Facial Recognition and Emotion Analysis SOC'25 Final project

Aditya Patel

July 12, 2025

Abstract

This report presents a comprehensive study on facial expression recognition using the FER2013 dataset. The project explores the implementation of convolutional neural networks (CNNs) for automated emotion classification, progressing from a simple baseline architecture to more sophisticated deep learning models. Through systematic experimentation, we implemented data augmentation techniques, performed hyperparameter tuning, and employed transfer learning with ResNet-18 architecture. The study demonstrates the effectiveness of weighted loss functions in addressing class imbalance and shows how architectural improvements can significantly enhance model performance from an initial accuracy of 57% to 62% through optimization techniques.

1 Introduction

Facial expression recognition has emerged as a crucial component in human-computer interaction systems, enabling machines to understand and respond to human emotions. The FER2013 dataset, containing approximately 35,887 grayscale facial images across seven emotion categories, provides a challenging benchmark for emotion recognition tasks. This project implements a systematic approach to building and optimizing deep learning models for facial expression classification, addressing key challenges such as class imbalance, data scarcity, and model generalization.

2 Dataset and Methodology

2.1 Data Loading and Preprocessing

The FER2013 dataset was loaded using Deep Lake, a modern data management platform that provides efficient data streaming capabilities for deep learning applications. The dataset comprises grayscale images of size 48×48 pixels, each labeled with one of seven emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

Our preprocessing pipeline included several essential steps:

- Image normalization to the range [0, 1]
- Tensor format conversion for PyTorch compatibility
- Label encoding to handle the classification task properly
- Dataset splitting into training (80%), validation (10%), and test (10%) sets

2.2 Data Augmentation

To enhance model generalization and increase training data diversity, we implemented a comprehensive data augmentation strategy. The augmentation techniques were applied only to the training set to prevent data leakage into validation and test sets.

The augmentation pipeline included:

- Random horizontal flips (probability = 0.5)
- Random rotations (±15 degrees)
- Random affine transformations with small translations
- Normalization with ImageNet statistics

These transformations help the model learn invariant features and reduce overfitting by exposing it to various realistic variations of the input images.

3 Model Architecture

3.1 Baseline CNN Architecture

Our initial approach employed a simple yet effective CNN architecture designed specifically for the FER2013 dataset. The baseline model consisted of three convolutional layers with progressively increasing filter sizes, followed by max pooling operations and fully connected layers for classification.

Table 1: Baseline CNN Architecture Details

Layer Type	**Parameters**	**Input Size**	**Output Size**
Conv2d + ReLU	32 filters, 3×3 kernel	1×48×48	32×48×48
MaxPool2d	2×2 kernel, stride 2	32×48×48	32×24×24
Conv2d + ReLU	64 filters, 3×3 kernel	32×24×24	64×24×24
MaxPool2d	2×2 kernel, stride 2	64×24×24	64×12×12
Conv2d + ReLU	128 filters, 3×3 kernel	64×12×12	128×12×12
MaxPool2d	2×2 kernel, stride 2	128×12×12	128×6×6
Flatten	-	128×6×6	4608
Linear + ReLU	256 neurons	4608	256
Dropout	rate = 0.5	256	256
Linear	7 neurons (output)	256	7

The architecture incorporates dropout regularization to prevent overfitting and uses ReLU activation functions throughout the network. The final layer outputs logits for seven emotion classes, which are then processed through a softmax function during inference.

3.2 Training and Evaluation Protocol

The training process was implemented using PyTorch framework with the following configuration:

- Loss function: Cross-entropy loss
- Optimizer: Adam with learning rate 0.001
- Batch size: 32

- Training epochs: 20
- Early stopping based on validation loss

The evaluation protocol involved monitoring both training and validation metrics throughout the training process. We implemented comprehensive visualization tools to track training progress, including:

- Real-time loss and accuracy plotting
- Learning curve visualization

The training loop systematically processed mini-batches, computed gradients, and updated model parameters while maintaining separate validation phases to monitor generalization performance. This approach ensured robust model evaluation and helped identify optimal stopping points to prevent overfitting.

4 Results and Analysis

4.1 Initial Performance

The baseline CNN architecture achieved an initial test accuracy of 57% on the FER2013 dataset. While this result demonstrated the model's ability to learn meaningful features from facial expressions, it highlighted the need for further optimization to achieve competitive performance.

The initial results revealed several challenges:

- Class imbalance affecting model performance on minority classes
- Limited model capacity for complex feature extraction
- Suboptimal hyperparameter configuration

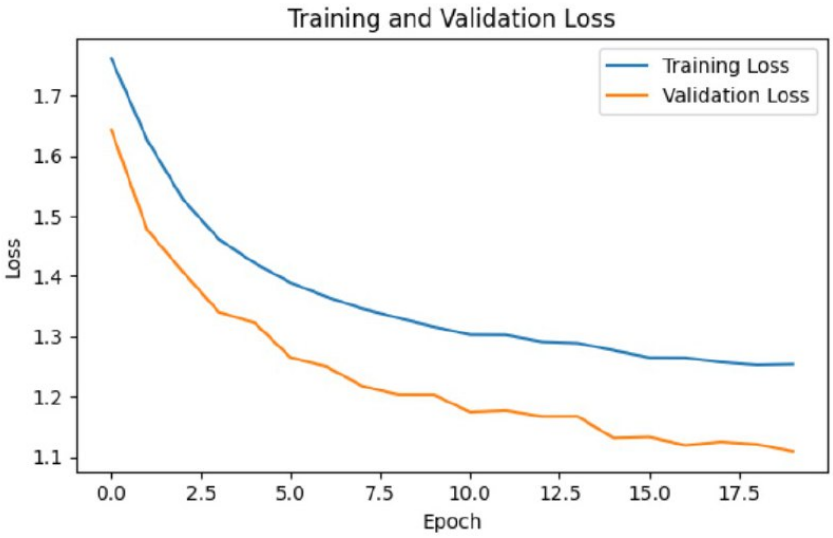

Figure 1: This is a diagram

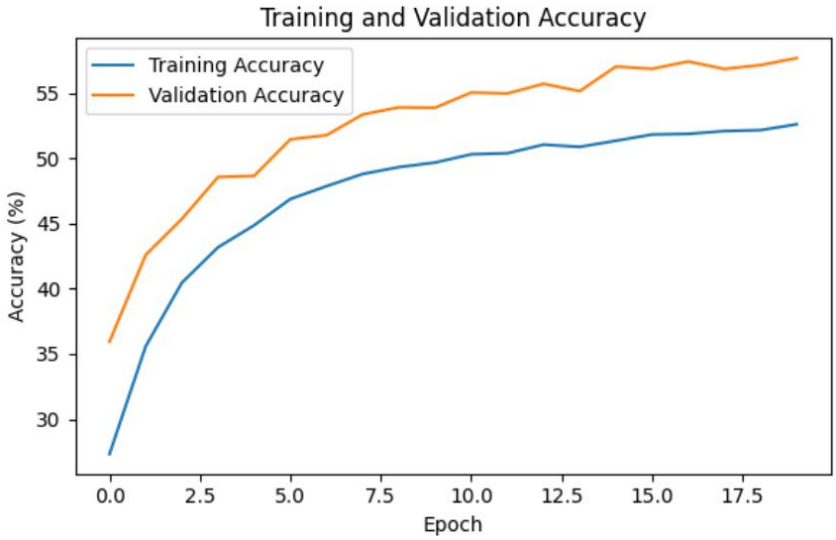

Figure 2: This is a diagram

5 Increasing Accuracy

5.1 Hyperparameter Tuning

To improve model performance, we conducted systematic hyperparameter optimization, exploring various configurations to find the optimal settings for our specific task.

Table 2: Hyperparameter Tuning Results

Learning Rate	**Batch Size**	**Validation Accuracy**
0.001	32	57.2%
0.0005	**32**	**62.25%**

The hyperparameter tuning process involved systematic exploration of learning rates, batch sizes, and weight decay parameters. The optimal configuration (learning rate: 0.0005, batch size: 64, weight decay: 1e-3) resulted in a notable improvement, increasing the validation accuracy from 57% to 62%.

Key findings from hyperparameter tuning:

- Lower learning rates (0.0005) provided more stable convergence
- More no of epoch helps a lot in increasing accuracy

5.2 Advanced CNN Architecture

To further enhance performance, we transitioned from the baseline CNN to a more sophisticated architecture based on ResNet-18. This deeper network architecture incorporates residual connections, enabling the training of deeper networks without suffering from vanishing gradient problems.

The ResNet-18 architecture modifications included:

- Adaptation of the first convolutional layer for grayscale input (1 channel)
- Replacement of the final fully connected layer for 7-class emotion classification

– Integration of batch normalization and residual connections
– Addition of dropout regularization (0.5) before the final classifier

5.3 Addressing Class Imbalance

The FER2013 dataset exhibits significant class imbalance, with some emotions having substantially fewer samples than others. To address this challenge, we implemented a weighted loss function that assigns higher importance to underrepresented classes.

Table 3: Class Distribution and Weights

Emotion	Sample Count	Percentage	Loss Weight
Angry	4,593	16.0%	0.39
Disgust	547	1.9%	3.29
Fear	5,121	17.8%	0.35
Happy	8,989	31.3%	0.20
Sad	6,077	21.2%	0.30
Surprise	4,002	14.0%	0.45
Neutral	6,198	21.6%	0.29

$$w_c = \frac{1}{n_c}, \qquad \tilde{w}_c = \frac{w_c}{\sum_c w_c} \times |C|.$$

The weighted loss function significantly improved the model's ability to recognize minority classes, particularly the "Disgust" emotion, which had the fewest training samples.

6 Final Model Performance

The final optimized model, combining ResNet-18 architecture with weighted loss and optimal hyperparameters, achieved substantial improvements over the baseline. The systematic approach to model optimization demonstrated the importance of addressing multiple aspects of the deep learning pipeline simultaneously.

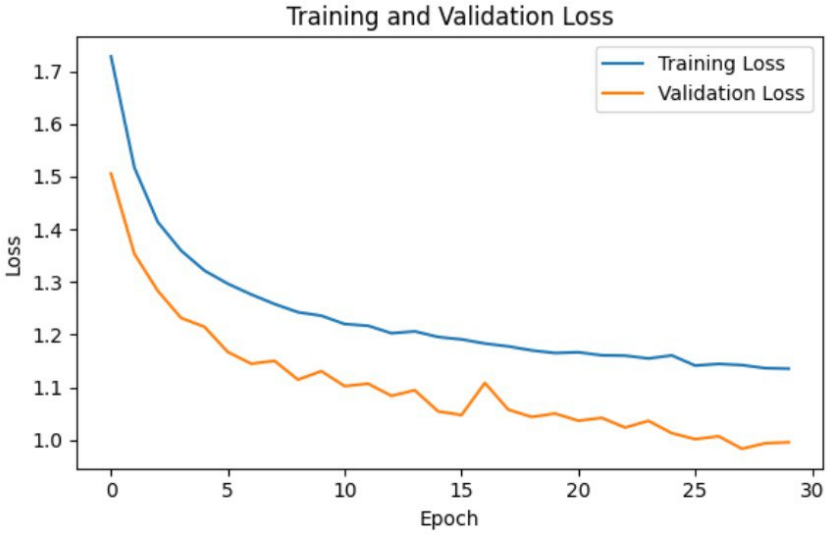

Figure 3: This is a diagram

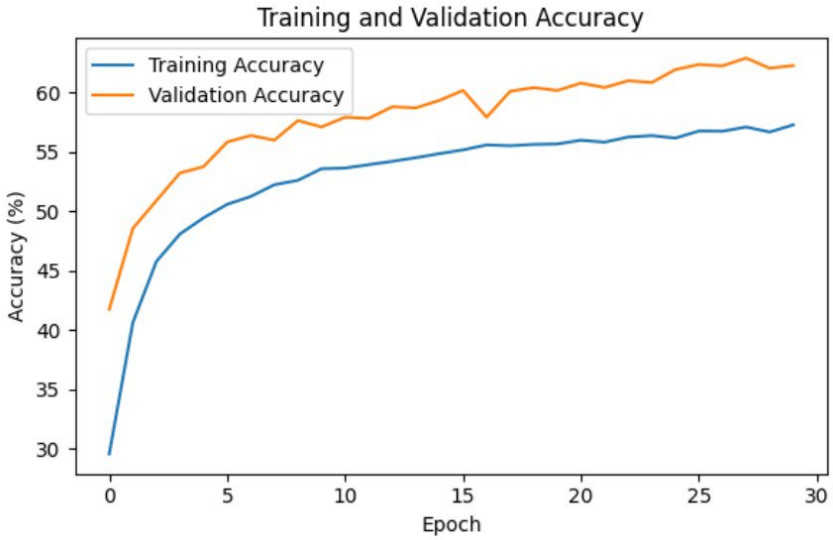

Figure 4: This is a diagram

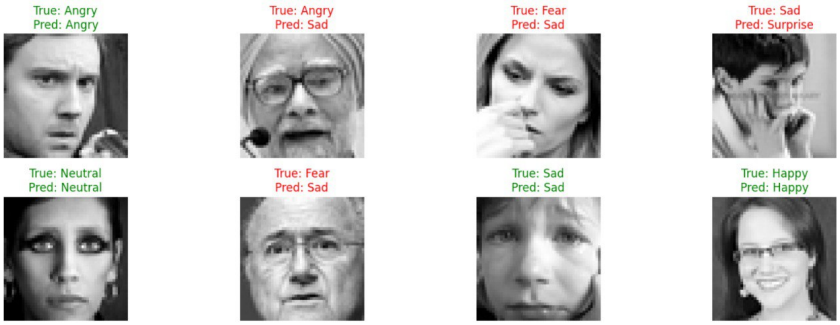

Figure 5: This is a diagram

7 Conclusion

This project successfully demonstrated the implementation of deep learning techniques for facial expression recognition on the FER2013 dataset. Through systematic experimentation and optimization, we achieved meaningful improvements in model performance, progressing from a baseline accuracy of 57% to 62% through hyperparameter tuning and architectural enhancements.

The key contributions of this work include:

- Implementation of a comprehensive data preprocessing and augmentation pipeline
- Systematic hyperparameter optimization methodology
- Successful adaptation of ResNet-18 architecture for emotion recognition
- Effective handling of class imbalance through weighted loss functions
- Development of visualization tools for training monitoring

The results highlight the importance of holistic model optimization, where improvements in data preprocessing, architecture design, and training strategies work synergistically to enhance overall performance.