Facial Emotion Recognition

Tània Pazos and Yuyan Wang
Pompeu Fabra University
Deep Learning
June 9, 2025

GitHub Repository: https://github.com/yuyanwang03/DeepLearningProject

1. Introduction

Facial Emotion Recognition (FER) is a computer vision task that automatically classifies human emotions by analyzing facial expressions captured in static images or video. As a branch of affective computing, FER focuses on enabling machines to recognize and interpret human affective states, facilitating more natural and emotionally responsive interactions between humans and machines. Originally rooted in psychological and human-computer interaction research, FER has gained traction in recent years thanks to advances in deep learning, pattern recognition, and the widespread availability of camera-equipped devices.

The relevance of FER continues to grow across a wide range of applications. In healthcare, FER systems are used for detecting depression, monitoring therapy response, and identifying early signs of neurodegenerative disorders. In education, they support adaptive learning environments by tracking student attention and emotional engagement. Other use cases include personalized advertising, customer behavior analysis, driver fatigue detection, and fraud prevention. This broad applicability has led to substantial investment by both industry and publicly funded research initiatives such as Horizon 2020.

Despite this momentum, many state-of-the-art FER models rely on large and computationally intensive architectures, which are impractical in our project setting due to limited GPU access and the short two-week development timeline. The FER2013 dataset, comprising 48×48 grayscale images labeled across seven basic emotions, provides a challenging but widely used benchmark for evaluating FER systems under constrained conditions.

The goal of this project was to develop compact convolutional neural network (CNN) architectures for emotion recognition that maintain competitive performance while operating under strict parameter and computational constraints. To this end, we set the following measurable objectives:

1. Build emotion classification models with fewer than 500,000, ensuring feasibility on low-resource hardware.
2. Compare training strategies using different optimizers (Adam vs. SGD), dropout rates (0%, 15%, 20%), learning rates (1e-3, 1e-2), and weight decay, across fixed 15-epoch training runs.
3. Benchmark all models against a VGG16 baseline pretrained on ImageNet and fine-tuned on FER2013, with over 14 million parameters.
4. Evaluate models on overall accuracy, macro-averaged F1-score, and per-class precision and recall, to fairly assess performance across all emotion categories.

As a reference point, we trained and corrected a VGG16-based baseline from Kaggle, pretrained on ImageNet and fine-tuned on FER2013, which contains over 14 million parameters. To improve upon this, we designed and trained two lightweight CNNs based on ResNet and MobileNetV2, each adapted to the FER2013 dataset and our computational constraints. By exploring the trade-off between model size and recognition performance, the project contributes to building FER systems that are both efficient and practical for deployment in resource-limited academic and real-world settings.

2. State-of-the-Art

Facial Emotion Recognition (FER) has been widely explored in both academic and industry settings, with deep learning driving much of the recent progress. Early methods combined handcrafted features with shallow learning approaches, while newer models rely heavily on deep architectures that leverage convolutional, residual, and attention-based mechanisms. However, these models often come at the cost of high parameter counts, making them impractical for low-resource or real-time scenarios like ours.

One of the first models to surpass 75% accuracy on FER2013 was proposed by Georgescu et al. (2018), which fused Bag-of-Visual-Words (BoVW) handcrafted features with deep CNN embeddings. Their hybrid system achieved 75.42% accuracy on FER2013 using a local learning framework that retrains SVM classifiers per test image. Although effective, this method depends on both heavy inference-time computation and explicit feature engineering.

In contrast, recent deep learning approaches focus on end-to-end architectures. ResMaskingNet (Pham et al., 2021) combines a Deep Residual Network with a segmentation-inspired masking block to guide attention to

discriminative facial regions. This network, which integrates a U-Net-like refinement stage, achieved 74.14% accuracy on FER2013 but required approximately 143 million parameters, posing scalability challenges.

ResEmoteNet (Roy et al., 2024) introduced a deep architecture combining residual blocks with Squeeze-and-Excitation (SE) mechanisms. These SE blocks dynamically recalibrate channel-wise features to emphasize important facial regions while suppressing irrelevant ones. Evaluated across multiple datasets, including FER2013, AffectNet, RAF-DB, and ExpW, ResEmoteNet achieved strong results, with 79.79% accuracy on FER2013. While the model shows clear improvements in performance, no information is provided about its parameter count or suitability for low-resource environments, which limits its direct applicability to our constrained setting.

EfficientFER (Konuk, 2025) used transfer learning from EfficientNetV2 along with attention mechanisms to capture facial detail while maintaining robustness to noise. Despite using data augmentation, the model still required 23.8M parameters to reach 82.47% accuracy on FER2013, again highlighting the cost of high performance in terms of size.

These approaches, though effective, are incompatible with the constraints of our project, which targets architectures under 500,000 parameters. Instead, we investigated compact alternatives based on MobileNetV2 and ResNet, adapting their structure to operate effectively on 48×48 grayscale images. Our goal was not to outperform the latest benchmarks, but rather to identify designs that strike a balance between size, generalization, and class-level performance under strict computational and time limitations.

3. Methodology

3.1. Data Analysis & Preprocessing

The dataset used in this project was FER2013, a widely adopted benchmark for facial emotion recognition tasks. It contains 35,887 grayscale images of size 48×48 pixels, each labeled with one of seven basic emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The dataset is split into 28,709 training samples and 3,589 test samples. Sample images for each emotion class are shown in Figure 1.

Figure 1: Sample images from the FER2013 dataset across the 7 emotion classes.

3.1.1. Class Distribution and Imbalance

A key challenge of the FER2013 dataset is its significant class imbalance. As shown in Figure 2 (left), the Happy class is heavily overrepresented (7,215 samples), while Disgust is critically underrepresented (436 samples). This imbalance can bias training, causing models to perform poorly on minority classes, an issue we observed in preliminary runs' recall and F1-score metrics.

3.1.2. Offline Data Augmentation for Class Balancing

To address this imbalance, we implemented a two-stage offline augmentation pipeline:

1. Data injection from AffectNet: We selectively extracted real facial emotion samples for the Angry, Disgust, and Surprise classes from AffectNet. Each image was preprocessed to match FER2013 specifications: converted to grayscale and resized to 48×48. This step allowed us to increase minority class representation using semantically relevant samples.
2. Synthetic transformations: After merging AffectNet images, we still had a significant imbalance in the training set. We used Keras's ImageDataGenerator to apply transformations such as rotation (±15°), width/height shift (10%), zoom (10%), and horizontal flips. These synthetic images were saved to disk to bring all training classes to at least 4,500 samples. Figure 2 (right) shows the resulting balanced distribution, where blue bars indicate a more uniform training set.

This offline augmentation specifically targets distributional imbalance and enhances class-level generalization.

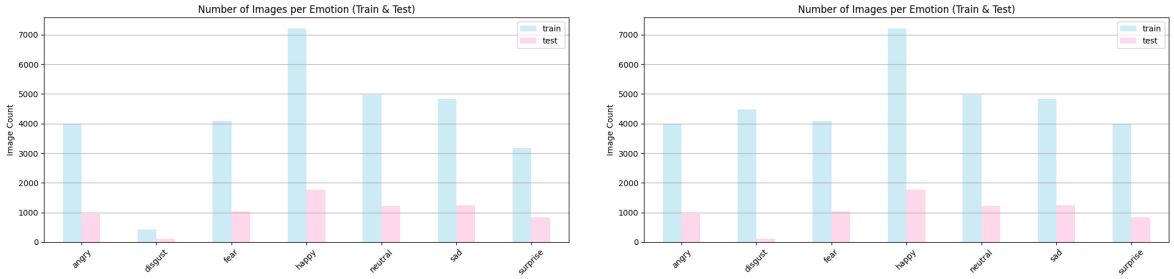

Figure 2: Class distribution in the FER2014 training and test sets before (left) and after (right) data augmentation.

3.1.3. Channel Normalization

Since both ResNet and MobileNetV2 require 3-channel RGB input, we converted all grayscale images to RGB by duplicating the single channel across three. This was done programmatically by iterating over all image files and overwriting them with their RGB-converted versions using PIL.

3.1.4. Online Data Augmentation

In addition to offline augmentation, we applied online data augmentation during training. This means new transformations were applied randomly in each epoch using PyTorch's transforms module. The goal here is different: to promote spatial robustness and reduce overfitting. With small 48×48 images, models can easily overfit to static spatial patterns. Applying transformations like random rotation, affine transformations, and horizontal flips forces the model to learn more generalizable features.

We normalized all input images to the range [-1, 1] using a mean of 0.5 and a standard deviation of 0.5 per channel. This normalization is particularly helpful for training stability with ReLU6 activations (used in MobileNetV2) and accelerates convergence by ensuring consistent input scale.

3.1.5. Initial Feature Analysis

To gain early insight into the dataset's intrinsic structure, we projected the FER2013 test set using t-SNE, a nonlinear dimensionality reduction technique. We first reduced each image to 50 dimensions using PCA and then applied t-SNE to map them into two dimensions. As shown in Figure 3, the raw pixel intensities do not yield clear clusters by emotion class. However, this plot serves as a useful baseline to compare against post-training embeddings, where meaningful separation may emerge.

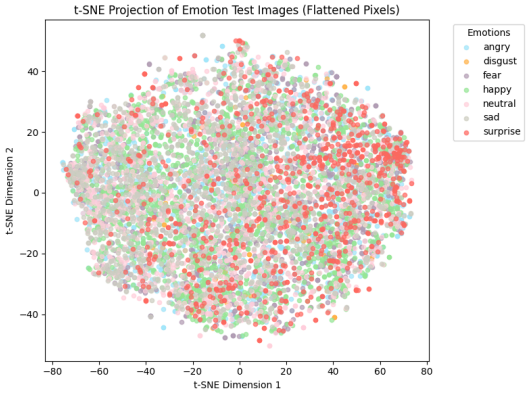

Figure 3: t-SNE projection of FER2013 test images based on raw pixel intensities.

3.2. Models and Optimization

In this section, we describe the architectural choices, model adaptations, and optimization strategies used in our project. We emphasize compact architectures suitable for constrained computational environments, while maintaining competitive recognition performance.

3.2.1. Baseline Model: VGG16

As a baseline, we employed a VGG16 model pretrained on ImageNet and subsequently fine-tuned on FER2013. The model architecture was based on a public Kaggle notebook by Y. H. Shakir. We intentionally did not modify the core architecture, which consists of 13 convolutional layers followed by 3 fully connected layers, with

dropout and batch normalization applied after the fully connected layers. The total model size was approximately 14.7 million parameters.

While the original implementation used binary cross-entropy and binary accuracy, we corrected this setup to reflect the true multiclass nature of the FER2013 task. Loss was changed to categorical cross-entropy; metrics were expanded to include macro F1-score, per-class precision and recall, and overall accuracy.

We preserved the training setup of the original model to allow for fair comparison: all models were trained for 15 epochs, without early stopping.

3.2.2. Lightweight ResNet Variant

Residual Networks (ResNets), introduced by He et al. (2015), are one of the most influential architectures in deep learning for image recognition. Their key innovation is the use of residual connections, which allow the network to learn identity mappings and enable gradients to flow more effectively through deep networks. This helps mitigate the vanishing gradient problem that typically hinders training very deep models. In a standard ResNet block, two convolutional layers (each followed by batch normalization and ReLU) process the input, which is then added element-wise to the original input via a skip connection. This residual addition facilitates the learning of residual functions rather than full transformations, improving both convergence and performance.

Given our project constraints (specifically, the target of keeping the model under 500,000 parameters), we designed a custom lightweight ResNet variant that balances model compactness with the ability to generalize well across classes.

We introduced two key modifications after the residual addition:

- A MaxPooling layer, which reduces spatial resolution and consequently lowers the number of computations in subsequent layers. While this does not reduce the number of parameters, it substantially reduces both memory usage and inference time.
- An optional Dropout layer, which acts as a regularizer to prevent overfitting and promote better generalization on unseen data.

Our final ResNet-based architecture consists of two modified residual blocks, each followed by a MaxPooling layer and a Dropout layer. The two modified residual blocks are followed by a fully connected (FC) layer that maps the resulting feature representation to the seven emotion classes. The entire model comprises approximately 418k parameters, making it substantially lighter than the VGG16 baseline.

3.2.3. Lightweight MobileNetV2 Variant

MobileNetV2, introduced by Sandler et al. (2018), was originally developed for efficient image classification on mobile and embedded devices, where computational and memory constraints are critical. The architecture is built around two main innovations: depthwise separable convolutions, which significantly reduce computational cost compared to standard convolutions, and inverted residual blocks with linear bottlenecks, which preserve representational power while reducing model complexity.

The original MobileNetV2 architecture was designed for high-resolution 224×224 RGB images and contains approximately 3.4 million parameters. To adapt MobileNetV2 to low-resolution 48×48 grayscale images and to align with our project's goal of building highly compact models, we made several architectural modifications.

We then carefully adjusted the downsampling strategy: in the original architecture, stride=2 layers appear early in the network, which would have overly reduced spatial resolution in our case. We therefore removed these early stride=2 layers to preserve sufficient spatial information necessary for accurate facial emotion recognition on small input images.

In addition, we reduced the number of bottleneck block repetitions and omitted the highest capacity blocks (those with 160 and 320 channels), as these were unnecessary at low resolution and contributed significantly to the model size. To further reduce the parameter count, we reduced the dimensionality of the final projection layer from 1280 to 128 channels. And finally, we inserted a Dropout layer with a rate after the final convolutional block to improve generalization and reduce overfitting.

The resulting MobileNetV2 variant contains approximately 130k parameters.

3.3. Experimental Procedure and Training Pipeline

For each model architecture (VGG16 baseline, ResNet variant, MobileNetV2 variant), we conducted a systematic hyperparameter search across optimizer choice, learning rate, Dropout rate, and, in the case of

MobileNetV2, weight decay. Each hyperparameter configuration was trained independently, always starting from the same augmented and preprocessed training set. Models were trained for a fixed number of 15 epochs in all initial experiments to ensure comparability across runs and architectures.
Training was performed using categorical cross-entropy, the standard loss function for multi-class classification. Validation performance was evaluated at each epoch using macro-averaged F1-score (our primary metric), overall accuracy, and per-class precision and recall. We deliberately prioritized F1-score (and reported per-class precision and recall) over simple accuracy because FER2013 is a highly imbalanced dataset. In such cases, accuracy can provide a misleading picture of model performance, as it is dominated by the majority classes. In contrast, the F1-score provides a more balanced and informative assessment of model behavior across all classes, reflecting both how many relevant instances are captured (recall) and how many predicted instances are correct (precision). We also monitored training and validation loss curves to detect signs of overfitting or underfitting.
To ensure stable optimization and fast convergence, all models used batch normalization. A batch size of 64 was used consistently across runs, balancing memory usage with effective gradient estimation. For reproducibility, we fixed the training pipeline, including data loading, augmentation, and evaluation steps.
Finally, to gain qualitative insights into the learned feature representations, we extracted the penultimate layer embeddings from each trained model and projected them using t-SNE. This provided an additional perspective on the extent to which different architectures succeeded in learning separable feature spaces for the seven emotion classes.
Based on the results of this initial 15-epoch hyperparameter sweep, we selected the best-performing configuration for each architecture based on a combination of validation accuracy, macro F1-score, per-class stability, and smooth loss trends. These configurations were retrained for 40 epochs using either extended training or early stopping with learning rate scheduling, depending on model behavior.

4. Experiments

This section summarizes the specific experiments conducted for each model, focusing on the impact of regularization and optimization choices under parameter and time constraints. Each experiment was designed to test a targeted hypothesis and guide the final model selection.

4.1. ResNet

Using the ResNet-based architecture described in Section 3.2.2, we designed five experimental configurations to systematically study the influence of three hyperparameters: dropout, learning rate, and optimizer choice.
1. Dropout: We tested dropout rates of 0%, 15%, and 20%, applied after each residual block and before the final fully connected layers. FER2013's low resolution makes the model prone to memorizing spatial patterns, so dropout was essential for regularization.
2. Learning rate: We compared learning rates of 1e-3 and 1e-2 to evaluate convergence stability. The lower rate promotes smoother optimization, while the higher rate may accelerate early learning at the cost of potential oscillation.
3. Optimizer Choice: We tested both Adam, for its fast convergence via adaptive learning, and SGD with momentum = 0.9, for its capacity to generalize better in some cases by following more consistent update directions.
After analyzing the results, the configuration with Adam, learning rate = 1e-3, and no dropout was selected for the 40-epoch final training. It consistently achieved the best validation accuracy and macro F1-score, along with stable per-class performance and smooth validation loss curves. Full metrics and plots supporting this selection are provided in Section 5.2.

4.2. MobileNetV2

For MobileNetV2, we used the compact, depthwise-separable CNN described in Section 3.2.3. The initial unregularized runs showed clear signs of overfitting, prompting the design of five configurations exploring the impact of dropout, weight decay, and optimizer choice.

1. Dropout: We evaluated rates of 0% and 15%, applied after the final convolutional block. Dropout helps prevent overfitting near the classifier, particularly in lightweight networks.
2. Weight Decay: We included L2 regularization with a value of 1e-4, both with and without dropout. While dropout acts on activations, weight decay penalizes large weights and improves generalization by implicitly controlling model complexity.
3. Optimizer Choice: We compared Adam and SGD with momentum = 0.9, using the same rationale as in the ResNet experiments.

Each configuration was trained for 15 epochs using the categorical cross-entropy loss and evaluated on accuracy, macro F1-score, and per-class metrics. Batch normalization was consistently used in all layers to stabilize training.

After completing the 15-epoch sweep, we selected the configuration that used Adam with dropout = 0.15 and weight decay = 1e-4 as our final model. Although the variant without weight decay slightly outperformed it in macro F1-score, the added regularization provided more stability in validation accuracy and less volatility in per-class metrics. Given the small input size and class imbalance in FER2013, we prioritized this stability when choosing the best configuration.

For the final model training, we extended the run to 40 epochs and introduced early stopping (with a patience of 6 epochs) to avoid overfitting once validation loss plateaued. Additionally, we employed the ReduceLROnPlateau learning rate scheduler, which dynamically halves the learning rate when no improvement in validation loss is observed for 3 consecutive epochs. This helped fine-tune the model in later stages of training and improve generalization without requiring manual learning rate tuning.

5. Results

The complete set of experimental results, including all training configurations, can be found in the accompanying results folder. For the sake of brevity, many plots and detailed outputs are not included in this document, as they would require several additional pages. However, for every model and training configuration explored, we systematically generated and analyzed plots of loss, accuracy, macro F1-score, and per-class precision and recall across training epochs, providing a comprehensive view of model behavior and performance.

5.1. Baseline Model: VGG16

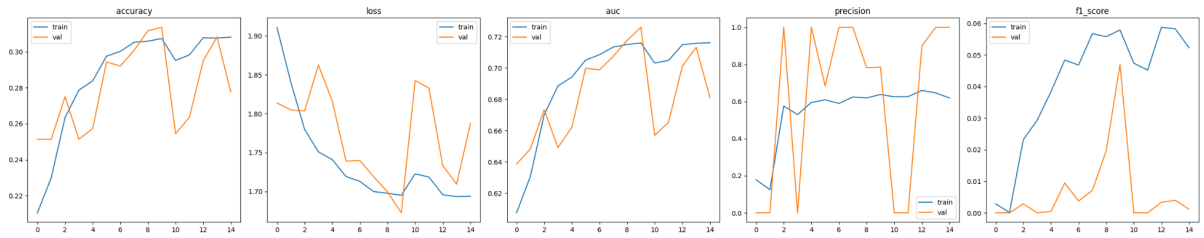

Figure 4: Training and validation curves for the VGG16 baseline model over 15 epochs.

5.2. ResNet

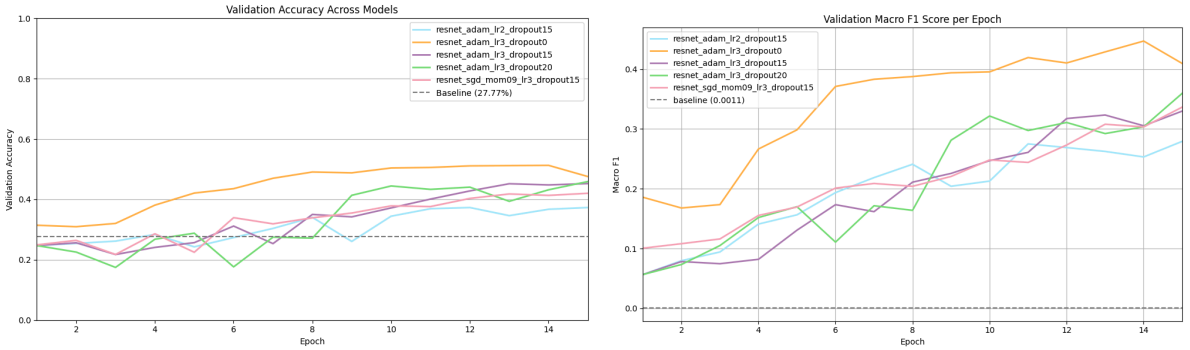

Figure 5,6: Accuracy (left) and Macro F1-Score (right) curves of the training and validation sets across different settings of ResNet.

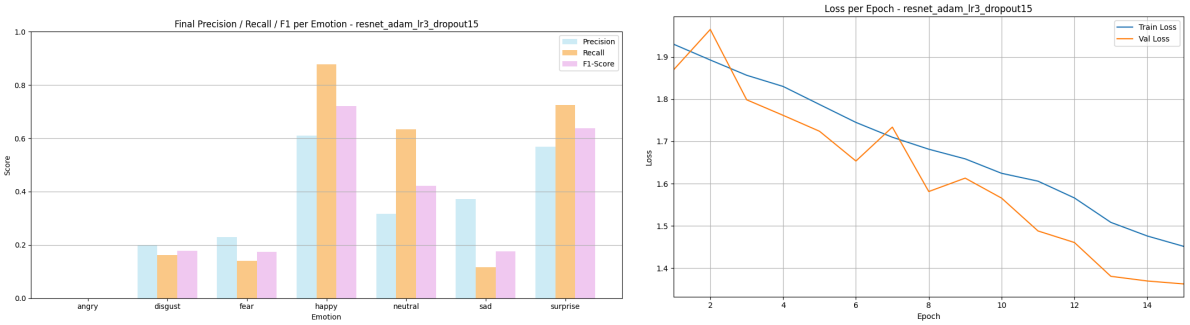

Figure 7, 8: Performance of one of the 5 ResNets (adam_lr3_dropout15) after 15 epochs. Incorporated for discussion in upcoming sections.

Model	Val Accuracy	Val F1	Happy Val F1	Disgust Val F1
Baseline (VGG16)	27,77%	0,11%	42,00%	0,00%
adam_lr3_dropout0_bn	47,60%	40,95%	76,30%	14,77%
adam_lr3_dropout15_bn	45,20%	32,99%	72,10%	17,82%
adam_lr2_dropout15_bn	37,30%	27,93%	61,60%	4,11%
adam_lr3_dropout20_bn	45,90%	35,96%	74,40%	10,80%
sgd_mom09_lr3_dropout15_bn	42,00%	33,65%	63,40%	3,54%

Table 1: Table summary of performance of the baseline model and ResNet models after 15 training epochs.

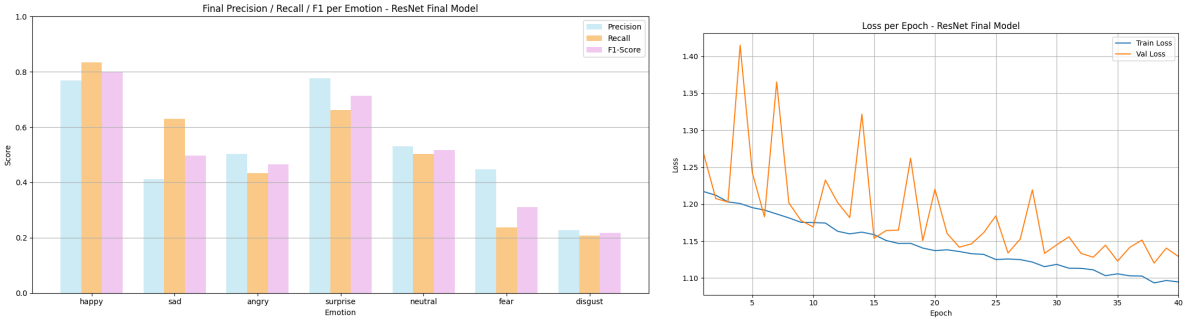

Figure 9, 10: Performance of the ResNet final model (adam_lr3_dropout0_bn) after 40 epochs.

Figure 11, 12: t-SNE projection of FER2013 test images after training of ResNet final model (adam_lr3_dropout0_bn).

5.3. MobileNetV2

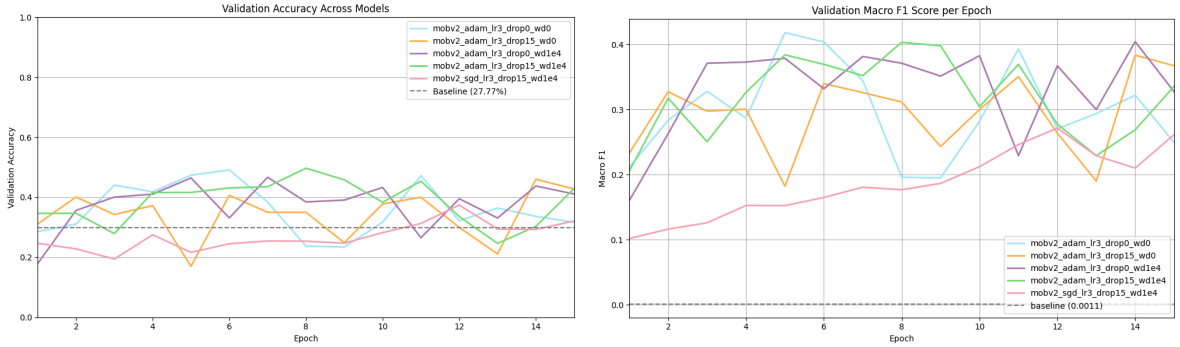

Figure 13, 14: Accuracy (left) and Macro F1-Score (right) curves of training and validation across different settings of MobileNetV2.

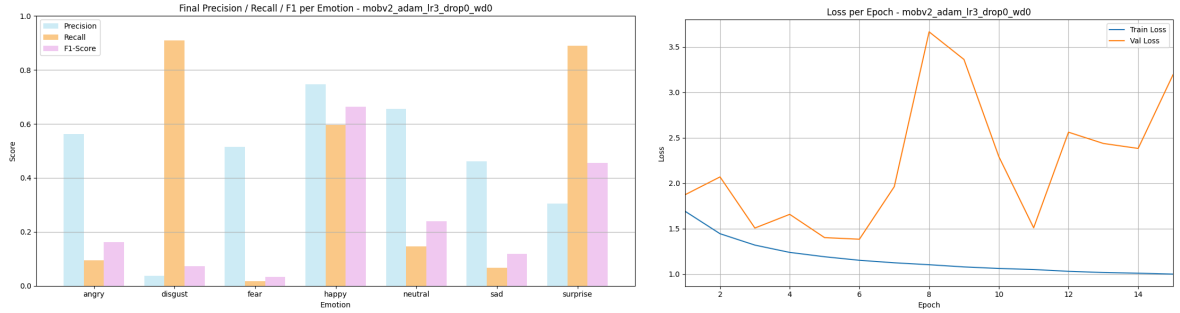

Figure 15, 16: Performance of one of the 5 MobileNetV2 models (lr3_drop0_wd0) after 15 epochs as Example.

Model	Val Accuracy	Val F1	Happy Val F1	Disgust Val F1
Baseline (VGG16)	27,77%	0,11%	42,00%	0,00%
adam_lr3_drop0_wd0_bn	31,70%	24,92%	66,35%	7,22%
adam_lr3_drop0_wd1e4_bn	41,08%	32,71%	17,34%	11,61%
adam_lr3_drop15_wd0_bn	42,76%	36,73%	75,81%	9,07%
adam_lr3_drop15_wd1e4_bn	42,96%	33,56%	74,32%	9,44%
sgd_lr3_drop15_wd1e4_bn	32,13%	26,13%	56,73%	2,66%

Table 2: Table summary of performance of the baseline model and MobileNetV2 models after 15 training epochs.

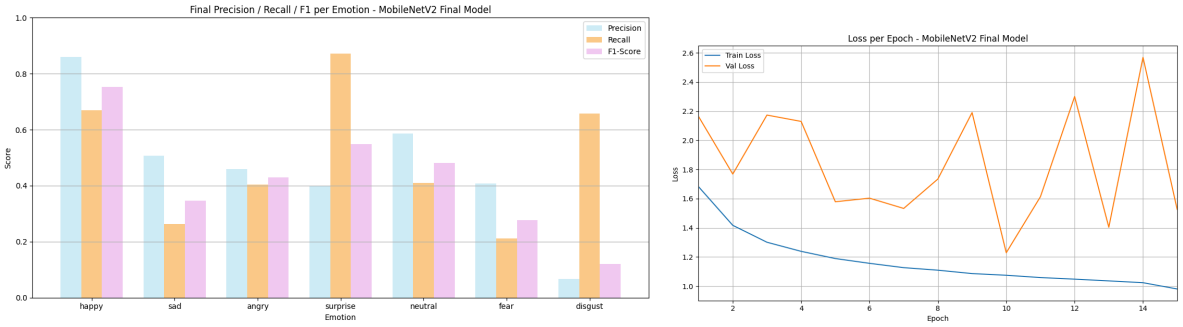

Figure 17, 18: Performance of the MobileNetV2 final model (adam_lr3_drop15_wd1e4_bn) used early stopping and LR scheduler,.

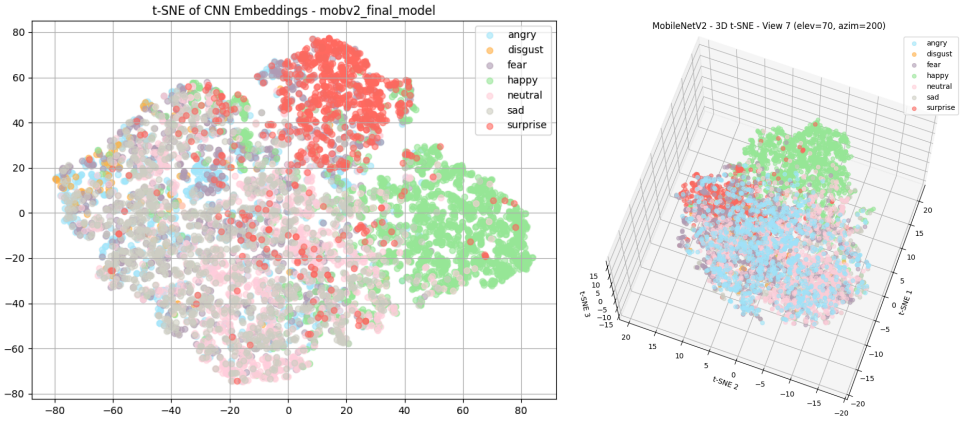

Figure 19, 20: t-SNE projection of FER2013 test images after training of ResNet final model (adam_lr3_dropout0_bn).

6. Discussion & Future Work

6.1. ResNet

The results of our ResNet experiments provide valuable insights into how architectural choices and optimization strategies affect performance in the low-resource FER setting. As shown in Figures 5 and 6, our lightweight ResNet variants consistently outperformed the VGG16 baseline across all configurations, both in terms of validation accuracy (most observably in the last 5 iterations) and, more importantly, macro F1-score. This confirms our initial hypothesis that a carefully designed small architecture can match or surpass the performance of a much larger model when combined with appropriate regularization and data augmentation.

Among the configurations tested, the adam_lr3_dropout0_bn variant achieved the best overall performance, reaching a macro F1-score of 40.95% after 15 epochs, a substantial improvement over the baseline's 0.11%. Interestingly, this configuration did not include dropout, and yet outperformed the dropout-based variants. One likely explanation is that in this particular setting (with small 48×48 pixel inputs and already strong spatial downsampling introduced by the MaxPooling layers), the model benefited from preserving as much feature information as possible. The MaxPooling layers already acted as a strong form of implicit regularization by enforcing spatial invariance and reducing overfitting to local patterns. Combined with batch normalization and online data augmentation, this likely provided enough regularization without the need for additional noise injection through dropout. In contrast, applying dropout on top of this may have degraded useful representations, especially when the input is already low resolution and the model capacity is constrained (~418k parameters). This interpretation is supported by the relatively smooth and stable validation curves observed for adam_lr3_dropout0_bn, while other dropout-based configurations showed more variability and slightly lower final F1-scores. Overall, this suggests that in low-resolution FER tasks, stacking too many regularization techniques can be counterproductive, and careful balancing is required to avoid underutilizing the limited model capacity.

The optimizer choice didn't seem to affect the performance much. That said, Adam consistently converged faster (see plots in the results folder); however, the SGD-based model was not a fundamentally poor choice. It

achieved reasonable overall performance, but it was notably less stable during training, as seen in the noisier validation curves. Moreover, it exhibited the worst recall and F1-score on the Disgust class, suggesting that it struggled more than the Adam-based variants on the most challenging minority classes. Learning rate lr2 (0.01) showed the worst overall F1-score and validation accuracy among all models, suggesting that it may not be a suitable choice in this particular training regime.

Analyzing per-class performance (Table 1), we see that Happy class F1-scores were consistently high (above 72% for most configurations), while Disgust remained challenging (best F1-score of 17.82%). This highlights the persistent difficulty of learning minority classes despite augmentation and suggests that future work should explore more targeted sampling or loss reweighting strategies.

An interesting behavior we observed in one of the models (adam_lr3_dropout15) is illustrated in Figures 7 and 8. In this run, the angry emotion exhibited an F1-score and precision/recall of zero, meaning that no angry test samples were correctly classified by the model. This could be attributed to several factors. First, while data augmentation and class balancing were applied, the angry class in FER2013 remains visually close to other negative emotions, such as sad and fear, making it particularly difficult to distinguish, especially with a compact model capacity. Second, individual training runs can be sensitive to initialization, and it is possible that the model in this configuration simply failed to converge to a useful representation for angry during this run.

Additionally, the loss curves for this model show another intriguing phenomenon: the validation loss remains consistently below the training loss throughout training. While this could at first seem counterintuitive, it likely reflects the heavy use of online data augmentation during training, which artificially increases the difficulty of the training set compared to the static validation set. Since the model is constantly exposed to perturbed, more challenging training images, its training loss remains higher, whereas on the unperturbed validation set, it achieves lower loss. This is a common and generally positive sign of good generalization, although in this particular case, the per-class metrics reveal that generalization is still uneven across the different emotion categories.

6.1.1. ResNet Final Model (adam_lr3_dropout0_bn)

The final selected ResNet model was further trained for 40 epochs to maximize its performance. The training and validation loss curves remained well-behaved throughout training, showing no major signs of overfitting. The model ultimately achieved a validation macro F1-score of approximately 50.32%, a notable improvement over the VGG16 baseline. The per-class precision/recall/F1 plot clearly reflects this performance: emotions like happy and surprise are classified reliably, while minority or subtle emotions such as fear and disgust remain more difficult to distinguish, consistent with trends observed in prior sections.

To qualitatively assess the model's ability to learn meaningful representations, we visualized the t-SNE projection of the penultimate layer embeddings for the FER2013 test set. The results reveal well-formed clusters for dominant emotions such as happy and surprise, with these classes exhibiting strong spatial separation in the embedding space. In contrast, emotions like neutral, sad, and fear remain entangled, which correlates with their lower per-class F1-scores and suggests that the visual boundaries between these emotions in FER2013 are inherently ambiguous. These visualizations reinforce that the ResNet model not only improved quantitative metrics but also learned an interpretable internal feature space.

Overall, these results validate the effectiveness of our ResNet modifications. The addition of MaxPooling after the residual addition improved computational efficiency without sacrificing accuracy. Furthermore, the low parameter count (~418k) demonstrates a favorable trade-off between model size and performance. Importantly, since no clear signs of overfitting were observed even after 40 epochs, a natural next step would be to experiment with increasing the model size; either by adding more residual blocks or expanding feature dimensions to explore whether greater capacity could further enhance performance.

However, several limitations remain in the current approach. Despite extensive offline and online augmentation, class imbalance continues to impact the model's ability to reliably classify minority emotions such as disgust and fear. Addressing this limitation may require more sophisticated augmentation techniques such as class-conditional GAN-based synthesis or targeted oversampling strategies that better capture the variability of minority classes.

Moreover, while the current training setup used a fixed learning rate, employing a learning rate scheduler improves convergence and enables the model to better exploit the limited number of training epochs. Our current architecture also uses standard batch normalization, but modern alternatives like Group Normalization or

Layer Normalization may provide better regularization, particularly when batch sizes are small or highly variable.
Another promising direction for future work is the incorporation of attention mechanisms. Spatial attention or channel-wise attention could help the model selectively emphasize facial regions that are most discriminative for subtle or ambiguous emotions—an area where even human annotators often struggle. Finally, since the current model architecture remains relatively shallow, increasing capacity through additional residual blocks or exploring compact Transformer-CNN hybrids could offer further improvements, particularly given that our results suggest the model has not yet reached an overfitting regime.

6.2. MobileNetV2

An interesting contrast emerged when analyzing the behavior of our MobileNetV2 variants. Across all configurations tested, MobileNetV2 consistently showed signs of overfitting. Despite employing several regularization techniques, the validation performance frequently deteriorated or fluctuated after initial gains. This is clearly visible in both the validation accuracy and macro F1-score plots (and also from the loss plots in the results folder), where none of the MobileNetV2 models achieved the stability observed in our ResNet variant. In particular, the learning curves displayed frequent oscillations and lacked a clear convergence trend, suggesting that the model was learning patterns in the training set that did not generalize well to unseen data.
Several factors may explain this behavior. First, MobileNetV2 was originally designed for large-resolution, high-content RGB images (224x224), whereas FER2013 provides low-resolution grayscale images (converted to 3 channels). The extensive downsampling operations and bottleneck layers in MobileNetV2 may lead to an excessive loss of spatial detail that is critical for recognizing subtle facial expressions in 48x48 images. Moreover, the aggressive compression inherent to MobileNetV2's architecture, combined with the small dataset size, likely amplifies its tendency to memorize training samples rather than learning robust representations.
Architecture choice also played a key role. Our modifications aimed to adapt MobileNetV2 for small input sizes by avoiding early stride-2 layers and reducing the number of bottleneck repeats. However, these changes may have disrupted the architecture's carefully balanced design, degrading its ability to generalize. Furthermore, the inverted residual blocks in MobileNetV2 introduce additional non-linearities and feature compression, which may be ill-suited to the low-variance patterns in FER2013 compared to the object-rich ImageNet dataset it was originally optimized for.
Optimizer choice and learning rate had more subtle effects. While models trained with Adam generally reached higher peak validation F1 scores, the variant trained with SGD and momentum exhibited the most stable validation curves over the course of training. In particular, the MobileNetV2 sgd_lr3_drop15_wd1e4_bn model showed less oscillation and smoother learning dynamics, despite achieving lower overall F1 than the best Adam configurations. This suggests that the implicit regularization effect of SGD may help counteract MobileNetV2's tendency to overfit, even if its final performance remained suboptimal. Weight decay provided marginal additional benefits across both optimizers but was insufficient on its own to fully regularize the network or prevent the overfitting trends observed.
A concrete example of this behavior can be observed in Figures 15, 16, which shows one of the MobileNetV2 variants (adam_lr3_drop15_wd1e4_bn) after 15 epochs. The class-wise performance is highly uneven: the recall for disgust reaches nearly 0.9, whereas the recall for fear remains close to zero. This discrepancy highlights both strengths and weaknesses of the model. In the case of disgust, the model has likely overfitted to a small number of highly distinctive training samples for that class, leading to very high recall but paired with low precision, indicating that it over-predicts disgust on ambiguous inputs. Conversely, fear suffers from both very low recall and low F1-score, suggesting that the model fails to capture reliable features for this class. One plausible reason is that fear is often visually confused with other negative emotions (e.g., surprise, sad), and MobileNetV2's aggressive downsampling may discard subtle local cues needed for differentiation. More generally, the large gaps between precision and recall across several classes reflect an unstable decision boundary, which is consistent with the oscillating validation loss and the overall tendency to memorize rather than generalize robust features.

6.2.1. MobileNetV2 Final Model (adam_lr3_drop15_wd1e4_bn)

This configuration achieved the best overall balance among MobileNetV2 variants, reaching a macro F1-score of ~42.26%. However, the training and validation loss curves clearly reveal persistent signs of overfitting:

despite applying dropout, weight decay, and a learning rate scheduler, the validation loss remains highly unstable and consistently higher than the training loss throughout the training window. This suggests that the MobileNetV2 architecture, although highly efficient, might not be fully well-suited to this small and noisy dataset, where the limited spatial resolution (48x48) combined with the aggressive bottleneck structure and inverted residuals of MobileNetV2 can potentially cause the model to overfit subtle noise patterns in the training set.

Per-class precision and recall analysis further supports this: while high precision and recall were achieved on dominant classes such as happy and surprise, minority classes such as fear and disgust remained poorly classified, showing large gaps between precision and recall, a typical sign of instability in learned representations. Interestingly, the t-SNE visualizations of the learned embeddings reveal that while happy and surprise classes are well-separated, the remaining emotions form largely overlapping and diffuse clusters, suggesting that MobileNetV2 struggled to extract robust features for subtle or ambiguous emotions. This is consistent with what was observed across all MobileNetV2 configurations: even though different optimizers and regularization schemes were explored, none fully mitigated the overfitting trend. The model often reached good training loss and high per-class precision for dominant classes, but this came at the expense of poor generalization on the test set.

These findings highlight the limitations of MobileNetV2 in this constrained setting. While highly efficient in terms of parameter count (~130k), its architecture may not be optimal for FER tasks on low-resolution, imbalanced datasets. For future work, experimenting with alternative lightweight architectures explicitly designed for small images (e.g., ShuffleNetV2, EfficientNet-lite) could prove more fruitful. Another avenue would be to incorporate attention mechanisms or facial landmark-based guidance to help the model focus on the most informative regions of the face. Nevertheless, in this project's context, ResNet demonstrated a superior parameter-efficiency tradeoff and more reliable performance for facial emotion recognition.

7. Conclusions

- Both custom models—ResNet (418k) and MobileNetV2 (130k)—successfully met the <500k parameter constraint and trained efficiently on standard hardware, fulfilling our core design objective for deployability in resource-constrained settings.
- Adam consistently outperformed SGD, confirming its advantage in fast convergence and higher F1-scores across both architectures. While dropout helped stabilize MobileNetV2, it proved unnecessary, and even counterproductive, for ResNet.
- The best ResNet configuration (adam_lr3_dropout0_bn) achieved a validation macro F1-score of 50.3%, significantly outperforming the VGG16 baseline, which only reached 0.11%. MobileNetV2 also surpassed the baseline, with the best configuration achieving 42.3% macro F1.
- Despite MobileNetV2's smaller parameter count, ResNet consistently achieved better F1-scores and more stable convergence. MobileNetV2 showed persistent overfitting, likely due to its architectural sensitivity to low-resolution data.
- ResNet learned more distinct and separable emotion clusters, particularly for dominant classes like happy and surprise. In contrast, MobileNetV2 embeddings were less structured, reflecting poor generalization for subtle or minority emotions.
- Both models consistently underperformed on minority classes despite offline balancing and online augmentation. These results suggest the need for future improvements, such as loss reweighting, GAN-based sample synthesis, or attention-based models better suited to subtle emotion distinctions.

8. References

- European Data Protection Supervisor. (2021, May). Facial emotion recognition (TechDispatch #1/2021). https://www.edps.europa.eu/system/files/2021-05/21-05-26_techdispatch-facial-emotion-recognition_ref_en.pdf

- Kim, B. J., Choi, H., Jang, H., Lee, D., & Kim, S. W. (2023, July). How to use dropout correctly on residual networks with batch normalization. In Uncertainty in Artificial Intelligence (pp. 1058–1067). PMLR.
- Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. IEEE Transactions on Affective Computing, 10(1), 18–31. https://doi.org/10.1109/TAFFC.2017.2740923
- Pham, L., Vu, T. H., & Tran, T. A. (2021, January). Facial expression recognition using residual masking network. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 4513–4519). IEEE. https://doi.org/10.1109/ICPR48806.2021.9412403
- Sambare, M. (n.d.). FER2013 facial emotion recognition dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/msambare/fer2013
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510–4520). https://doi.org/10.1109/CVPR.2018.00474
- Shakir, Y. H. (2021). Emotion recognition with VGG16. Kaggle. https://www.kaggle.com/code/yasserhessein/emotion-recognition-with-vgg16
- Papers with Code. (2025). Facial expression recognition on FER2013. Retrieved from https://paperswithcode.com/sota/facial-expression-recognition-on-fer2013