Final Writeup

Bengisu Bulur, Sophy Figaroa, Aiko Kato

December 2, 2025

1 Introduction

Emotion recognition is widely used in areas such as human-computer interaction, mental-health support tools, robotics, and accessibility technologies. These systems help interpret human emotions from facial expressions, enabling applications ranging from improving communication to enhancing interactive system feedback. In this project, we built a facial-emotion-recognition model that classifies images into seven emotion categories (angry, disgust, fear, happy, neutral, sad, surprise) using large-scale datasets. By experimenting with a Convolutional Neural Network (CNN) model using different numbers of layers, adjusting training strategies such as learning rates and epoch counts, and evaluating performance using accuracy, and confusion matrices, we aim to understand how these choices affect the model's ability to recognize emotions reliably. This also allows us to explore how emotion-recognition models can be applied in real-world settings while understanding their practical limitations.

2 Experimental Setup

2.1 Dataset

We used the Facial Emotion Recognition Dataset by Fahadullah for training, and the Facial Emotion Recognition Dataset by Jonathan Oheix for testing. The training dataset contains 49,779 images across seven emotion categories (angry, disgust, fear, happy, neutral, sad, surprise), and the testing dataset contains 7,066 images with the same categories.

2.2 Environment

We used Python as our language and ran our training and testing on the Pomona College's OnDemand HPC environment due to the large dataset size. For libraries and tools, we used PyTorch, TorchVision, and OpenCV for image preprocessing, dataset loading, CNN model construction, running the training pipeline, and doing a real-time prediction demonstration. Here is the link to our GitHub repository, which contains the final code, results, and our Jupyter notebook files:

https://github.com/sophycodes/CS158EmotionRecognitionModel

2.3 Experiment

To understand how different design choices affect emotion-recognition performance, we conducted a series of controlled experiments using our CNN-based models. All experiments were run through the terminal using our training script, which supports different model types, learning rates, and epoch counts. We trained and tested each configuration on the Kaggle datasets described earlier, using Cross-Entropy Loss as our objective function and the ADAM optimizer as our gradient-descent method.

(a) Model Architecture Comparison

We compared three CNN architectures implemented in our code: a 2-layer CNN (`Simple CNN`), a 3-layer CNN (`Medium CNN`), and a 4-layer CNN (`Large CNN`). This comparison allowed us to observe how adding more convolutional layers and regularization techniques changes the model's ability to learn more complex emotional features.

(b) Training Strategy Comparison

After identifying the best-performing model (the 4-layer CNN as the top model and the 2-layer CNN as the second-best), we continued testing different training strategies to examine their effects.

- Learning rate comparison: 0.001 vs. 0.01, tested on the best-performing model (`Large CNN`).
- Epoch count comparison: Since the 4-layer CNN consistently performed the best, we examined how extended training would affect the other two models. For the `Simple CNN`, we trained for 5 and 20 epochs, and for the `Medium CNN`, we trained for 20 and 35 epochs. This allowed us to evaluate whether additional training improved performance or led to overfitting.

(c) Evaluation

For each experimental configuration, we evaluated performance using accuracy (training and validation), loss (training and validation), and confusion matrices.

(d) Testing on a Separate Dataset

To measure generalization, we tested our trained models on the second dataset. This evaluation allowed us to observe how well our models performed on unseen images.

3 Results

For the 2-layer CNN, the best accuracy we obtained was 0.6140 at epoch 13, and its confusion matrix (Figure 1) shows a reasonably strong diagonal, indicating that the model correctly predicted most images across different emotion classes. The 3-layer CNN achieved its highest accuracy of 0.5997 at epoch 20, but its confusion matrix (Figure 2) shows a weaker diagonal with more off-diagonal errors, meaning it struggled more with distinguishing between emotions. Especially for certain emotions such as fear and sad, it classifies them incorrectly more often than it does correctly. The best-performing model overall was the 4-layer CNN, which reached an accuracy of 0.7200 at epoch 16. Figure 3 shows the confusion matrix, where the strong diagonal indicates the highest counts of correct predictions.

For the learning rate comparison, we tested two values: 0.001 and 0.01. While the overall differences were not large, the 4-layer CNN with a learning rate of 0.001 achieved a higher accuracy of 0.7200 at epoch 16 (Figure 4), whereas the same model with a learning rate of 0.01 reached a slightly lower accuracy of 0.7067 at epoch 12 (Figure 5).

For the epoch count comparison, we examined how increasing training duration affected performance. For the 2-layer CNN trained for 5 epochs, the model was still clearly improving, with both training and validation accuracy rising steadily (Figure 6). When we increased the training to 20 epochs, accuracy improved further, but we also began to see signs of overfitting, as the training accuracy continued increasing while validation accuracy started to plateau (Figure 7). Finally, training for 35 epochs with 3-layer CNN (Figure 9) did not provide meaningful improvement compared to the 20-epoch model (Figure 8), confirming that additional training did not benefit performance and only increased overfitting risk.

For testing on the second dataset, the model achieved an accuracy of 0.6111, correctly identifying 4,318 out of 7,066 images. The confusion matrix (Figure 10) shows a slightly weaker diagonal compared to Figure 3 (the 4-layer CNN with learning rate 0.001 at epoch 16). This result is expected because the test dataset contains unseen images that were not part of the training set. As a result, the model does not perform as strongly as it does on the data it was trained on. However, Figure 11 shows that the model performed very well on certain emotions, such as surprise (0.760) and happy (0.720), while achieving lower accuracy on emotions like fear (0.472) and sad (0.460).

4 Conclusions

Our experiments show that model depth, learning rate, and epoch size all have meaningful effects on facial-emotion-recognition performance. Among the algorithms tested, the 4-layer CNN consistently achieved the strongest results, reaching the highest accuracy of 0.7200. Learning rate differences had a smaller impact, but the lower rate of 0.001 produced slightly better accuracy and more stable training behavior than 0.01. Epoch comparisons further revealed that while additional training initially improved performance, too many epochs led to overfitting, with little to no gain beyond 20–35 epochs. Overall, deeper models with moderate learning rates and carefully selected epoch counts yielded the best balance between accuracy and generalization. Our model also performed well on the testing dataset, achieving accuracies as high as 0.76 for certain emotions, which is impressive given that this dataset contains entirely new images not seen during training. These findings also give us insight into how the model might perform in real-world conditions, where factors such as lighting, head angles, and subtle expressions introduce even more variation and challenge the model in similar ways to the differences we observed across algorithms tested, training strategies, and unseen test images. At the same time, these systems have practical limitations: they can misread subtle or ambiguous expressions, they may not generalize well across different demographic groups, and they raise privacy concerns because they analyze people's faces without explicit consent. Keeping these constraints in mind is important when considering any real deployment of emotion-recognition technology.

5 Supplemental Information

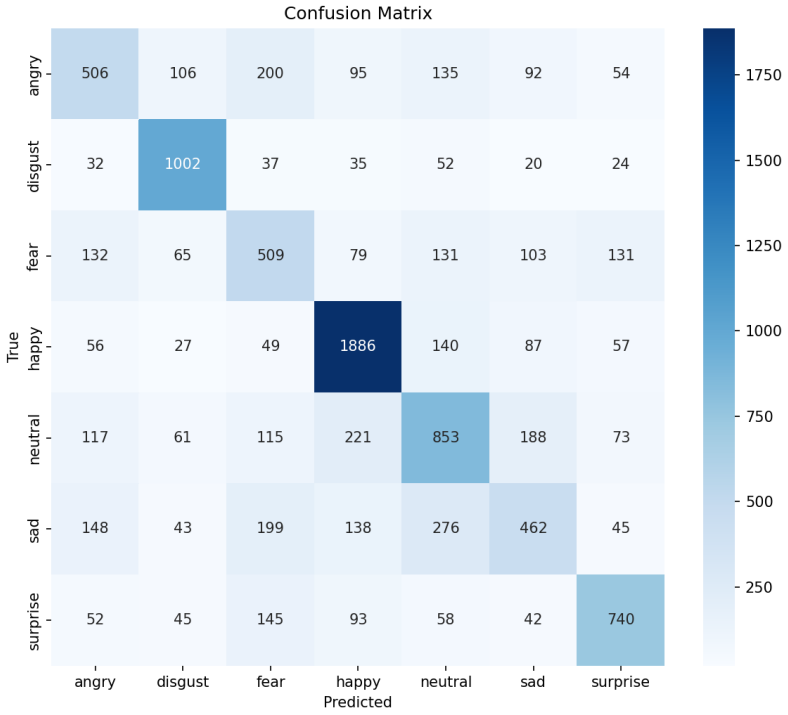

Figure 1: Confusion Matrix for 2-layer CNN with Learning Rate: 0.001, Epoch: 13

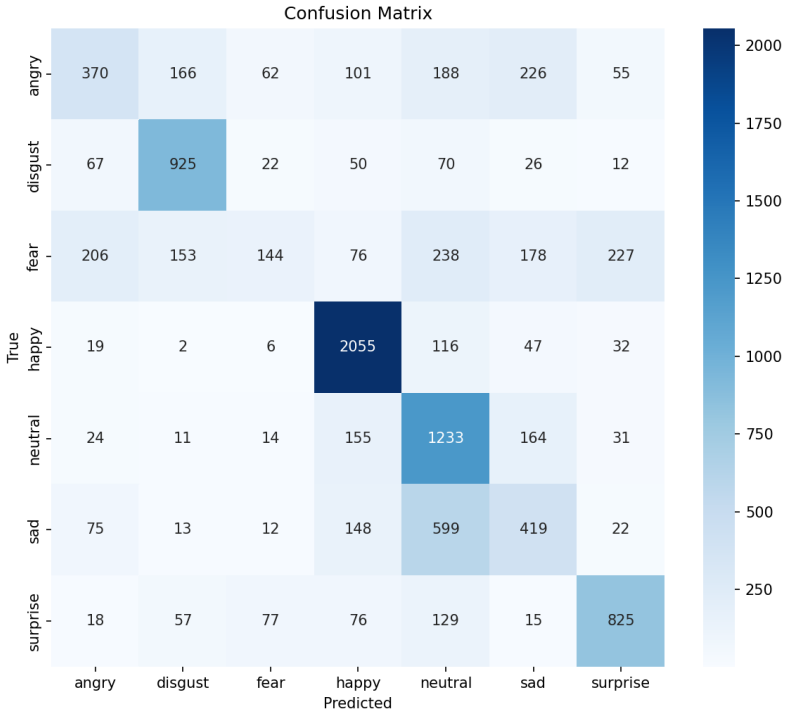

Figure 2: Confusion Matrix for 3-layer CNN with Learning Rate: 0.001, Epoch: 20

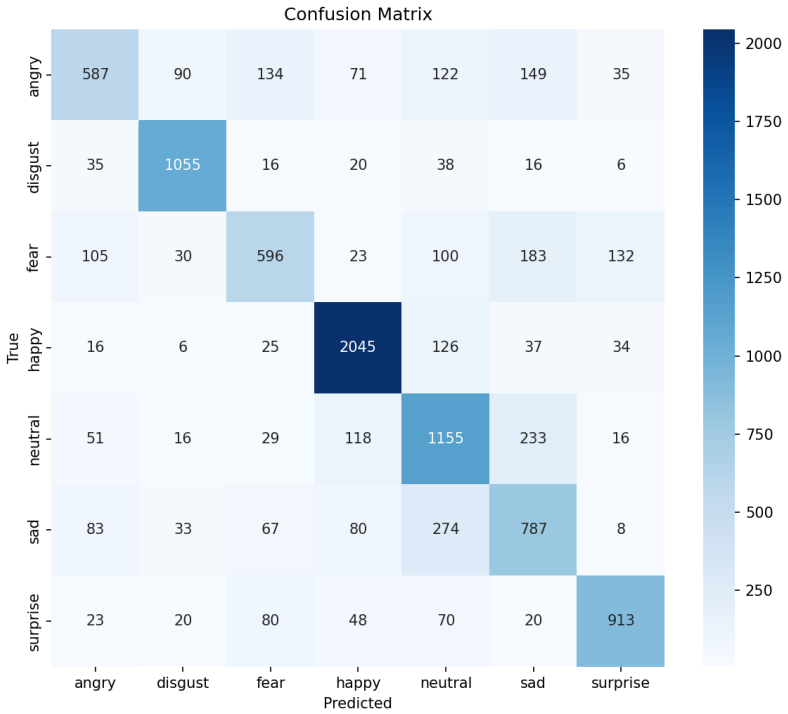

Figure 3: Confusion Matrix for 4-layer CNN with Learning Rate: 0.001, Epoch: 16

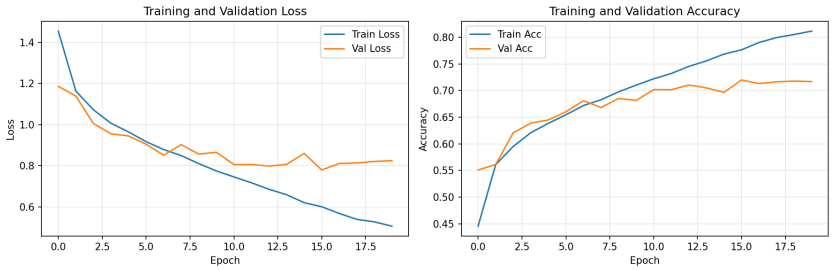

Figure 4: Learning Curves for 4-layer CNN with Learning Rate: 0.001

Figure 5: Learning Curves for 4-layer CNN with Learning Rate: 0.01

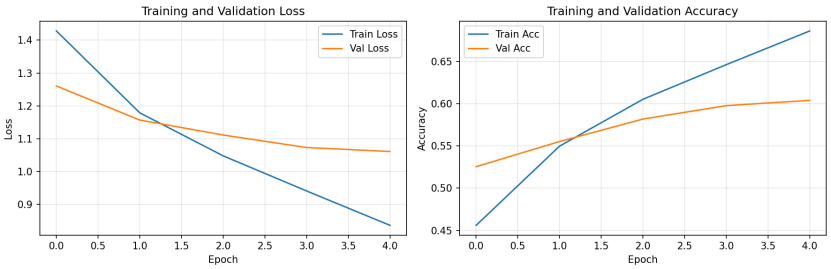

Figure 6: Learning Curves for 2-layer CNN with Epoch: 5

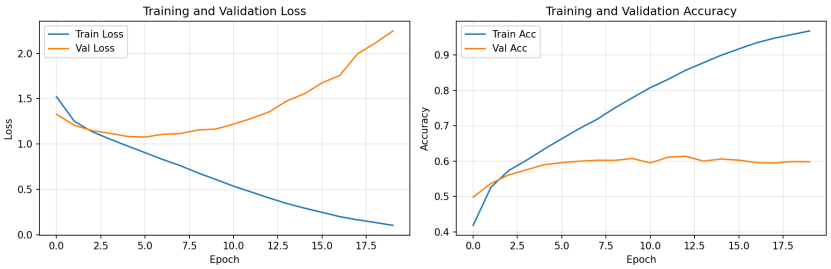

Figure 7: Learning Curves for 2-layer CNN with Epoch: 20

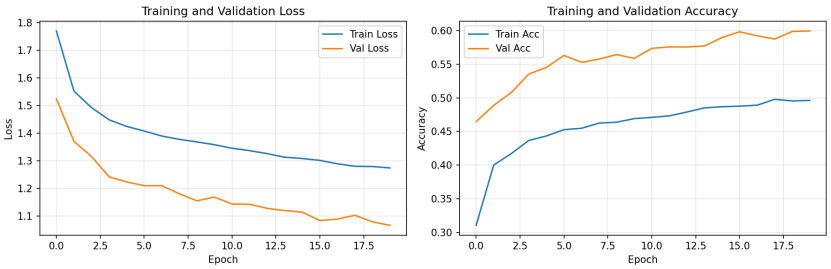

Figure 8: Learning Curves for 3-layer CNN with Epoch: 20

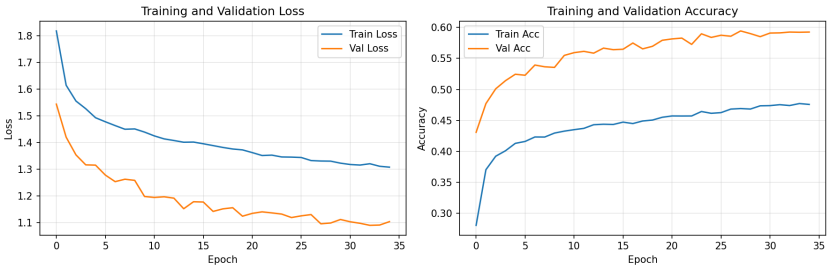

Figure 9: Learning Curves for 3-layer CNN with Epoch: 35

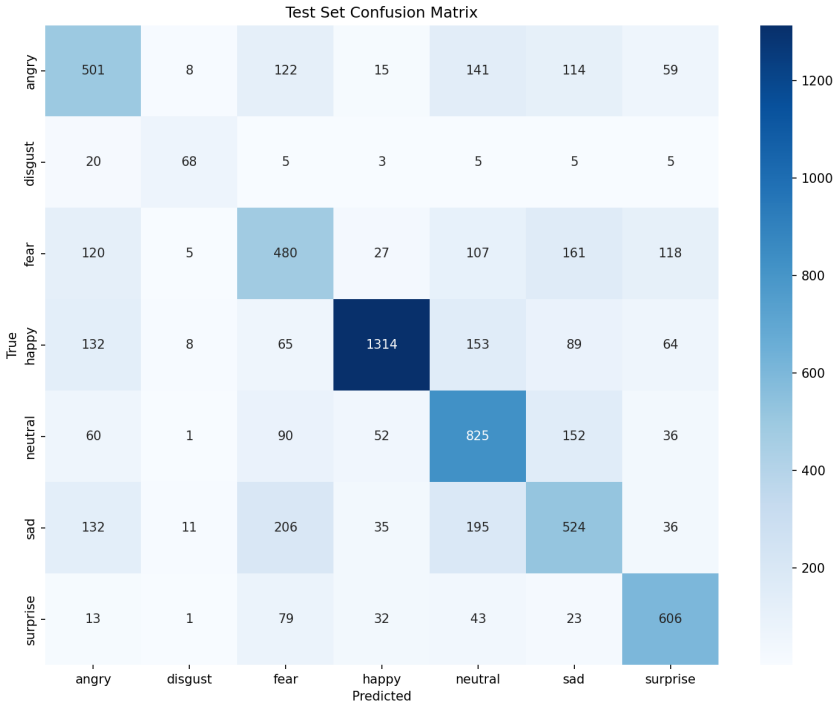

Figure 10: Confusion Matrix for the Test Dataset

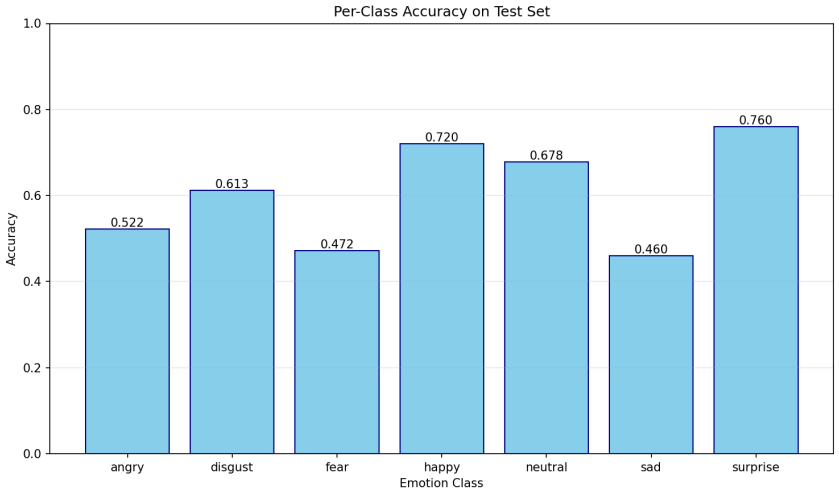

Figure 11: Per-Class Accuracy Plot for the Test Dataset

For our live demo during the presentation, we used the OpenCV environment we built, which was inspired by a YouTube video from Build with Akshit: Emotion Detection using Convolutional Neural Networks and OpenCV.

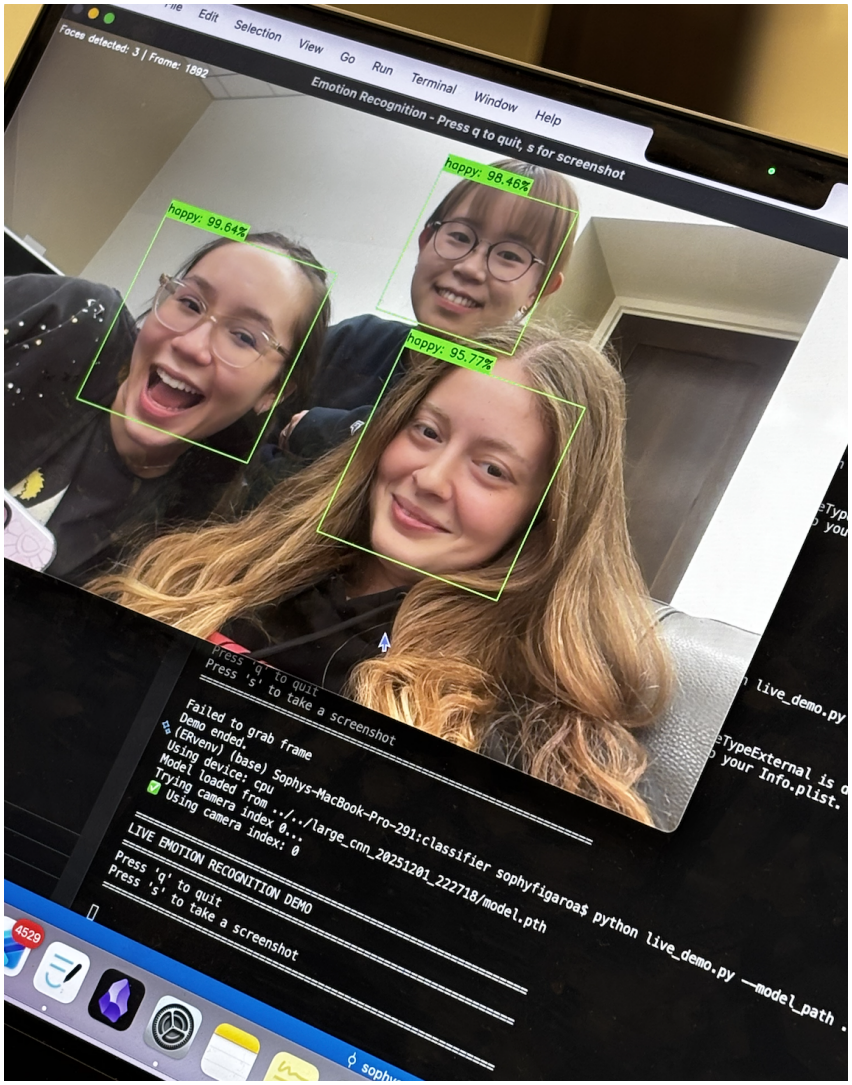

Figure 12: All 3 of us recognized as happy on the OpenCV environment after we finished the project.