Deep Learning Project Report: Facial Emotion Recognition

IST691 Deep Learning in Practice
Professor Mohammed Syed

Submitted: Dec,19 2024
Due: Dec, 19th 2024

Written by:
Christopher Murphy
Nishad Vinayek
John Zumel

Table of Contents

Table of Figures

1. Overview

Facial emotion recognition (FER) began in the 1970s by psychologist Paul Ekman, who established six universal emotions—happiness, sadness, anger, fear, surprise, and disgust. Early computational approaches for detecting human emotion in the 1990s relied heavily on geometric and appearance based methods to analyze facial features and textures. Advancements in facial emotion recognition emerged following the development of deep learning methods, which unlike earlier machine learning applications, are capable of automatically learning features from raw image data.

The primary goal in this analysis is to design a model that accurately classifies images into its corresponding emotion class. Through the integration of convolutional neural networks (CNNs) and preprocessing techniques, this study aims to address issues regarding imbalanced data sets, the variability of facial emotions, and the generalization of emotion recognition.

This report begins by describing the nature of the FER-2013 data set, addressing its components. It then delves into the implementation, covering areas including data preprocessing and model training and tuning. The results are analyzed to assess the model's performance and limitations, which in turn, help uncover potential improvements in accuracy and further optimization.

Facial emotion recognition is an important area in deep learning, with applications ranging from mental health monitoring to human-computer interaction. This project aimed to classify facial images into emotions as established by Ekman with the addition of a "neutral" emotion class—allowing for the differentiation between a total of seven human emotions. A secondary goal was to analyze misclassifications and refine the model by addressing underperforming classes. Through iterative improvements, including fine-tuning and class removal, we explored ways to optimize the model's performance.

2. About the Data

The FER-2013 (Facial Expression Recognition 2013) data set is a publicly available dataset introduced during the ICML 2013 Challenges in Representation Learning. It was found and downloaded from Kaggle. This data set contains a total of 35,887 images, each at 48x48 pixels. These images are organized in separate folders each representing one of the differing seven emotion classes: (0 = Anger, 1 = Disgust, 2 = Fear, 3 = Happiness, 4 = Sadness, 5 = Surprise, and 6 = Neutral)

2.1 Data Exploration

The dataset used was **FER-2013**, which consists of grayscale images of facial expressions. It is divided into three subsets:

- **Training Set**: 28,709 images labeled across seven classes.
- **Validation Set**: A balanced subset of images used for model tuning.
- **Test Set**: 7,177 images for final evaluation.

As depicted in Figure 1, data exploration revealed that the dataset suffered from **class imbalance**, particularly in the "disgust" category, which had significantly fewer samples. In contrast, the “happy” category represented a larger majority of the images when compared to the other six classes. The image quality varied, with some images showing low resolution or poor lighting, requiring preprocessing—as shown in Figure 2. All images were resized to **224x224** and normalized for compatibility with the chosen model architecture.

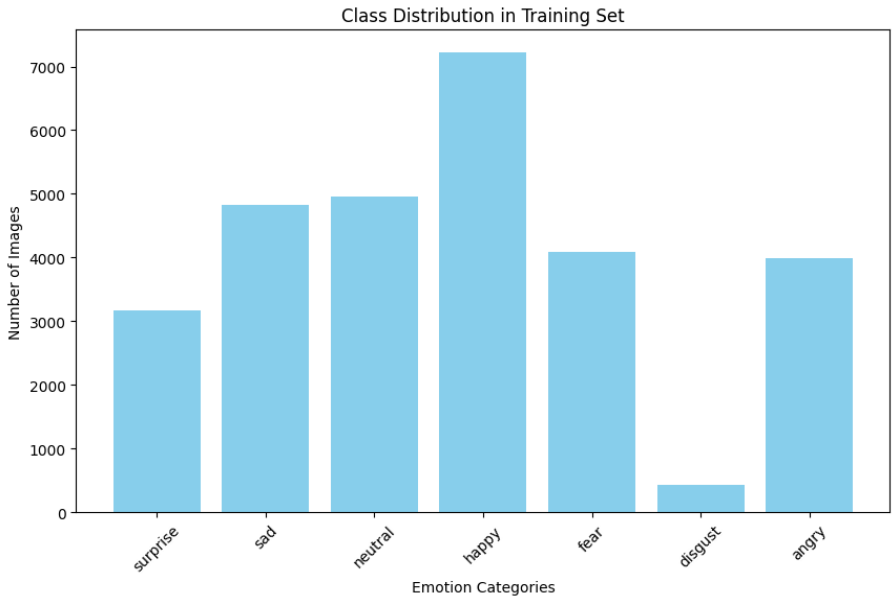

Figure 1: Distribution of Emotion Classes

Figure 2: Example “happy” Image

2.2 Data Preprocessing

After uncovering valuable insights from data exploration. Some data preprocessing techniques were utilized to prepare the data for more accurate model training and accuracy.

The FER-2013 data set was downloaded from Kaggle as a zip file and uploaded to Google Drive. Upon unzipping the file and conducting preliminary data exploration, preprocessing steps including resizing and normalization were conducted to ensure that the image data was prepared for model training. In conclusion of these preprocessing steps, the training set underwent an 80/20 split to create a validation folder—which was stored in the same parent folder where the training and testing set were uploaded. The validation folder was populated with 5,744 images from the 28,709 images from the training set. These 5,744 images moved to the validation folder were then removed from the training set—now containing 22,965 images— to avoid bias and overfitting.

Next, the class imbalance was addressed by deploying data augmentation, specifically to increase the number of images in the “disgust” class. Addressing this class imbalance is crucial in ensuring that the “disgust” emotion is not underrepresented—which may hinder the model’s ability to properly classify images into this class.

The original “disgust” images were augmented using a rotation with a limit of 20 degrees and horizontal, vertical, shear, and zoom shifts with a limit of 10 degrees. Additionally, horizontal flip was enabled to produce further variety without impacting the integrity and features of the images. These augmented images were then uploaded into the same training folder that the original “disgust” images were saved. As shown below in Figure 3, after following these steps, the disgust folder now contained 3,520 images compared to the original 348 images. Initial testing involved more rigorous image augmentation, utilizing greater shifts in degrees, though after running some preliminary models, model accuracy seemed lower than expected. Hence, the more modest shifts of 10-20 degree shifts were deployed which slightly increased accuracy overall.

3. Methods and Experimental Approach

3.1 Initial Model Training

We began with a pre-trained **InceptionV3** model for transfer learning. This approach enabled us to leverage the rich feature extraction capabilities of InceptionV3, which had been pre-trained on the ImageNet dataset. The following steps were taken during this phase:

1. **Model Architecture**: The base layers of InceptionV3 were frozen to retain the generalized features learned from ImageNet, while task-specific layers were added:
    - **Global Average Pooling Layer**: Replaced the dense layers of the base model to reduce dimensionality and prevent overfitting.
    - **Dropout Layer**: A dropout rate of 0.5 was applied to reduce overfitting by randomly deactivating neurons during training.
    - **Dense Layer**: A final fully connected layer with 7 output units (one for each emotion class) and a softmax activation function was added to classify images.
2. **Loss Function and Optimization**: We used the categorical cross-entropy loss function to account for the multi-class nature of the problem. The Adam optimizer, with a learning rate of **1e-4**, provided an adaptive and efficient optimization process.
3. **Training Procedure**: The model was trained on the full training dataset over 30 epochs. The validation dataset was used to monitor performance and tune hyperparameters. Early stopping was implemented to halt training when the validation loss stopped improving.
4. **Data Preprocessing**: Images were normalized to pixel values between 0 and 1, resized to **224x224**, and batched for efficient GPU processing using TensorFlow's `tf.data` API.

Initial results from this model showed a test accuracy of **44.14%**. The model struggled particularly with the "sad" and "disgust" classes, where misclassifications were frequent. Despite these limitations, this initial model provided a strong baseline for further experimentation and refinement.

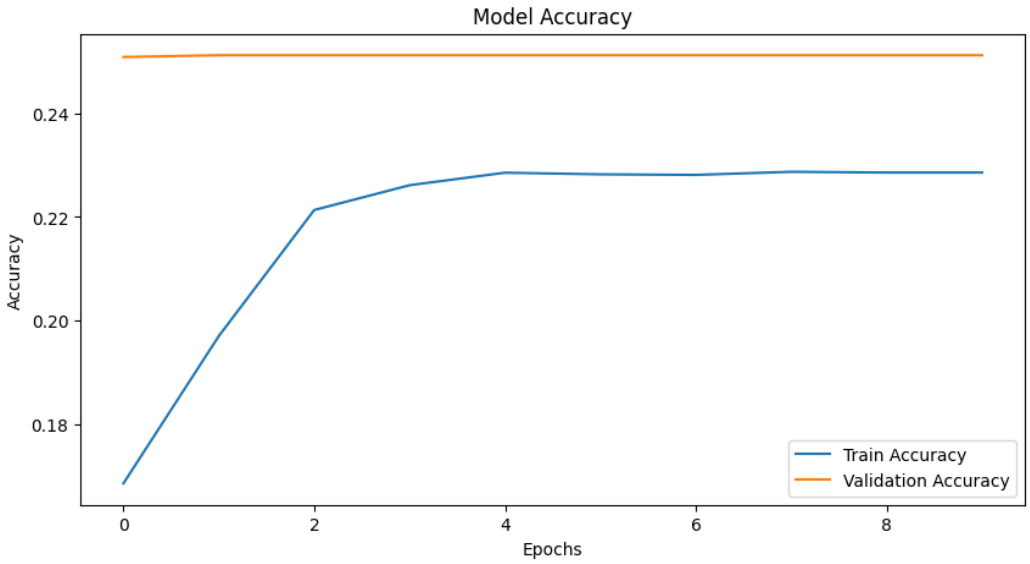

Figure 3a: Base Model Accuracy

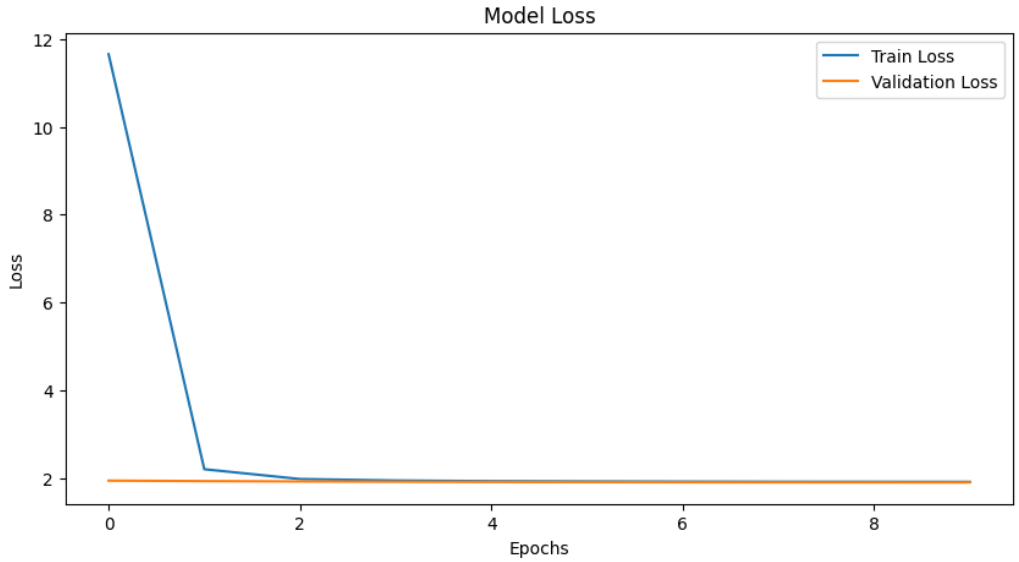

Figure 3b: Base Model Loss

3.2 Fine-Tuning the Model

To improve performance, we unfroze the top layers of the base model and fine-tuned it on the dataset. Key changes included:

1. **Learning Rate Reduction**: The learning rate was reduced to **1e-5** to ensure stable and gradual updates to the model weights. A lower learning rate allowed the model to refine

its understanding of complex features without large weight changes that could disrupt the learned representations.

2. **Layer Fine-Tuning**: Instead of training the entire model, we selectively unfroze the top 50 layers of the InceptionV3 base model. This approach leveraged the general features captured by the earlier layers while focusing on task-specific refinements in the deeper layers. The remaining layers were kept frozen to prevent overfitting and unnecessary computational overhead.
3. **Callbacks**:
    a. **EarlyStopping**: This callback monitored the validation loss and halted training when no improvement was observed for a predefined number of epochs. This prevented unnecessary overfitting to the training data.
    b. **ReduceLROnPlateau**: This dynamically reduced the learning rate when the validation loss plateaued, allowing for finer adjustments during training and improving the convergence process.
4. **Regularization**: To address potential overfitting during fine-tuning, dropout layers (with a rate of 0.5) were included in the custom layers. This randomly deactivated neurons during training, forcing the model to generalize better.
5. **Data Augmentation**: Enhanced the dataset variability by applying augmentations such as random flips, rotations, and brightness adjustments. These augmentations introduced subtle variations in the training images, improving the model's robustness to unseen data.

This fine-tuning significantly improved validation accuracy to **58.22%** and test accuracy to **49.00%**, demonstrating the value of leveraging pre-trained features and focusing training efforts on the most relevant parts of the model.

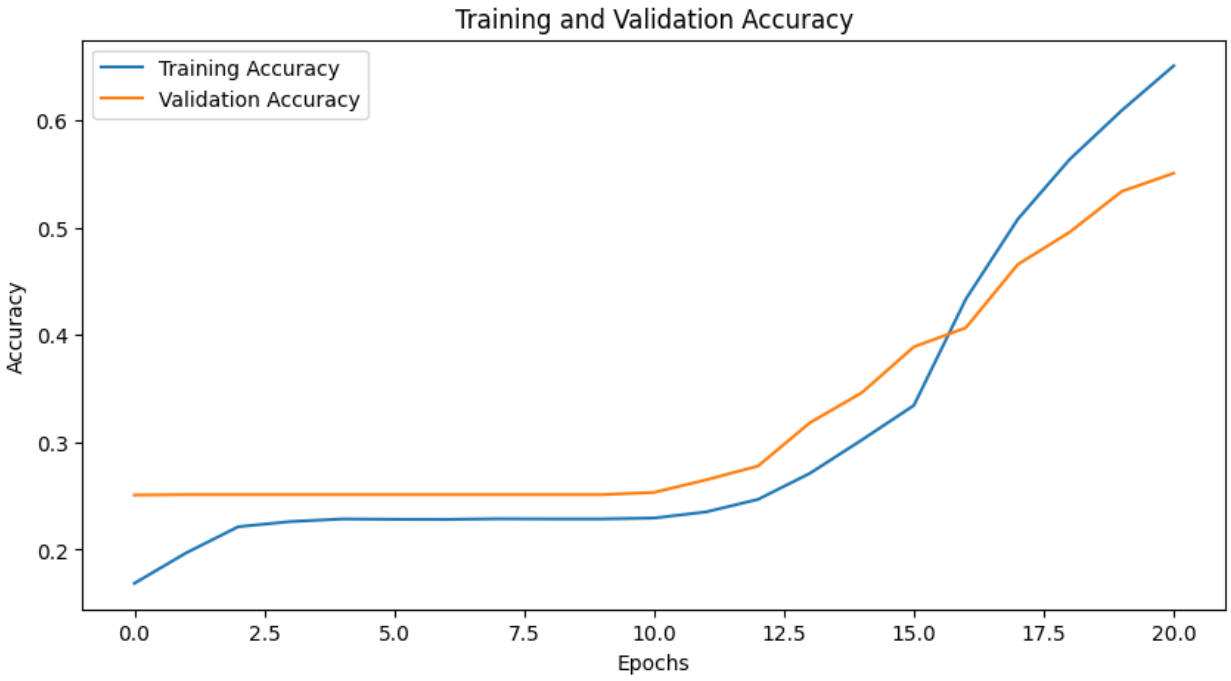

Figure 4a: Fine-Tuned Model Accuracy

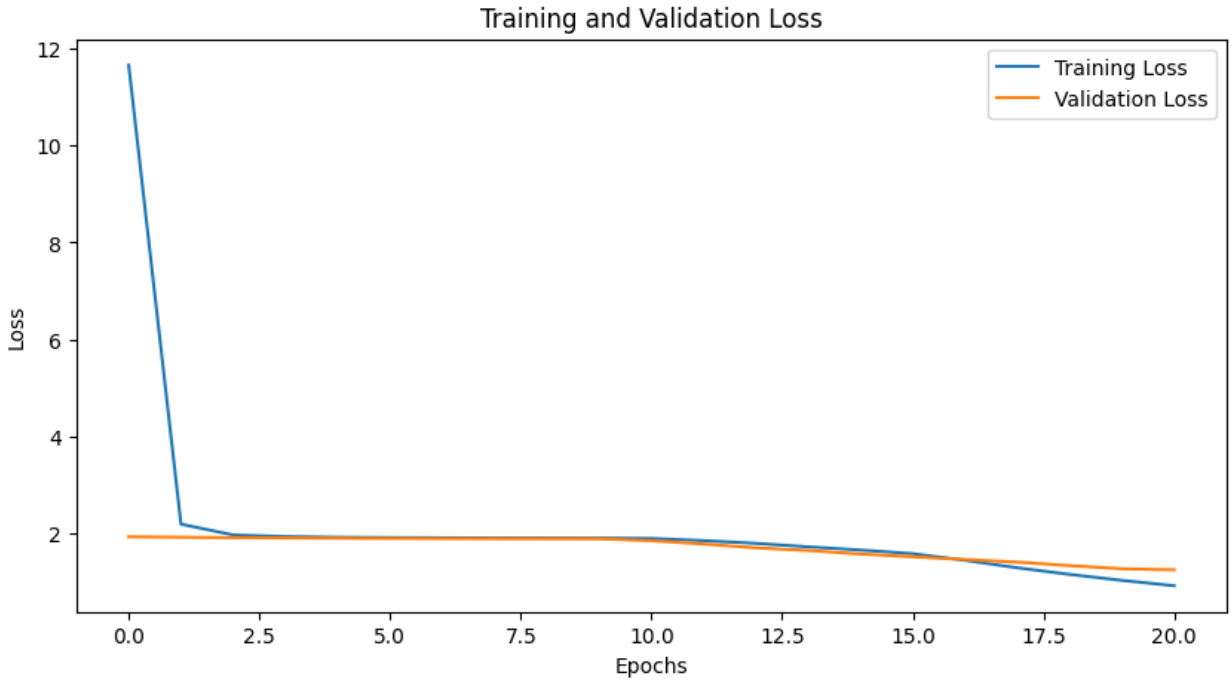

Figure 4b: Fine-Tuned Model Loss

3.3 Addressing Class Imbalance

The dataset exhibited noticeable **class imbalance**, particularly with the *"sad"* and *"disgust"* classes, which had fewer examples compared to dominant classes like *"happy"*. This imbalance led to biased predictions where the model often misclassified minority classes into majority ones. To mitigate this issue, the following steps were implemented:

1. **Class Weighting**: We assigned higher class weights to underrepresented classes to give them a proportional influence on the loss function. This helped the model pay more attention to minority classes during training.
2. **Data Augmentation**: Specific augmentations, such as flips, rotations, and brightness adjustments, were applied more frequently to the minority classes to synthetically increase their representation and improve generalization.
3. **Removing Poorly Performing Classes**: The *"disgust"* class consistently achieved low recall and contributed to confusion in other classes. After multiple experiments, we decided to remove the *"disgust"* class and retrain the model with 6 output classes. This simplification:
    - Improved overall validation accuracy to **67.98%**.
    - Focused the model's learning on the remaining classes, reducing confusion and improving predictions.

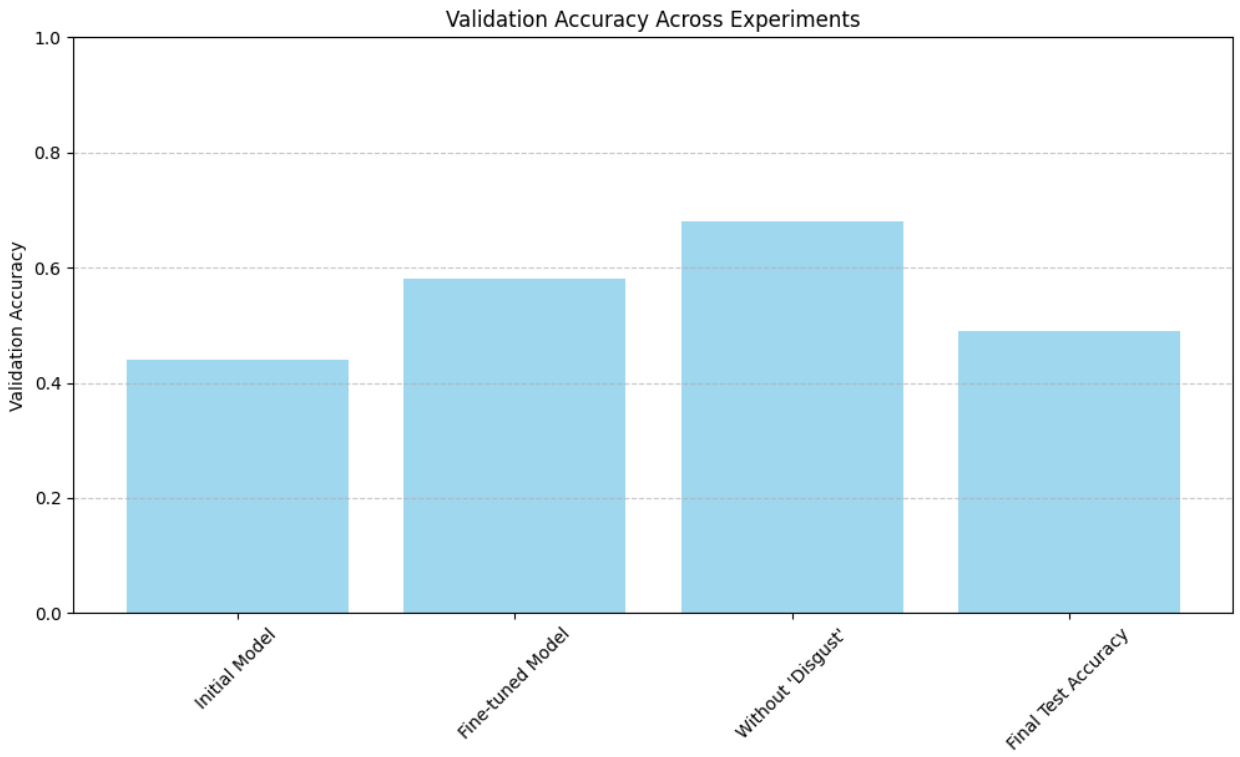

Figure 5a: Comparison of Model Validation Accuracy

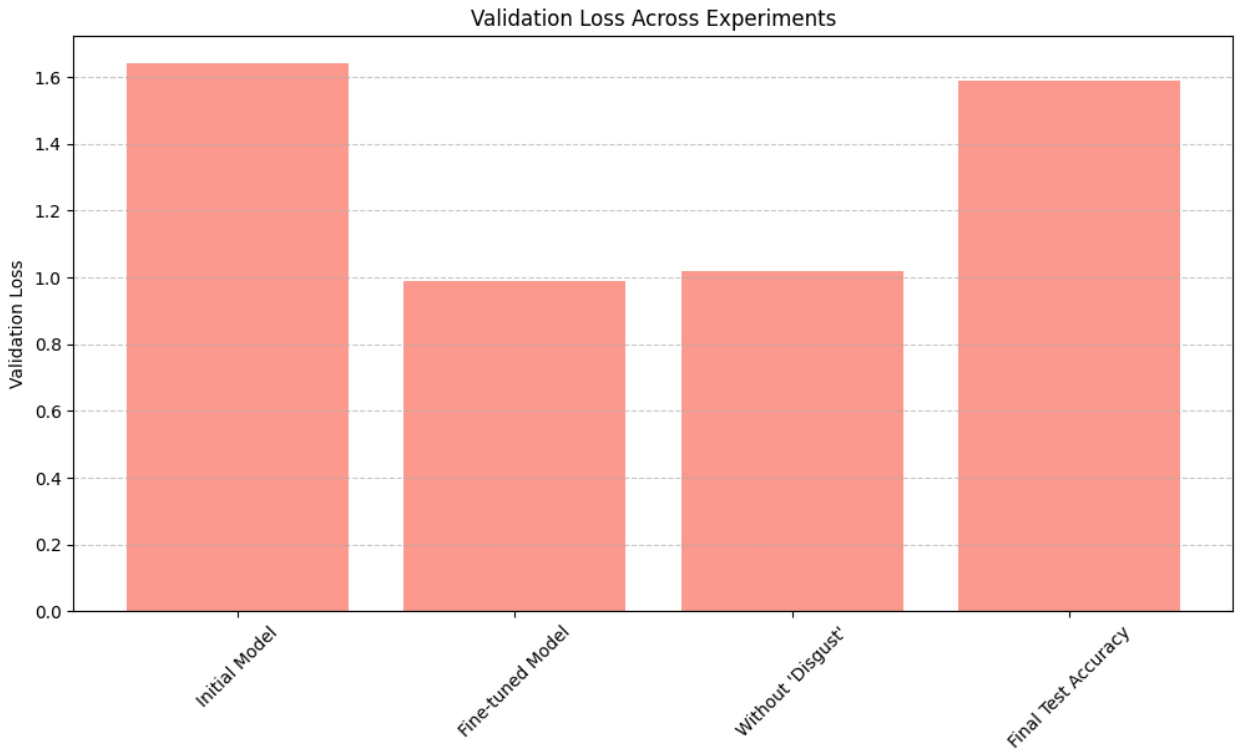

Figure 5b: Comparison of Model Validation Loss

3.4 Hyperparameter Tuning

Hyperparameter tuning played a critical role in optimizing the model. We experimented with multiple parameters to identify the best combination for training. The following adjustments were made:

1. **Batch Size**: We tested batch sizes of 16, 32, and 64. A batch size of **32** provided the best balance between training speed and performance.
2. **Dropout Rate**: Increasing the dropout rate to **0.6** during fine-tuning helped reduce overfitting by forcing the network to learn more robust features.
3. **Learning Rate Scheduling**: The initial learning rate of **1e-4** was reduced dynamically using the *ReduceLROnPlateau* callback. The learning rate decreased by a factor of 0.1 when the validation loss plateaued, leading to better convergence.
4. **Epochs and Early Stopping**: While the model was trained for up to 30 epochs, early stopping ensured that training terminated as soon as performance on the validation set stopped improving.
5. **Augmentation**: Random cropping, flipping, rotation, zooming, and contrast adjustments were applied to all classes, with increased frequency for minority classes to address imbalance.

- flips, rotations, and zooms to improve generalization.

3.5 Interesting Results

The model's performance revealed several noteworthy insights:

1. **Dominance of the "Happy" Class**: The *"happy"* class consistently achieved the best precision and recall across all experiments. This result reflects the distinguishable features of smiles, which are easier for the model to detect compared to subtle or ambiguous expressions.
2. **Challenges with "Sad" and "Disgust" Classes**: The *"sad"* and *"disgust"* classes exhibited consistently poor performance, with low recall and frequent misclassifications. The *"sad"* class was often confused with *"neutral"*, while *"disgust"* overlapped with *"fear"* due to similarities in facial expressions.
3. **Impact of Simplification**: Removing the *"disgust"* class led to a measurable improvement in overall test accuracy, increasing from **44.14%** to **48.66%**. This experiment highlighted the importance of simplifying classification tasks when certain classes perform poorly and contribute disproportionately to errors.
4. **Visualization of Errors**: Visual inspection of true vs. predicted results showed that misclassifications often occurred in images with ambiguous or overlapping facial features. For example, a slight frown could be labeled as either *"neutral"* or *"sad"*, depending on the image quality.

4. Results Summary

4.1 Model Accuracy

The table below summarizes the model's performance across different stages:

Experiment	Validation Accuracy	Validation Loss	Test Accuracy	Test Loss
Initial Model	44.14%	1.64	44.14%	1.64
Fine-Tuned Model	58.22%	0.99	49.00%	1.59
Without "Disgust" Class	67.98%	1.02	48.66%	1.59

The results showed significant improvements after fine-tuning and removing the "disgust" class. Validation accuracy reached nearly **68%**, and test accuracy improved to **48.66%**.

4.2 Training and Validation Accuracy

The accuracy chart illustrates consistent improvements over epochs, with validation accuracy stabilizing around **68%** after fine-tuning.

4.3 Confusion Matrix

The confusion matrix reveals areas where the model struggled, such as misclassifications between "neutral" and "fear."

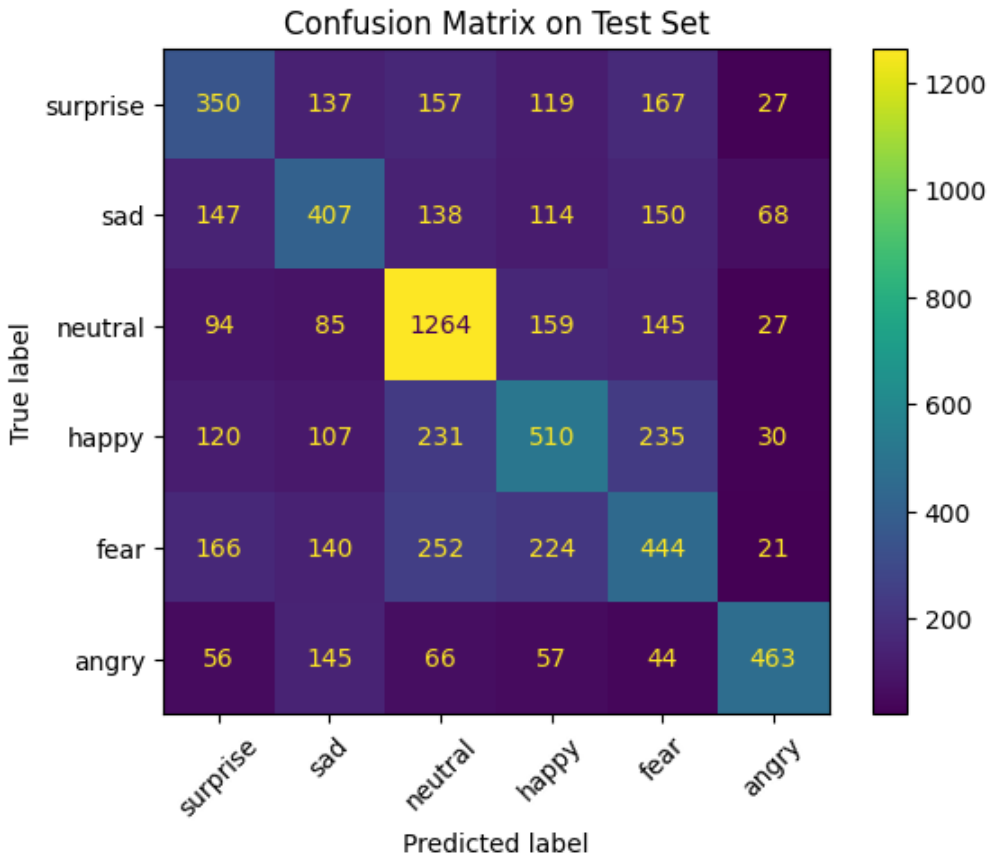

Figure 6: Confusion Matrix on Test Set

4.4 True vs. Predicted Images

To better understand the model's performance and provide a qualitative analysis, we visualized examples of correctly and incorrectly classified images. These visualizations offered the following insights:

1. **Correct Predictions (Green Captions)**: Images with clear, distinguishable features, such as wide smiles for the *"happy"* class or furrowed brows for the *"angry"* class, were classified correctly in most cases. This highlights the model's ability to recognize prominent facial landmarks associated with specific emotions.
2. **Misclassifications (Red Captions)**: Ambiguous facial expressions, low-quality images, or overlapping features between classes often led to errors. For example:
    - Slightly furrowed brows and neutral mouth expressions were misclassified as *"neutral"* instead of *"sad"*.
    - Facial expressions with overlapping features, such as *"fear"* and *"surprise"*, were frequently confused due to shared visual traits like wide-open eyes.

3. **Impact of Class Imbalance**: Misclassifications were more frequent for underrepresented classes like *"sad"* and *"fear"*, highlighting the lingering effects of imbalance despite augmentation and weighting efforts.

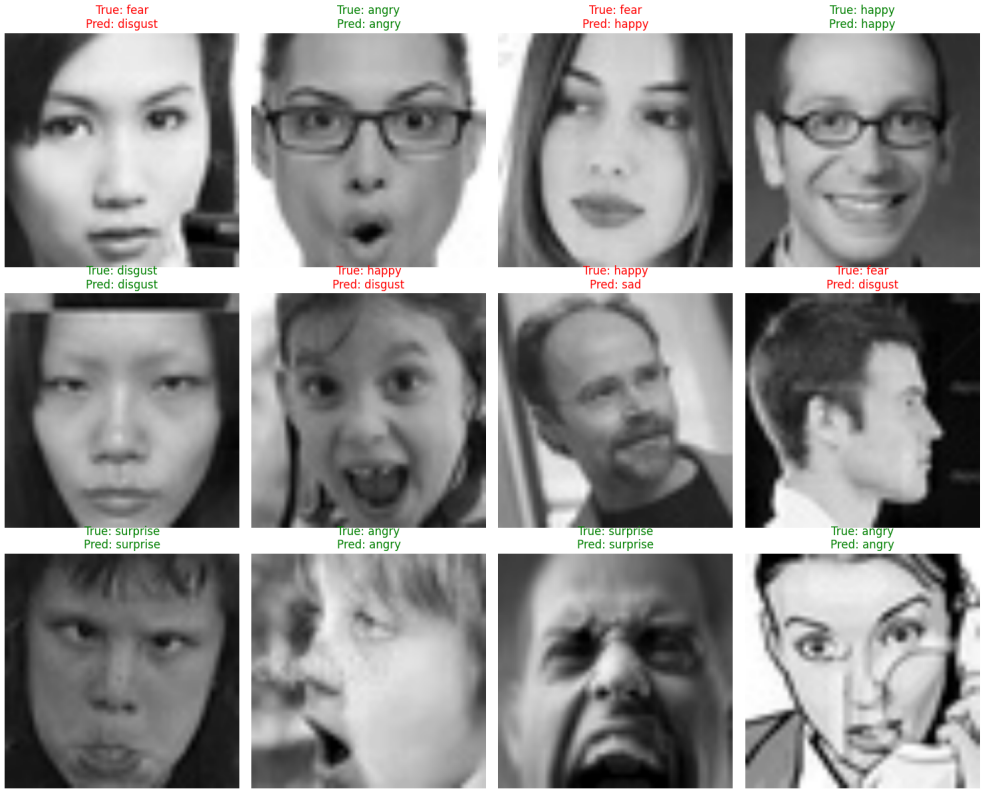

Figure 7: True vs. Predicted Facial Emotions

5. Challenges and Lessons Learned

Designing a facial emotion recognition system presented an array of challenges throughout its development. From managing the quality of the data to accounting for proper representation of classes—to optimize model performance, analyzing the FER-2013 data set instilled the importance of overcoming the nuanced challenges associated with facial emotion recognition. This section highlights a few obstacles faced during the analysis and provides possible remedies that could help mitigate the loss in model accuracy. The following challenges will be covered:

- Class Imbalance
- Data Quality
- Overfitting
- Hardware Constraints

5.1 Class Imbalance

The "sad" and "disgust" classes had significantly fewer samples, leading to poor performance in these categories. Balancing class weights and eventually removing "disgust" improved the model's ability to generalize. Issues regarding the **"disgust"** class arose immediately after data exploration. The "disgust" folder contained a significantly lower amount of images, only representing approximately **1.52%** of the entire training data set. In attempts to remedy this issue, steps for data augmentation on the "disgust" class were taken to further populate this folder. After running preliminary models, accuracy fell below expectations and fine-tuning steps, lowering augment intensity, were taken to ensure the augmented images kept their integrity and features required for successful model training. After tweaking the augmented images, accuracy remained below par. As a result, steps to remove the entirety of the class were taken, increasing overall validation accuracy from approximately 44% to 68%, an improvement of about **24%**.

All things considered, while removing the "disgust" class greatly increased accuracy, it also decreased the overall variability of the data set, which in turn could affect its real-world application. Though underrepresented, disgust is a key emotion involved in facial emotion recognition—as established by Ekman in the 1970s. The absence of this emotion reduces the model's ability to comprehensively classify all human emotions. Ultimately, the decision to remove the "disgust" class as a whole was defined by the desire to improve model accuracy.

Additionally, in further attempt to remedy the class imbalance, we adjusted class weights, increasing weights for underrepresented classes and giving them proportional influence. This helped the model pay more attention to minority classes during training.

5.2 Data Quality

The training set contained 28,709 grayscale images which varied in quality. Most images were low-resolution, 48x48 pixels, and poorly lit which consequently hindered the model's potential. Steps to resize these images to a higher resolution of 224x224 pixels were necessary for compatibility with deep learning tools like CNNs. Though, this process of rescaling introduced a trade-off between resolution and image quality, as resizing did not necessarily help improve the

sharpness of features in images. For further study, it may be beneficial to look into using images with higher quality for model training. Additionally, in cases where high quality images aren't readily available, deploying resolution techniques such as Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) or Super Resolution via Repeated Refinement (SR3). ESRGAN is a deep learning-based method which reconstructs a higher-resolution image from lower-resolution images (Gu et al.). SR3 uses probabilistic models for conditional image generation to create "super-resolution" images. The model iteratively refines the generated outputs, in this case images, using a U-Net model (Xintao et al.).

5.3 Overfitting

Resulting directly from the aforementioned challenges, it was inevitable to come across the obstacle of overfitting. Given the limited variety of data, imbalanced classes, and poor image quality, the model was prone to memorizing features in the training set rather than learning generalized features. When considering this hindrance at large, significant drops in validation and test accuracy raise the question of whether changes in data preprocessing steps were necessary. After rounds of tweaking data preprocessing steps and in considering the time constraints of this project, a large scale change was not feasible. In attempts to remedy this obstacle, our team attempted to optimize current techniques and adjusted model architecture. Smaller tweaks included adjusting dropout, reducing learning rate, and deploying stronger augmentations.

5.4 Hardware Constraints

Another challenge encountered during model training was limited computational power. Because of the large data set being used and the training of complex deep learning models, running code took some members several hours to load. Additionally, in some cases, running code took so long that the process would time out. In an effort to combat this, initial testing involved sampling only 750 images from every class—giving a total training set size of 5,250 images. By reducing the training set to fewer images, training the model became a faster process. However, as a result, the highest accuracy provided by a model was approximately **54%**. In attempts to increase accuracy, we resampled 2500 images from every class, which yielded an accuracy of **62%**, an increase of about **8%**. In considering this change, it's important to note that as we downsized the total number of images being used for training, run-time and accuracy decreased. In contrast, as we increase the number of images for training, run-time and accuracy increases, illustrating the trade-off between computational efficiency and model performance. In the end, group members upgraded to Google Colab Pro to access GPU acceleration, noticeably improving efficiency.

6. Conclusion

Facial emotion recognition has the potential to transform a variety of industries including security and healthcare. As demonstrated in this analysis, achieving high classification accuracy has its own set of challenges. Despite the aforementioned challenges regarding data imbalance, data quality, and among others, some valuable insights regarding necessary changes in data preprocessing and model architecture were extracted, significantly impacting model performance.

This project demonstrated the importance of iterative experimentation in deep learning. Starting with a pre-trained InceptionV3 model, we incrementally improved performance through fine-tuning, hyperparameter adjustments, and class simplification. The final validation accuracy reached **67.98%**, and test accuracy improved to **48.66%**. These results highlight the value of transfer learning and systematic refinement.

Future work could include testing alternative architectures like MobileNetV2 for faster training, augmenting the dataset further, and exploring techniques like ensembling to improve generalization. This project underscores the iterative nature of machine learning, where trial and error play a vital role in achieving meaningful results.

7. References

Chollet, Francois. “Keras documentation: Transfer learning & fine-tuning.” *Keras*, 15 April 2020, https://keras.io/guides/transfer_learning/.

Gu, Jinjin, et al. *ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks*, 2018, p. 23. *ResearchGate*, https://www.researchgate.net/publication/327434971_ESRGAN_Enhanced_Super-Resolution_Generative_Adversarial_Networks.

Xintao, Wang, et al. *Image Super-Resolution via Iterative Refinement*, vol. 2, 2021. *arxiv*, https://arxiv.org/abs/2104.07636.