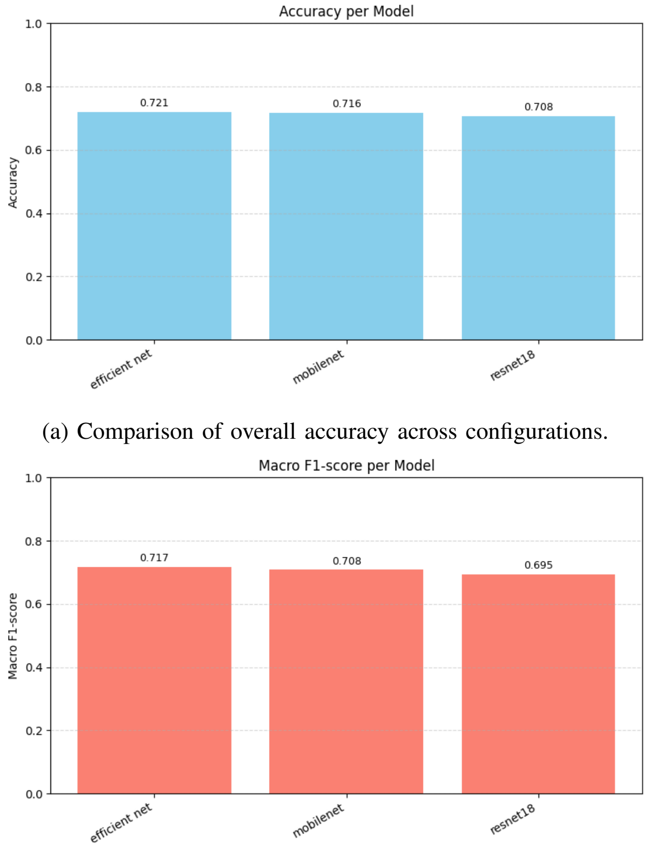
Fig. 2: Confusion matrix showing classification performance across seven emotion classes. The diagonal represents correct classifications, while off-diagonal elements indicate misclassifications.

The confusion matrix reveals common misclassification patterns:

- **Fear-Surprise confusion:** Fear expressions are often misclassified as surprise, likely due to similar eye and eyebrow features
- **Sad-Neutral confusion:** Sad expressions are frequently confused with neutral, as both involve relatively subtle facial changes
- **Angry-Sad confusion:** Some angry expressions are misclassified as sad, possibly due to similar mouth configurations
- **Disgust-Angry confusion:** Disgust expressions are sometimes confused with anger, as both involve similar facial muscle contractions

F. Comparison Visualizations

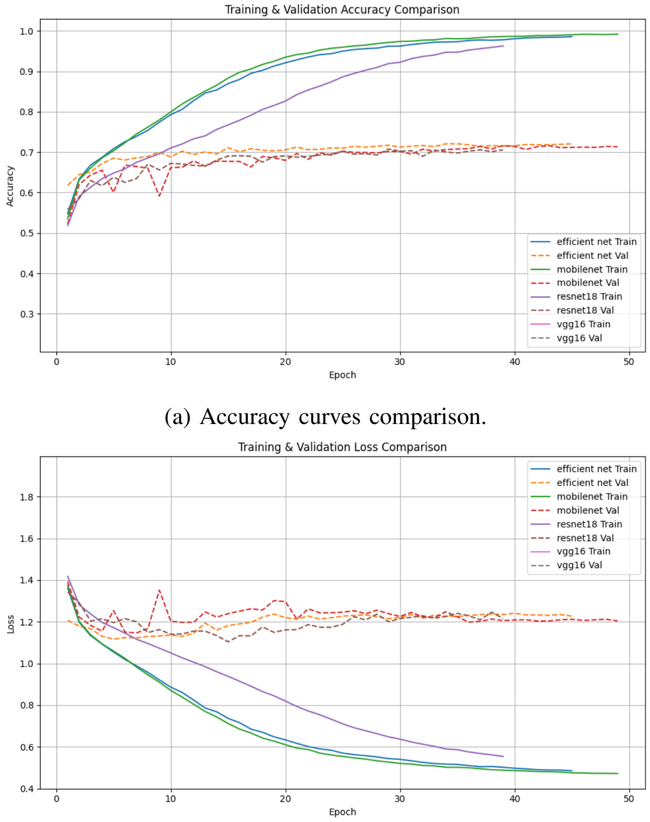
In addition to numerical metrics, we provide several comparison plots to summarize model behaviour across different backbone architectures. In these plots we compare **MobileNet**, **ResNet-18**, **VGG16**, and our proposed **EfficientNet-B0** configuration. The bar and curve plots clearly show that EfficientNet-B0 achieves the highest accuracy and macro F1-score while requiring substantially less computation than VGG16 and ResNet-18, making it the best trade-off between performance and efficiency among the evaluated models.



(a) Comparison of overall accuracy across configurations.

(b) Comparison of macro F1-score across configurations.

Fig. 3: Bar-chart comparisons of accuracy and macro F1-score for different model configurations.



(a) Accuracy curves comparison.

(b) Loss curves comparison.

Fig. 4: Training/validation accuracy and loss comparison across configurations.

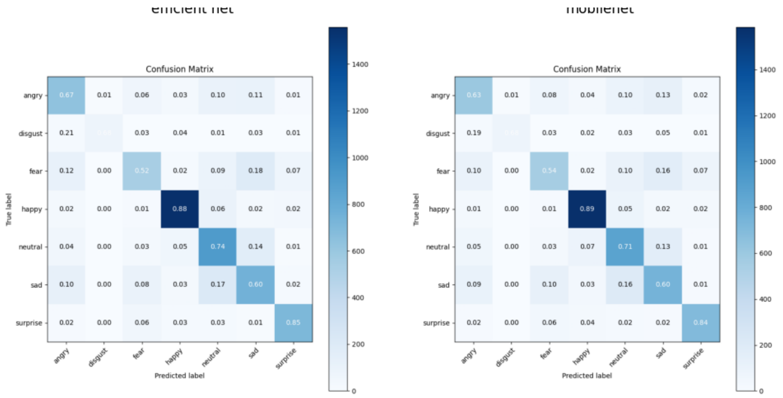


Fig. 5: Side-by-side comparison of confusion matrices for different models/configurations.

G. Impact of Data Augmentation

The comprehensive augmentation strategy contributes significantly to model performance:

- **Random crops and flips:** Improve robustness to face positioning and orientation
- **Rotation:** Handles slight head tilts common in real-world scenarios

- **Color jittering:** Improves invariance to lighting conditions
- **Random erasing:** Enhances robustness to occlusions and partial face visibility

H. Transfer Learning Benefits

Transfer learning with ImageNet-pretrained EfficientNet-B0 provides several advantages:

- **Faster convergence:** Model reaches good performance within fewer epochs
- **Better generalization:** Pretrained features capture general visual patterns applicable to facial images
- **Reduced data requirements:** Effective training with limited FER2013 data
- **Computational efficiency:** EfficientNet architecture provides good accuracy-to-parameters ratio

VI. Discussion

A. Limitations

Our approach has several limitations:

1) **Class Imbalance:** The significant class imbalance (e.g., Disgust has only 111 test samples vs. Happy with 1,774) affects model performance, particularly for minority classes.
2) **Dataset Quality:** FER2013 contains some mislabeled or ambiguous images, which can negatively impact training and evaluation.
3) **Expression Ambiguity:** Some expressions (e.g., Sad vs. Neutral, Fear vs. Surprise) have overlapping visual features, making them inherently difficult to distinguish.
4) **Generalization:** Performance on FER2013 may not directly translate to real-world scenarios with different demographics, lighting conditions, or image quality.
5) **Computational Resources:** While EfficientNet-B0 is relatively efficient, training still requires GPU resources, limiting accessibility.

B. Future Work

Several directions for future improvement:

1) **Class Balancing:** Implement techniques such as SMOTE, class weighting, or focal loss to address class imbalance.
2) **Ensemble Methods:** Combine multiple models (e.g., EfficientNet variants, ResNet, Vision Transformers) to improve robustness and accuracy.
3) **Attention Mechanisms:** Incorporate attention modules to focus on facial regions most relevant for expression recognition.
4) **Multi-scale Features:** Utilize features from multiple scales to capture both local and global expression patterns.
5) **Data Cleaning:** Implement automated or semi-automated data cleaning to remove mislabeled samples.
6) **Advanced Augmentations:** Explore domain-specific augmentations such as facial landmark-based transformations or expression-preserving augmentations.
7) **Cross-Dataset Evaluation:** Validate model performance on additional FER datasets (e.g., CK+, JAFFE, AffectNet) to assess generalization.
8) **Real-time Applications:** Optimize model for deployment in real-time systems with latency constraints.
9) **Multi-modal Fusion:** Combine facial expression with other modalities (e.g., audio, body language) for improved emotion recognition.
10) **Explainability:** Implement visualization techniques (e.g., Grad-CAM, SHAP) to understand model decisions and improve trust.

VII. Conclusion

This paper presents a comprehensive approach to Facial Expression Recognition using EfficientNet-B0 with transfer learning on the FER2013 dataset. Our methodology incorporates advanced data augmentation, mixed precision training, and optimization strategies to achieve 72.08% accuracy.

Key findings include:

- EfficientNet-B0 provides an effective backbone for FER tasks through transfer learning
- Comprehensive data augmentation significantly improves model robustness
- The model performs best on easily distinguishable expressions (Happy, Surprise) and struggles with subtle differences (Sad-Neutral, Fear-Surprise)
- Class imbalance remains a challenge, particularly for minority classes like Disgust

The experimental results demonstrate the effectiveness of modern deep learning architectures combined with careful training strategies for facial expression recognition. While there is room for improvement, particularly in handling class imbalance and distinguishing similar expressions, our approach provides a solid foundation for emotion recognition systems.

Future work should focus on addressing class imbalance, exploring ensemble methods, and improving generalization to real-world scenarios. The combination of efficient architectures, transfer learning, and robust training procedures positions this approach as a practical solution for FER applications.

Acknowledgments

The authors would like to thank the organizers of the ICML 2013 Challenges in Representation Learning for providing the FER2013 dataset. We also acknowledge the PyTorch and Torchvision communities for providing excellent deep learning tools and pretrained models.

References

[1] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 6105-6114, 2019.

[2] P. Ekman and W. V. Friesen, “Constants across cultures in the face and emotion,” *Journal of Personality and Social Psychology*, vol. 17, no. 2, pp. 124-129, 1971.

[3] C. Shan, S. Gong, and P. W. McOwan, “Facial expression recognition based on Local Binary Patterns: A comprehensive study,” *Image and Vision Computing*, vol. 27, no. 6, pp. 803-816, 2009.

[4] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, C. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari, M. Mirza, S. Jean, P. Carrier, Y. Dauphin, N. Boulanger-Lewandowski, J. Aggarwal, J. Zumer, P. Lamblin, J. P. Raymond, G. Desjardins, R. Pascanu, D. Warde-Farley, A. Torabi, A. Sharma, E. Bengio, M. Côté, K. R. Konda, and Z. Wu, "Combining modality specific deep neural networks for emotion recognition in video," in *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, 2013, pp. 543-550.

[5] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D. H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, "Challenges in representation learning: A report on the machine learning contest of the International Conference on Learning Representations 2013," *arXiv preprint arXiv:1307.0414*, 2013.

[6] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 2015, pp. 435-442.

[7] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2852-2861.

[8] C. Pramerdorfer and M. Kampel, "Facial expression recognition using convolutional neural networks: State of the art," *arXiv preprint arXiv:1612.02903*, 2016.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.