SmartClass A.I.ssistant Report

By

Mamadou Diao Kaba (27070179) Training Specialist

Jaskirat Kaur (40138320) Evaluation Specialist

Kaloyan Kirilov (40245658) Data Specialist

Github repo: https://github.com/mdkaba/SmartClass-A.I.ssistant.git

"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality"

Signed by:

Mamadou Kaba (27070179)

Jaskirat Kaur (40138320)

Kaloyan Kirilov (40245658)

Table of Content

1. Introduction

This report details first the process of data collection, cleaning, labeling, and preliminary analysis for developing suitable datasets to train a Convolutional Neural Network (CNN) for facial expression recognition. The objective was to gather and prepare diverse datasets, map them to specific classes, and perform necessary preprocessing and analysis. We combined the FANE Facial Expressions and Emotion Dataset and the AffectNet Training Dataset, ensuring they met our criteria for diversity and relevance. This involved standardizing image sizes, normalizing pixel intensity values, and addressing ambiguities in expression labeling. Through meticulous data cleaning and merging, we aimed to create a robust and consistent dataset for effective model training and evaluation.

Then, followed the process of training various Convolutional Neural Network (CNN) architectures for facial expression recognition using the meticulously prepared dataset from the previous data cleaning phase. The primary objective was to develop, train, and evaluate models capable of accurately classifying facial expressions into predefined categories. Using our cleaned dataset, we implemented three distinct CNN architectures to compare their performance. The training process involved defining optimal model architectures, applying suitable activation functions, and incorporating regularization techniques to enhance generalization. We employed PyTorch and torchvision for constructing and training the models, while scikit-learn facilitated data splitting and evaluation. Each model was trained with carefully chosen hyperparameters, and the best-performing models were saved for further analysis. We aimed to identify the most effective CNN architecture for robust facial expression recognition.

Finally, we extended our efforts by addressing potential biases in our dataset related to race and gender. The goal was to detect and mitigate any biases to ensure fair and balanced model performance across different demographic groups. This involved creating new bias-specific datasets, including Asian, Black, and White datasets, as well as Male and Female datasets. We refined these datasets by adding synthetic images and removing non-expressive images, resulting in more balanced classes. Using these improved datasets, we trained and evaluated our models to analyze and mitigate bias. We also implemented a 10-fold cross-validation approach to enhance the robustness and reliability of our model evaluation. This section details the steps taken for bias detection and mitigation, as well as the cross-validation results, providing a comprehensive analysis of our efforts to create a fair and effective facial expression recognition system.

2. Data Cleaning

2.1. Datasets

We combined the FANE Facial Expressions and Emotion Dataset with the AffectNet Training Dataset, two publicly accessible datasets, for this study. These datasets were selected because they contain a large number of facial expressions, which are necessary for our Convolutional Neural Network (CNN) to be trained to identify and classify different emotional states.

- FANE Facial Expressions and Emotion Dataset:
  - Total Number of Images: 16,913 images
  - Number of Classes: 7 (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral)
  - Number per class:
    - Angry: 1774 images
    - Happy: 1922 images
    - Neutral: 1835 images
  - Special Characteristics: The dataset consists of a wide variety of facial expressions and a substantial number of images, which is useful for precise analysis and machine learning models. Each image is annotated with labels indicating the emotion being expressed. This dataset includes images of individuals from diverse demographic backgrounds, including different ages, genders, and ethnicities.
  - License: CC0: Public Domain

- AffectNet Training Dataset:
  - Total Number of Images: Approximately 29,000
  - Number of Classes: 7 (Neutral, Happy, Sad, Surprise, Fear, Disgust, Anger, and Contempt)
  - Number per class:
    - Angry: 3218 images
    - Happy: 5044 images
    - Neutral: 5126 images
  - Special Characteristics: This dataset has a large and diverse range of facial expressions. It consists of RGB images (96x96 pixels) captured in uncontrolled conditions, providing a variety of lighting and backgrounds. Includes both posed and spontaneous expressions, enhancing the dataset's diversity.
  - License: Available for educational and research purposes under specific terms and conditions

2.2. Cleaning

2.2.1. Justification for Dataset Choices

The FANE and AffectNet datasets were chosen due to their diversity and size, providing a large and varied training set essential for developing a robust model. Both datasets include the necessary emotional categories, ensuring the model can effectively recognize Neutral, Angry, and Happy states. The FANE dataset offers high-resolution images and a mix of posed and spontaneous expressions, while the AffectNet dataset provides a vast number of images under varied conditions. Combining these datasets leverages their complementary strengths. Additionally, these datasets are well-recognized in facial expression research, allowing us to benchmark our results against existing studies.

2.2.2. Challenges and Solutions

We faced several challenges during data collection and preprocessing. One major challenge was the inconsistent image sizes and formats between the datasets. The FANE dataset includes high-resolution images of varying sizes, while AffectNet images are uniformly 96x96 pixels. To address this, we've resized all images to a standard size of 96x96 pixels. Another issue was the presence of duplicate images in the AffectNet dataset, which could disrupt the training process. We resolved this by implementing a Python script to detect and remove duplicate images, ensuring the uniqueness of each image in the dataset.

The diverse lighting conditions and backgrounds in the images also posed a challenge, as these variations could affect the model's accuracy. We applied normalization techniques to standardize the pixel intensity across all images, mitigating the impact of these variations. Additionally, we had to manually collect images representing the focused state, as neither dataset included this category. This involved sourcing images from various online platforms, cropping them to focus on the face, normalizing, and resizing them to 96x96 pixels. This manual effort, although time-consuming, was necessary to ensure consistency and accurate representation of the focused expression in our dataset.

2.2.3. Changes Made to the Dataset for Bias Mitigation

To mitigate bias in our system, we made several changes to our dataset, focusing on balancing the representation of different demographic groups. Specifically, we added synthetic images from Generated Photos to increase the representation of underrepresented groups. Additionally, we removed images from the AffectNet dataset that were not expressive enough for a specific class, ensuring that each class had clear and distinguishable expressions. These steps were crucial in creating a more balanced and fair dataset for training our models. After the bias mitigation steps, our final dataset, termed the "Definitive Dataset," consists of images from various demographic groups, categorized by both race and gender. The dataset includes the following subcategories:

- **Bias Dataset:** Includes these subcategories: Asian Dataset 2.0, Black Dataset 2.0, Female Dataset, Male Dataset 2.0, and White Dataset.

The images in the definitive dataset are labeled with their respective expressions, races, and genders, following a systematic and standardized naming convention to maintain consistency.

By making these changes, we aimed to enhance the model's ability to generalize across different demographic groups, thereby reducing potential biases in facial expression recognition.

2.2.4. Information Provenance

Image Batch	Source	License Type	Link
FANE	Kaggle, Generated.photos	CC0: Public Domain	FANE Dataset[1] Generated Photos
AffectNet	Kaggle, Generated.photos	Educational and Research Use	AffectNet Dataset[2] Generated Photos
Focused/Engaged	Google Images, Unsplash, iStock, Adobe Stock, Generated.photos	Various (Public Domain, Stock Photo Licenses)	Google Images, Unsplash, iStock, Adobe Stock, Generated Photos

Table 1: sources of each image batch, including links to the datasets and the licensing type

The type of images mainly used in our dataset are frontal face shots, and the face expression is mostly centered in the images. We roughly aim to have around 500 images per class or more. The extra class we chose to do was happy since that seemed to be more distinct and different compared to the original three that we needed to include. With the definitive dataset, we ended up having 2221 images in total (around 550 images per class).

2.3. Data Cleaning

2.3.1. Cleaning Process

We worked with two datasets for facial expressions (Angry, Neutral, and Happy): the FANE Facial Expressions and Emotion Dataset and the AffectNet Training Dataset. To ensure consistency across the datasets, we applied several standardization techniques while addressing various challenges. One significant challenge was the presence of duplicate images in the FANE dataset. These duplicates could skew the training process if not properly addressed. To tackle this, we implemented a python script (duplicate_check.py)[3] to detect and delete duplicate images, thus ensuring a unique and diverse set of images for each expression. This step was necessary to maintain the integrity and variability of the dataset. Then, we tackled the varying image sizes. The FANE dataset included images of different dimensions, while the AffectNet images were uniformly sized at 96x96 pixels. We decided to resize all images to 96x96 pixels for uniformity. For the FANE dataset, we selected images larger than 96x96 pixels and resized them using a script (resize.py)[4]. This step was essential to streamline the preprocessing pipeline and maintain a consistent input size for our model.

Normalization was another crucial step in our standardization process. By normalizing the pixel intensity values, we ensured that all images fell within a consistent range, which is vital for stable and effective model training. We coded a Python script (normalize.py)[5] to perform this task on all selected images. The normalize.py script standardizes pixel intensity values of images by scaling them to a range of 0 to 1 using the cv2.NORM_MINMAX method. This process adjusts the minimum pixel value to 0 and the maximum to 1, with intermediate values scaled proportionally, ensuring consistent intensity distribution across all images. Each RGB channel is normalized independently, preserving the relative differences between channels while maintaining color composition. After normalization, the values are scaled back to the 0-255 range for compatibility with standard image formats. This uniform scaling is particularly beneficial for our dataset, as it combines two datasets (FANE and AffectNet) with varying image sizes and intensity distributions. By normalizing the images, we ensure that both datasets have similar intensity distributions, improving the training process of our machine learning model by providing consistent input data and enhancing the model's performance.

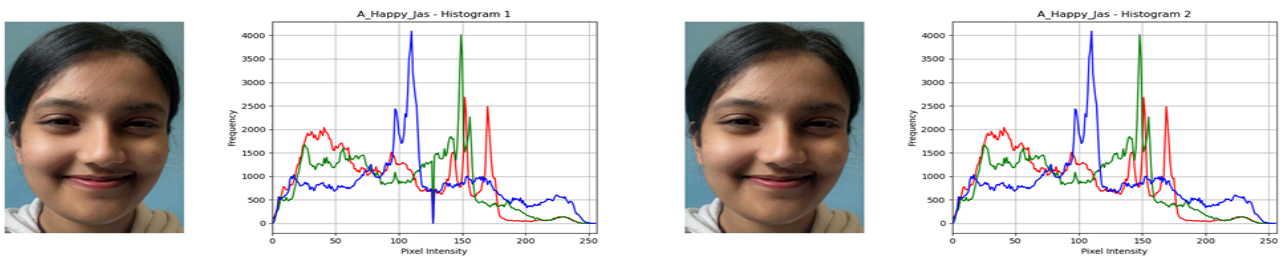

Figure 1: Pixel Intensity Correction of an Original Picture vs Normalize Picture

For the Focused expression, we manually collected images from various online sources, including Unsplash, Google Photos, and Shutterstock. These images were often full-body shots, requiring cropping to focus on the face and a small portion of the background, aligning them with the other datasets. After cropping, we normalized these images using the same script and resized them to 96x96 pixels. This method was also applied to pictures taken by team members. Collecting and standardizing these images presented a challenge due to the manual effort involved, but it was crucial to ensure consistency across all expressions.

Figure 2: Original Image vs Cropped Image vs Normalized Image

Figure 3: Cleaned Image (96x96)

Through these systematic steps, we successfully standardized our dataset, creating a consistent and robust set of images for training our model. The before-and-after examples illustrate the impact of our standardization techniques, highlighting improvements in image uniformity and quality, which are crucial for the model's performance.

2.3.2. Added Images Cleaning Process

For the synthetic images added to mitigate bias, the same data cleaning process was applied. Since these images were already well-cropped and clear, no additional cropping was necessary. The steps followed were:

- **Resizing:** The synthetic images were resized to 96x96 pixels.
- **Normalization:** The pixel intensity values were normalized.

These cleaning steps were implemented using the existing data cleaning scripts, ensuring consistency across the entire dataset.

2.4. Labeling

2.4.1. Labeling Process

The FANE and AffectNet datasets were already categorized by expressions, making it easier to extract the appropriate labels for our project: Angry, Neutral, and Happy. We further filtered these datasets to include only those images with the most defined expressions, ensuring a clearer distinction between similar expressions such as Neutral and Focused. This refinement process was crucial to avoid confusion in the dataset and improve the model's accuracy in recognizing subtle differences. To address potential biases, we ensured that the datasets had a proportional representation of different demographics, including a balanced number of light-tone females, light-tone males, dark-tone females, and dark-tone males. Ensuring diversity in the dataset helps create a model that works effectively across various origins, genders, and skin colors.

The datasets did not include images representing the Focused state. To avoid confusion with the Neutral state, we manually collected images for this category from sources such as Google Images, Unsplash, iStock, and Adobe Stock. We applied the same bias filters to these manually collected images to maintain diversity. Manual selection was essential to ensure that the Focused images were distinct from Neutral and accurately represented the focused expression. During the data collection process, we encountered several ambiguities, such as differentiating between similar expressions and handling images with mixed emotions. For similar expressions like Neutral and Focused, we used the most defined expressions for precise labeling. If a person's expression was clearly focused, it was labeled as Focused. If there was ambiguity, it was labeled as Neutral. For combined emotions, we selected the predominant emotion when reviewing the dataset and filtered out images with a predominant emotion that was not part of our labels.

After selecting images from the FANE, AffectNet, and manually collected Focused datasets, we merged them into a single dataset with four standardized class labels: Angry, Happy, Focused, and Neutral. For consistency, we mapped labels from different datasets to standardized class labels. For instance, FANE's "Angry" and AffectNet's "Anger" were both mapped to "Angry," and FANE's "Happy" and AffectNet's "Happy" were both mapped to "Happy." This standardization ensured uniformity across the combined dataset. The process of merging the datasets presented several challenges, including maintaining a uniform format for all images from different sources. The images varied in quality, lighting conditions, and backgrounds, making it difficult to achieve consistency. To address this, we standardized all images by using the methods provided in the Data Cleaning Section .

By addressing these ambiguities and challenges during the retrieval, labeling, and merging of datasets, we ensured that our combined dataset was accurately labeled and suitable for training a Convolutional Neural Network (CNN) to classify facial expressions.

2.4.2. Bias and Definitive Datasets Labeling Process

For the bias detection and mitigation in our project, we focused on two attributes: race and gender. The labeling process was detailed and methodical to ensure accurate categorization of the images based on these attributes.

- **Initial Manual Sorting:**
    - **Gender-Based Datasets:** We began by taking the original dataset and manually sorting images into separate folders based on gender. For instance, to create the female dataset, we navigated to the corresponding class folders in the original dataset (e.g., Original Dataset/Angry) and manually moved all images of females into the corresponding class folders in the female dataset (e.g., Female Dataset/Angry). The same procedure was followed to create the male dataset.

    - **Race-Based Datasets:** Similarly, we manually sorted images based on race to create the Asian, Black, and White datasets. Images were moved from a duplicate original dataset's class folders into the appropriate class folders in the respective race-based datasets.

- **Automated Labeling:**
    - After manually sorting the images, we employed a script called bias_dataset_label.py to label all images in the bias datasets. The script renamed the images in the format Expression_Race_Index or Expression_Gender_Index. This systematic approach ensured that each image was labeled correctly based on its classification.

- **Definitive Dataset Labeling:**
    - For the definitive dataset, which included all images from the race and gender bias datasets, we used another script named definitive_dataset_label.py. This script labeled the images in the format Expression_Race_Gender_Index. This comprehensive labeling ensured that each image was uniquely identified by its expression, race, and gender.

2.5. Data Visualization

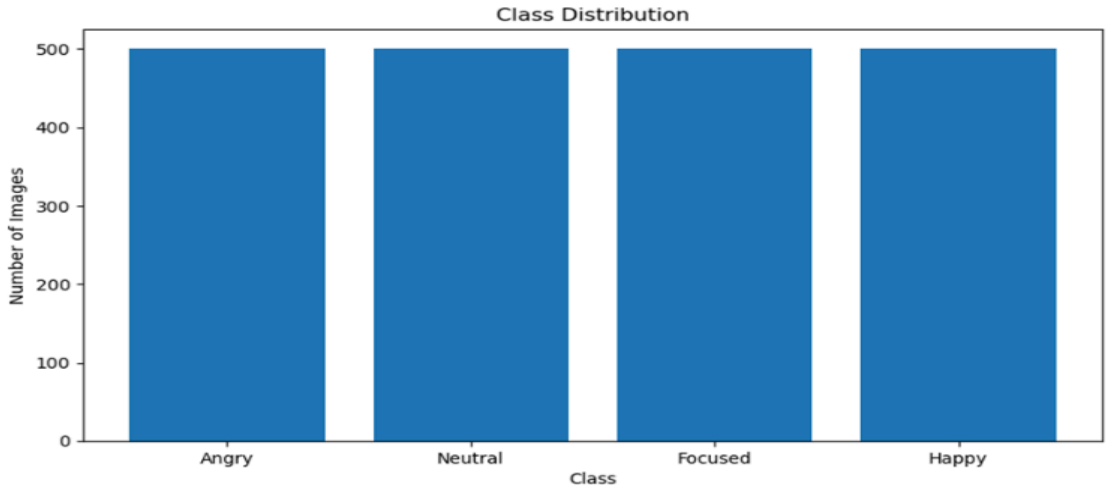

Figure 4: Bar Distribution of the number of Images in Each Class in the Original Dataset (500 each)

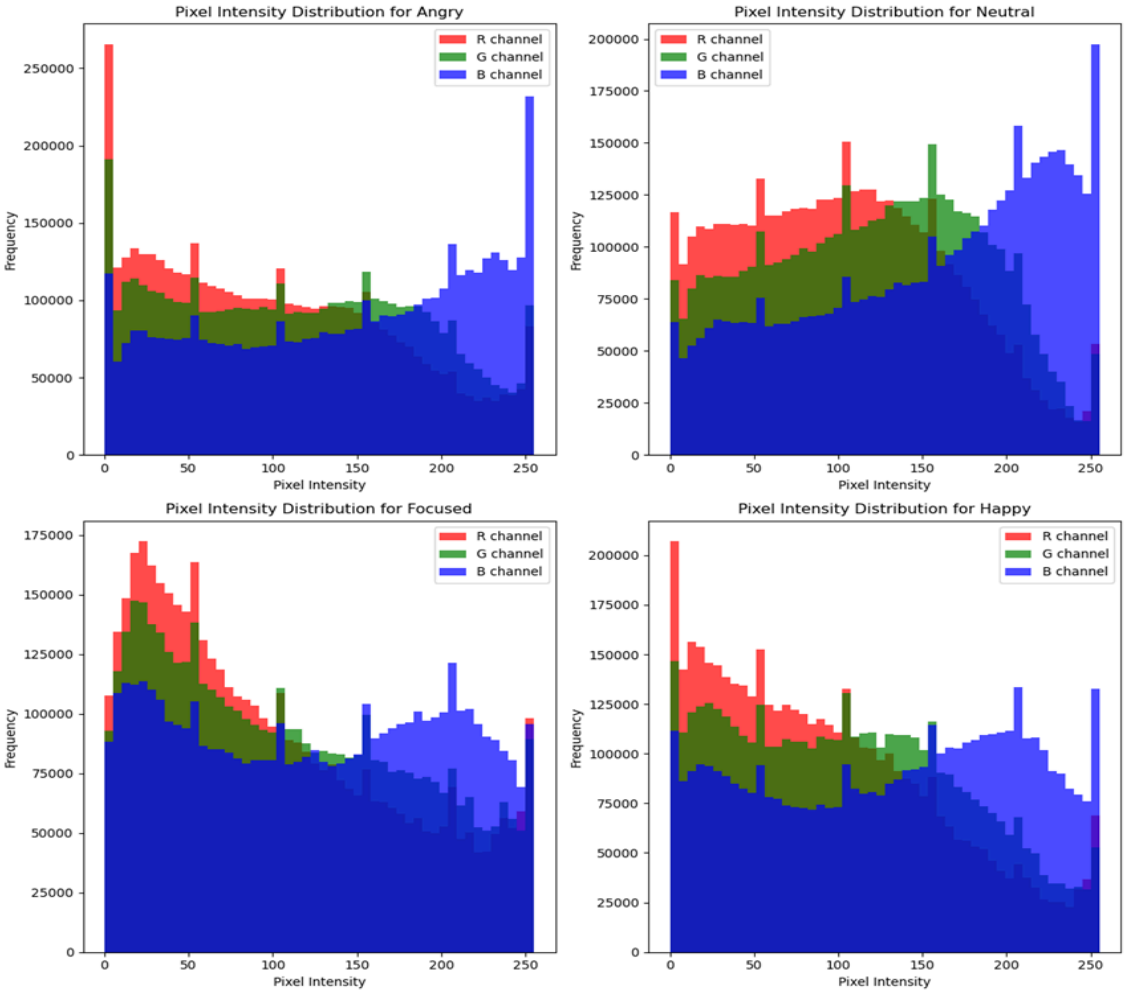

Figure 5: Pixel Intensity Distribution for Each Class in the Original Dataset

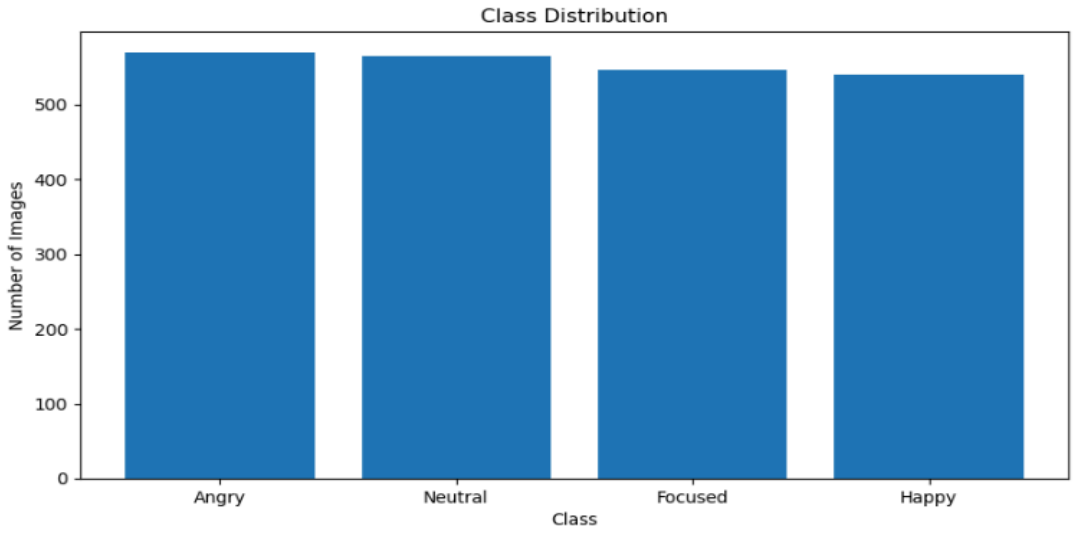

Figure 6: Bar Distribution of the number of Images in Each Class in the Definitive Dataset (569, 565, 547, 540)

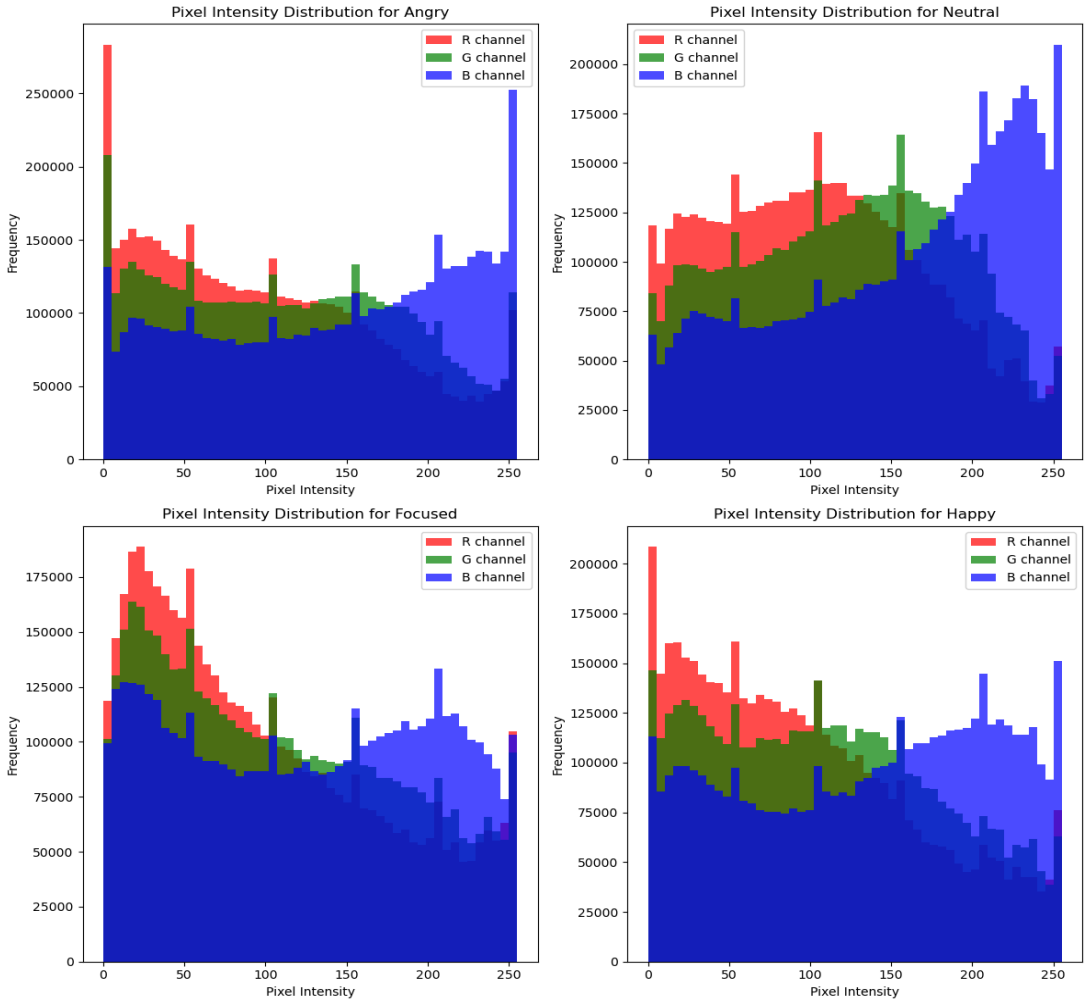

Figure 7: Pixel Intensity Distribution for Each Class in the Definitive Dataset

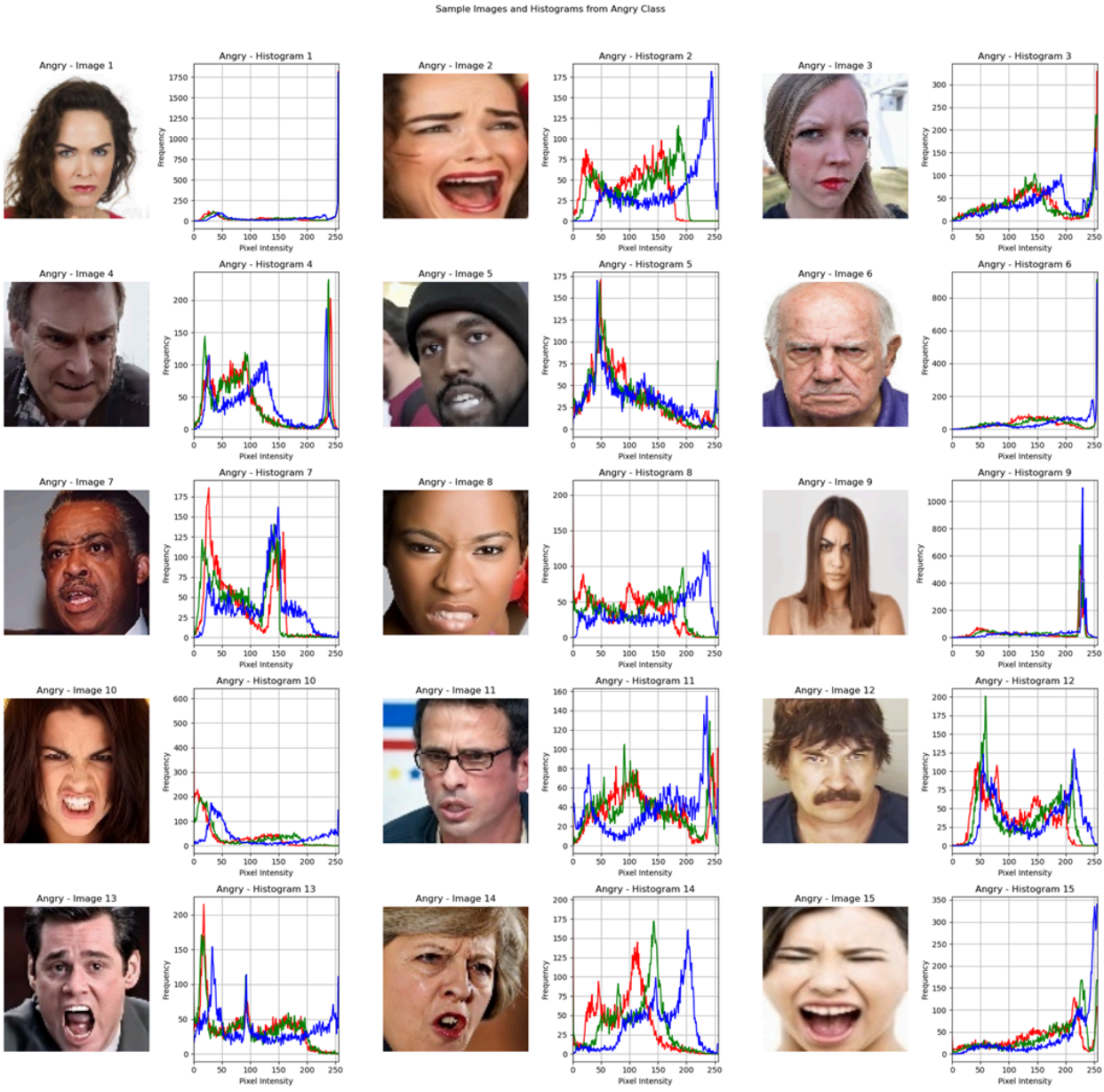

Figure 8: Collection of 15 Sample Images along its Pixel Intensity Histogram for the Angry Class

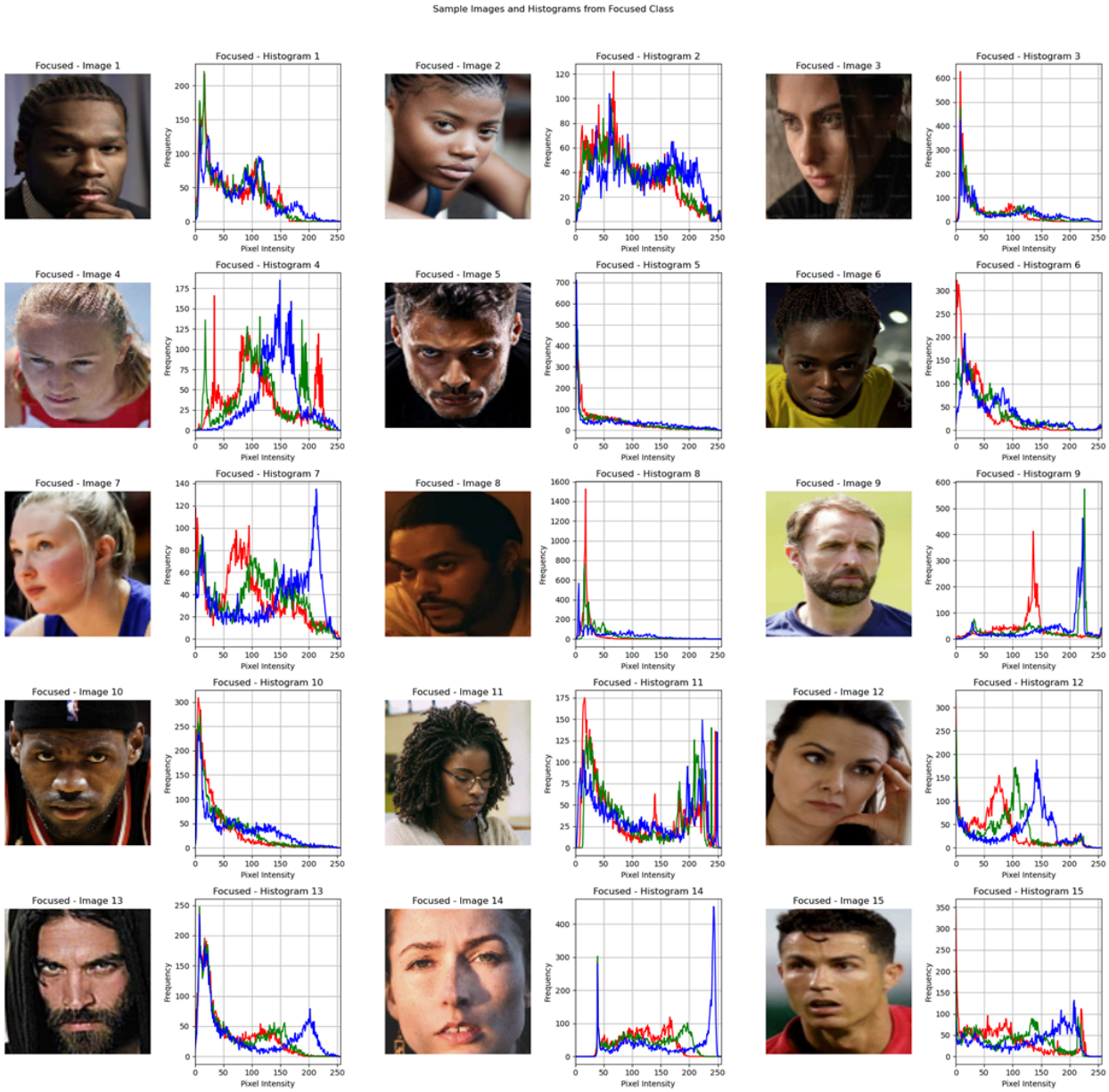

Figure 9: Collection of 15 Sample Images along its Pixel Intensity Histogram for the Focused Class

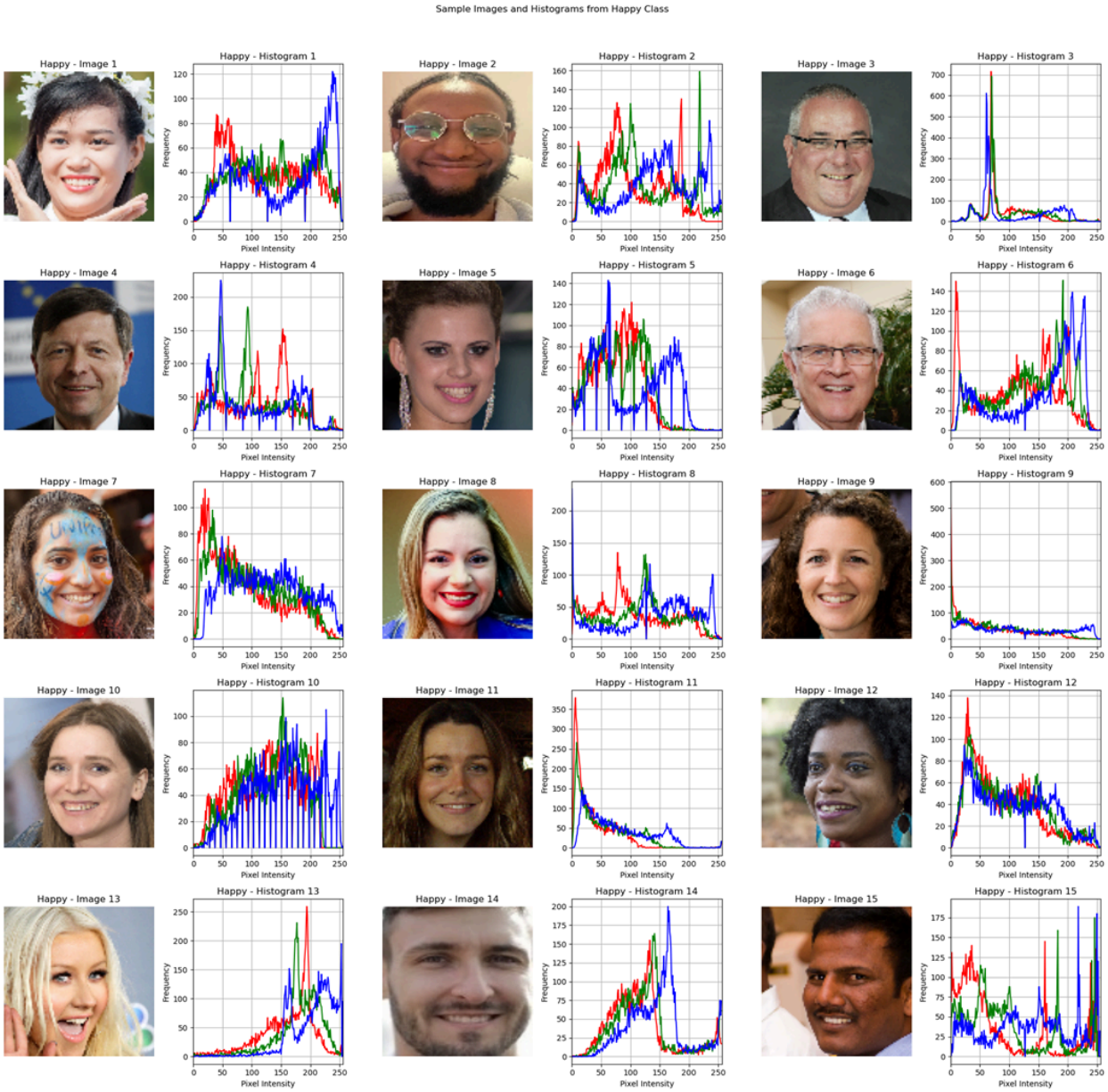

Figure 10: Collection of 15 Sample Images along its Pixel Intensity Histogram for the Happy Class

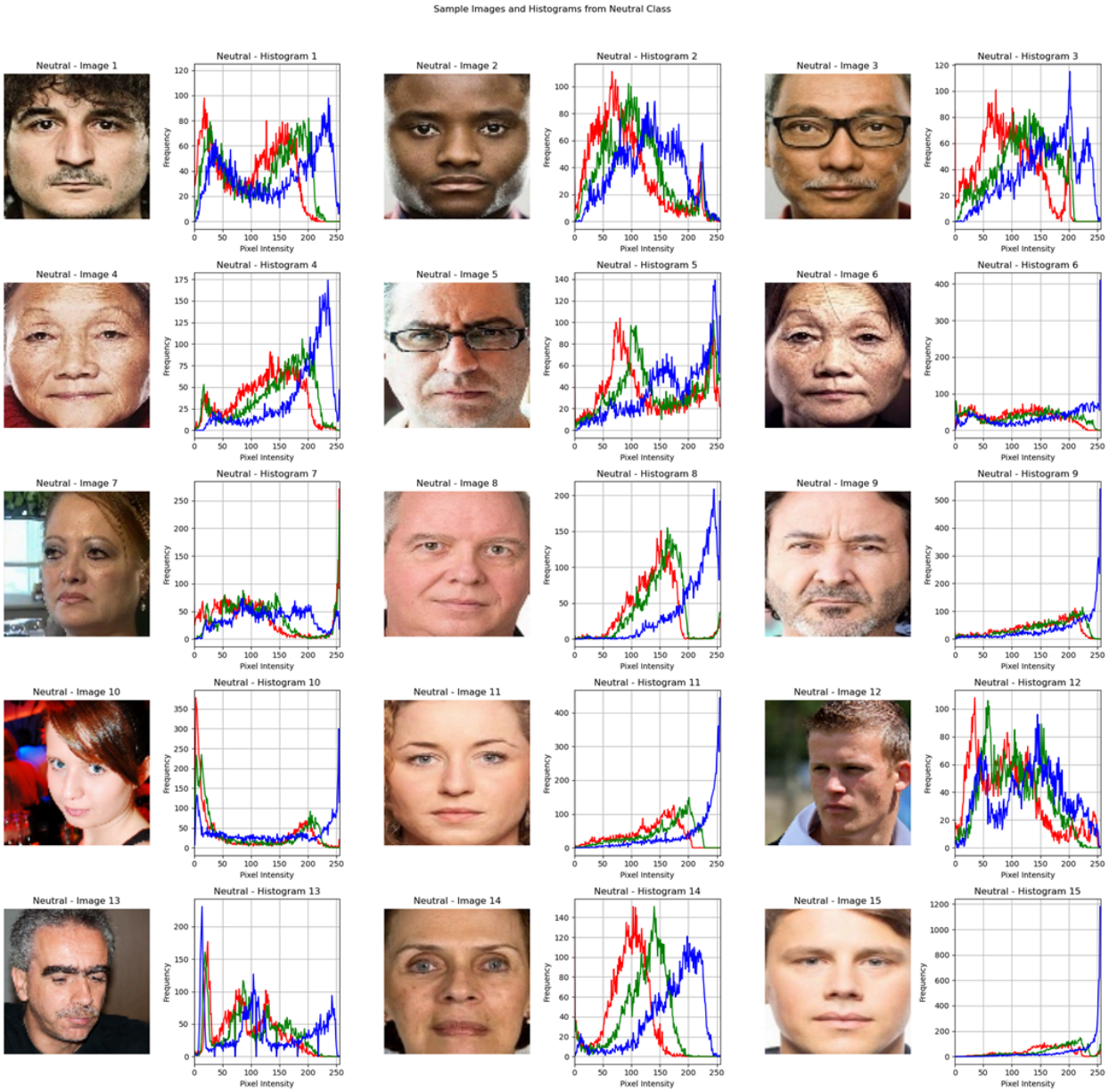

Figure 11: Collection of 15 Sample Images along its Pixel Intensity Histogram for the Neutral Class

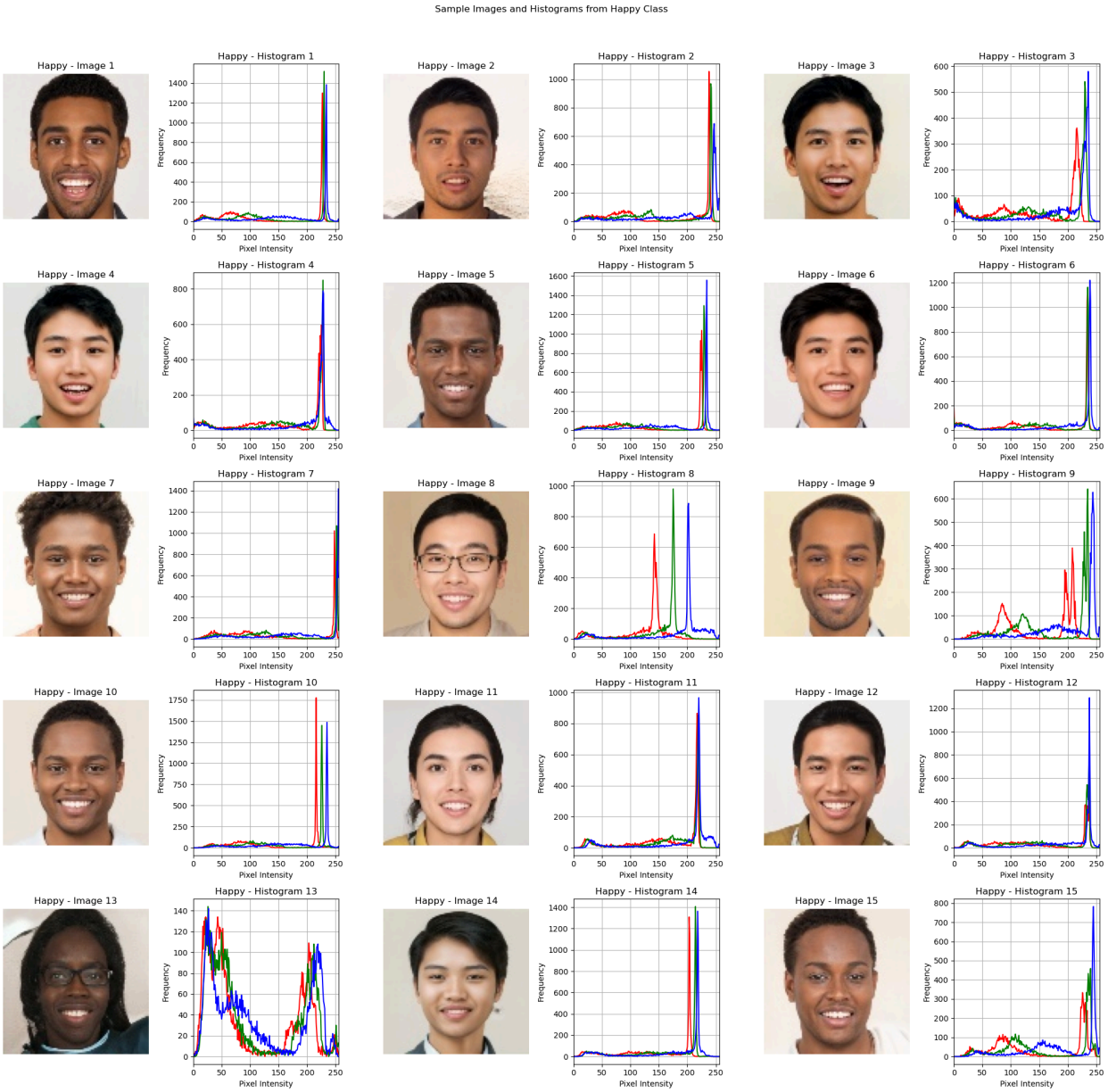

Figure 12: Collection of 15 Synthetic Sample of the Added Images for the Happy Class in Definitive Dataset

3. Data Training

3.1. CNN Architecture

This project utilizes PyTorch and torchvision for constructing the different Convolutional Neural Network (CNN) architectures for training our dataset. Scikit-learn is employed for splitting the dataset and for the evaluation phase of our three different models. The architectures of the models are implemented in three different Python scripts: main_model.py, model_variant1.py, and model_variant2.py. The training process is managed in a separate script, data_train.py and the models trained are evaluated in the run_model.py script for the whole dataset or for individual images. Two modules (dataset_utils.py and model_utils.py) were created to ease the coding of data_train.py and run_model.py. During the training, the best models are saved in a folder called 'saved_models' in the format of .pth files for the model weights and .json files for the training and validation metrics.

3.1.1. Main Model Overview: Leaky ReLU activation

The main model is structured with five convolutional layers, each followed by batch normalization and activation functions. The first two layers each consist of 32 filters with a kernel size of 3x3, a stride of 1, and padding of 1. These layers are followed by a max pooling layer that reduces the spatial dimensions from 96x96 to 48x48. The subsequent two layers each have 64 filters, maintaining the same kernel size, stride, and padding as the first layers, with another max pooling layer added after them, reducing the dimensions further to 24x24. The final convolutional layer has 128 filters, followed by a max pooling layer, which reduces the dimensions to 12x12. The primary activation function used in this model is Leaky ReLU, which prevents dead neurons by allowing a small, non-zero gradient when the unit is not active. The output of the final convolutional layer is passed through a dropout layer to prevent overfitting, followed by two fully connected layers. Batch normalization is applied after each convolutional layer to normalize the output and stabilize the learning process. A dropout layer with a probability of 0.3 is used before the fully connected layers to prevent overfitting by randomly setting a fraction of input units to zero at each update during training. These techniques help in improving the generalization capabilities of the model.

3.1.2. Variant 1 Overview: PReLU Activation and Architectural Adjustments

The first variant, MultiLayerFCNetVariant1, introduces several changes. This model replaces Leaky ReLU with PReLU (Parametric ReLU), allowing the negative slope to be learned during training, potentially improving the model's flexibility and performance. The architecture includes four convolutional layers: the first two layers have 32 and 64 filters respectively, followed by a max pooling layer that reduces the spatial dimensions from 96x96 to 48x48. The third layer, with 128 filters, uses a stride of 2 to downsample the feature maps, reducing the dimensions to 24x24. The final layer has 128 filters, followed by another max pooling layer, reducing the dimensions to 12x12. This structure aims to balance feature extraction and computational efficiency.

3.1.3. Variant 2: Sigmoid Activation and Kernel Size Modifications

The second variant, MultiLayerFCNetVariant2, uses Sigmoid activation functions throughout the model. This variant also experiments with different kernel sizes: the first layer uses a larger kernel size of 7x7, the second layer uses a kernel size of 5x5, and the final convolutional layer uses a smaller kernel size of 2x2. The spatial dimensions change accordingly: from 96x96 to 48x48 after the first max pooling layer, then to 24x24 after the second max pooling layer, and finally to 12x12 after the third max pooling layer. These changes aim to investigate the impact of different kernel sizes on the model's ability to capture spatial features.

3.2. Training Process

3.2.1. Data Splitting

The dataset, containing 2000 images, was split into training, validation, and test sets. The images were split as follows:

- **Training Set:** 1400 images (70%), ensuring each class is equally represented.
- **Validation Set:** 300 images (15%), ensuring a balanced representation for model tuning.
- **Test Set:** 300 images (15%), used for final model evaluation.

3.2.2. Training Methodology

The training process is meticulously designed to ensure robust learning and generalization. The model is trained for a maximum of 60 epochs with a learning rate set to 0.001, balancing the convergence speed and stability. Cross-entropy loss is used, suitable for multi-class classification problems. The training employs a batch size of 128 to ensure efficient training and stability.

3.2.3. Optimization Techniques

The Adam optimizer is employed due to its adaptive learning rate capabilities, which help in faster convergence and handling sparse gradients. Early stopping is implemented with a patience of 6 epochs, meaning the training stops if there is no improvement in validation loss for 6 consecutive epochs. This technique helps prevent overfitting by ensuring that the model does not continue training once it stops improving on the validation set. For instance, in the main model, the best model was found at epoch 11 with a training loss of 0.1004, training accuracy of 96.29%, and validation accuracy of 87.00%. The training did not complete all 60 epochs and stopped early at 17 epochs due to the early stopping criteria. Data augmentation was also applied to the training data to enhance the model's generalization capabilities. Techniques include random horizontal flipping and random rotations of up to 10 degrees. Additionally, L2 regularization (weight decay) with a coefficient of 1e-4 is applied to the Adam optimizer to prevent overfitting by penalizing large weights.

3.2.4. Training and Validation Monitoring

During training, the model's performance is monitored on both training and validation sets. Metrics such as accuracy and F1-Score are calculated for both sets to evaluate the model's performance. The model with the lowest validation loss is saved as the best model. Metrics for the best model, including training loss, training accuracy, training F1-score, validation loss, validation accuracy, and validation F1-score, are saved for analysis. This comprehensive monitoring ensures that the model not only learns effectively from the training data but also generalizes well to unseen data. For example, Variant 1 found its best model at epoch 22 with a training accuracy of 93.86% and a validation accuracy of 88.33%. Variant 2 found its best model at epoch 16 with a training accuracy of 93.93% and a validation accuracy of 85.67%.

3.3. Evaluation

After training, the best model is evaluated on the test set using various metrics. Accuracy measures the overall correctness of the model, while precision, recall, and F1-Score provide a detailed performance overview. Both macro and micro averages are calculated to understand the model's performance across different classes. Additionally, a confusion matrix is visualized to gain insights into the model's classification performance for each class.
This structured approach to training and evaluation ensures that the model is well-tuned for recognizing facial expressions, with a robust framework to generalize well in real-world scenarios.

3.3.1. Performance Metrics

Model	Macro			Micro			Accuracy
	P	R	F	P	R	F	
Main Model	86.73%	86.67%	86.67%	86.67%	86.67%	86.67%	86.67%
Variant 1	81.20%	80.76%	80.78%	80.67%	80.67%	80.67%	80.67%
Variant 2	84.15%	84.00%	83.99%	84.00%	84.00%	84.00%	84.00%

Table 2: Performance Metrics of the Three Models

3.3.1.1. Metrics Analysis

The main model demonstrates the best overall performance with a balanced set of metrics across the board. With an accuracy of 86.67%, it achieves the highest precision (86.73%), recall (86.67%), and F1-Score (86.67%) for both macro and micro averages. This balance suggests that the model is consistently reliable across all classes, making it the most robust choice for recognizing facial expressions. The high precision indicates that the model makes fewer false positives, which is crucial in applications where incorrect classification could lead to significant issues, such as misinterpreting emotions in a critical setting. The high recall signifies that the model successfully identifies a high proportion of actual positive instances, ensuring that most of the relevant facial expressions are correctly recognized.

Variant 1, while showing competitive performance, falls behind the main model with an accuracy of 80.67%. It has a macro precision of 81.20% and a macro recall of 80.76%, resulting in a macro F1-Score of 80.78%. The micro metrics are identical, all standing at 80.67%. This model’s lower precision and recall indicate a trade-off between identifying correct expressions and

minimizing false positives. The lower precision means that it has a higher rate of false positives compared to the main model, potentially misclassifying neutral expressions as emotional ones. However, its competitive recall suggests it is still quite effective in recognizing most relevant expressions. In contexts where it is critical to capture as many true expressions as possible, even at the cost of some false positives, Variant 1 could still be valuable.

Variant 2 strikes a middle ground with an accuracy of 84.00%, better than Variant 1 but not as good as the main model. It achieves a macro precision of 84.15%, a macro recall of 84.00%, and a macro F1-Score of 83.99%. Similarly, its micro precision, recall, and F1-Score are all 84.00%. This model's performance indicates a relatively balanced approach with slightly better precision and recall than Variant 1 but still trailing behind the main model. The balanced metrics suggest that Variant 2 is reasonably effective at both correctly identifying expressions and avoiding false positives, making it a good choice in applications where a moderate level of both precision and recall is acceptable. The use of Sigmoid activation functions and varied kernel sizes likely contribute to its improved ability to capture different features, leading to its enhanced performance compared to Variant 1.

Note: Micro precision and recall are the same across each model because they are calculated from the same underlying data (true positives, false positives, and false negatives) aggregated over all classes. Specifically, for a multi-class classification problem, these metrics focus on the overall counts of true positives, false positives, and false negatives, rather than on a per-class basis. Given that each class in our dataset contains 500 images, the distribution is balanced, which contributes to the similarity in these metrics.

3.3.1.2. Comparative Insights

The table highlights that the main model excels in all metrics, making it the best choice for applications requiring high accuracy and balanced precision and recall. Variant 1's lower precision but similar recall compared to Variant 2 implies it may be more prone to false positives, which could be a disadvantage in scenarios where misclassification carries significant consequences. Conversely, Variant 2, with its balanced performance, presents a good compromise between the two, offering better precision and recall than Variant 1 while still not matching the main model's robustness. In the context of facial image analysis, where accurately identifying expressions is crucial, the main model's higher precision ensures that fewer incorrect expressions are detected, reducing the risk of misinterpretation. Higher recall means it misses fewer actual expressions, which is vital in ensuring comprehensive emotion detection. Variant 1 might be suitable for applications where capturing as many expressions as possible is more important than minimizing false positives. Variant 2, with its balanced metrics, could be a versatile option where both precision and recall are moderately important, but neither can be sacrificed significantly.

3.3.2. Confusion Matrix Analysis

3.3.2.1. Main Model Confusion Matrix

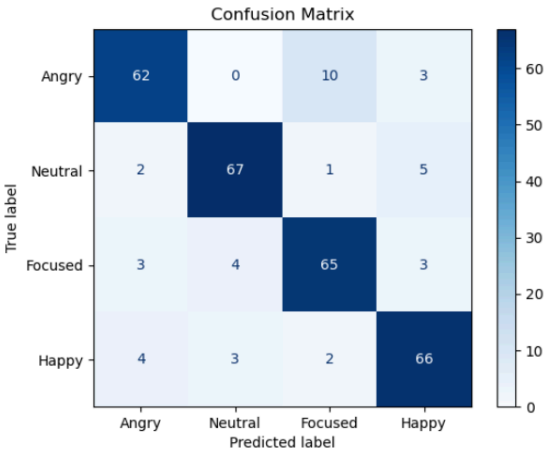

Figure 9: Confusion Matrix of the Main Model

The confusion matrix for the main model reveals several key points of performance. The model most frequently confuses 'Angry' and 'Focused' expressions, with 10 instances of 'Focused' being misclassified as 'Angry'. However, the model demonstrates a strong ability to distinguish 'Neutral' expressions, with 67 correctly classified instances and only minimal misclassifications: 2 as 'Angry', 1 as 'Focused', and 5 as 'Happy'. Notably, there are no misclassifications of 'Neutral' expressions as 'Angry'. Additionally, the model shows strong performance in recognizing 'Happy' expressions, with 66 correctly classified instances, and only 4 instances each misclassified as 'Angry' and 'Neutral', and 2 as 'Focused'. This indicates that 'Neutral' was the best classified class, followed closely by 'Happy'.

3.3.2.2. Variant 1 Confusion Matrix

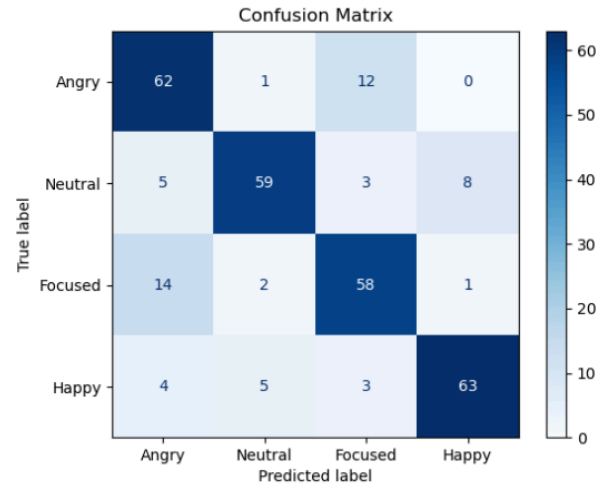

Figure 10: Confusion Matrix of Variant 1

Variant 1 demonstrates an improved balance in classification accuracy compared to Variant 1 but still follows the same confusion pattern between 'Angry' and 'Focused' expressions, with 12 instances of 'Focused' misclassified as 'Angry'. The model correctly classifies 'Neutral' expressions 62 times, with misclassifications distributed as 5 'Angry', 2 'Focused', and 6 'Happy'. This indicates a better performance than Variant 1 but still not as accurate as the main model. The 'Happy' expressions are correctly classified 62 times, with misclassifications of 5 'Angry', 6 'Neutral', and 2 'Focused'. This model shows an overall balanced performance, slightly improving upon Variant 1 in recognizing 'Neutral' expressions but still falling short of the main model's accuracy.

3.3.2.3. Variant 2 Confusion Matrix

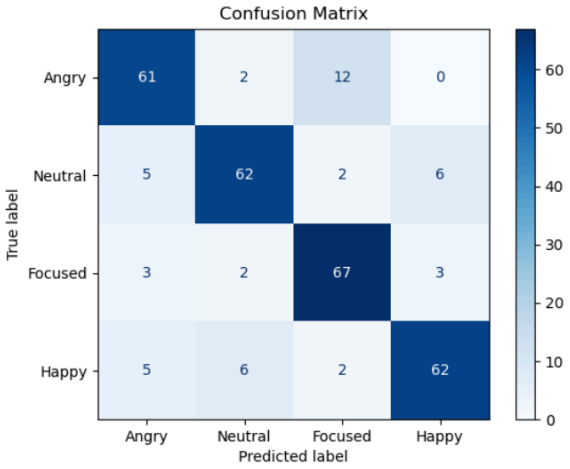

Figure 11:Confusion Matrix of Variant 2

Variant 2 demonstrates an improved balance in classification accuracy compared to Variant 1 but still follows the same confusion pattern between 'Angry' and 'Focused' expressions, with 12 instances of 'Focused' misclassified as 'Angry'. The model correctly classifies 'Neutral' expressions 62 times, with misclassifications distributed as 5 'Angry', 2 'Focused', and 6 'Happy'. This indicates a better performance than Variant 1 but still not as accurate as the main model. The 'Happy' expressions are correctly classified 62 times, with misclassifications of 5 'Angry', 6 'Neutral', and 2 'Focused'. This model shows an overall balanced performance, slightly improving upon Variant 1 in recognizing 'Neutral' expressions but still falling short of the main model's accuracy.

3.3.2.4. Speculation for Misclassification

The consistent misclassification of 'Angry' and 'Focused' expressions across all three models suggests a few possible underlying reasons:

- **Similar Features**: The facial features of 'Angry' and 'Focused' expressions are inherently similar, with common characteristics such as furrowed brows and intense gazes; when a person is focused, their eyes are typically fixed on the task at hand, and similarly, an angry person may also have an intense stare[14]. Facial tension is another common feature: when a person is focused, they might furrow their brows in concentration, and an angry person might also furrow their brows, clench their jaws, or tighten other facial muscles[15]. This similarity is particularly pronounced in scenarios where individuals are deeply concentrating, such as during sports activities or when wearing glasses, making it challenging for the models to distinguish between the two[16].
- **Dataset Insufficiency**: There may not be enough images in the dataset to capture the variability within these classes caused by the similar features, leading to insufficient learning.
- **Mislabeling**: Although we tried to pick the most expressive images from both datasets (FANE and AffectNet) and the ones online, we can’t exclude that some images might have been incorrectly labeled, causing the models to learn incorrect associations between features and expressions.

3.3.2.5. Well-Recognized Classes

While 'Angry' and 'Focused' expressions pose a challenge, classes such as 'Neutral' and 'Happy' are well-recognized by all models. For example, the main model correctly classified 67 out of 75 'Neutral' instances and 66 out of 75 'Happy' instances. The 'Neutral' expression does not exhibit expressive facial features like other expressions, making it easier for the model to recognize. Key features of a neutral expression include unfocused eyes that hold a steady gaze, relaxed eyebrows that are not raised or furrowed, slack cheeks that are neither raised nor tensed, and a straight-lined mouth that remains closed or slightly open in a neutral position[17]. Similarly, 'Happy' expressions are well-recognized due to their distinct and consistent features. A happy expression typically includes slightly squinting eyes, wrinkles appearing at the corners of the eyes, raised cheeks, smile lines, mouth corners moving up at a diagonal, and often exposing teeth[18]. These well-defined characteristics make 'Happy' expressions easier for the models to classify correctly. The success in recognizing these expressions can be attributed to the clearer distinctions in facial features, which the models can effectively learn and generalize.

3.3.3. Impact of Architectural Variations

3.3.3.1. Depth and Performance

The depth of the convolutional layers had a notable impact on the performance of the models. The main model, with its five convolutional layers, struck an optimal balance between capturing detailed features and avoiding overfitting. This depth allowed the model to extract a rich hierarchy of features from the input images, contributing to its superior performance in recognizing facial expressions. In contrast, Variant 1, which had fewer layers, showed a decline in performance. The reduced depth likely hindered the model's ability to capture intricate details, leading to a higher rate of misclassifications. On the other hand, Variant 2, which added additional layers, demonstrated better performance than Variant 1 but still did not surpass the main model. This suggests that while increasing depth can enhance feature extraction, there is a threshold beyond which additional layers may lead to diminishing returns or even overfitting, where the model learns the training data too well but fails to generalize to unseen data.

3.3.3.2. Kernel Size and Feature Recognition

The variation in kernel sizes across the models also had a significant impact on their recognition abilities. The main model utilized a combination of smaller kernel sizes (3x3), which are effective in capturing fine details and subtle features in facial expressions. This contributed to its high precision and recall, as the smaller kernels were able to detect minute variations in the facial features that distinguish different expressions. Variant 1, which maintained similar kernel sizes but with fewer layers, struggled to achieve the same level of detail, resulting in lower performance. Conversely, Variant 2 experimented with larger kernel sizes (e.g., 5x5), which helped in recognizing broader features but sometimes missed finer details. This variation allowed Variant 2 to perform better than Variant 1 by capturing a wider context of facial features, although it still fell short of the main model's performance. The choice of kernel sizes demonstrated a trade-off: larger kernels can capture more contextual information, which is beneficial for broader feature recognition, while smaller kernels excel at identifying finer details essential for distinguishing subtle facial expressions.

3.3.4. Confusion Matrix Analysis of the Definitive Model

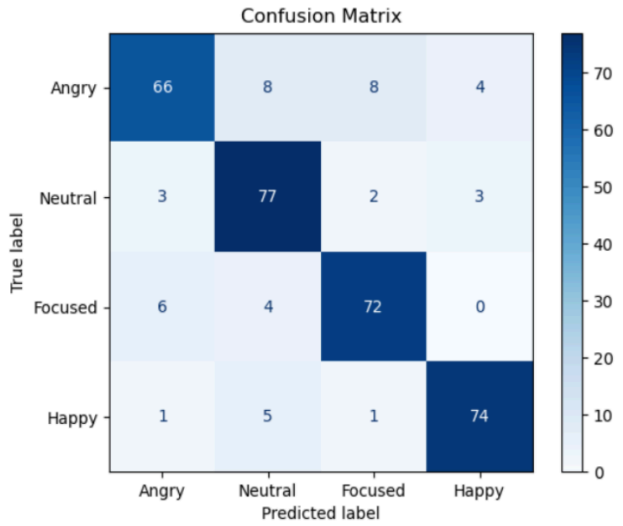

Figure 12: Confusion Matrix of Definitive Model

The confusion matrix for the definitive model highlights a notable improvement in classification accuracy across most classes. The model uses the architecture of the main model but with the augmented Definitive dataset (bias-mitigated dataset). It effectively distinguishes 'Happy' expressions, with 74 correctly classified instances and minimal misclassifications: 5 as 'Neutral', 1 as 'Angry', and 1 as 'Focused'. 'Neutral' expressions are also well-recognized, with 77 correct classifications and a few errors: 3 as 'Angry', 2 as 'Focused', and 3 as 'Happy'. There is a slight confusion between 'Angry' and 'Focused' expressions, with 8 instances of 'Angry' misclassified as 'Focused' and 8 instances of 'Focused' misclassified as 'Angry'. However, the majority of 'Angry' expressions are correctly identified, with 66 correctly classified instances. The model shows strong performance in recognizing 'Focused' expressions, with 72 correct classifications and only 4 instances misclassified as 'Neutral'. This indicates that 'Neutral' and 'Happy' are the best classified classes, followed by 'Angry' and 'Focused'.

3.3.5. K-fold cross-validation

We used ‘KFold’ from scikit-learn to split the dataset into 10 folds. This replaces the single train test split from Part 1. For each fold, the test set is one of the 10 folds, and the remaining 9 folds are combined and then split into training (85%) and validation (15%) sets. This adapts the original train/test/validation split to work within each fold. We maintained the early stopping mechanism from Part 2 with the patience parameter set to 6 epochs. Results from all folds are collected and stored, allowing for analysis across the folds.

Two tables given below shows the results from the 10-fold cross-validation (one table for your final model from Part II (main version), one for the final model)

Fold	Macro			Micro			Accuracy (%)
	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	
1	83.49	83.12	83.21	83.50	83.50	83.50	83.50
2	83.63	84.24	83.66	84.00	84.00	84.00	84.00
3	84.55	84.62	84.57	84.50	84.50	84.50	84.50
4	84.74	83.76	84.08	84.00	84.00	84.00	84.00
5	85.45	85.16	85.12	85.00	85.00	85.00	85.00
6	73.55	75.78	73.65	73.50	73.50	73.50	73.50
7	85.48	85.95	85.44	86.00	86.00	86.00	86.00
8	85.88	85.98	85.81	86.00	86.00	86.00	86.00
9	87.35	87.60	87.26	88.00	88.00	88.00	88.00
10	83.63	83.33	83.38	83.50	83.50	83.50	83.50
Average	83.77	83.95	83.62	83.80	83.80	83.80	83.80

Table 3: 10-Fold Cross-Validation Performance Metrics of the Main Model

Fold	Macro			Micro			Accuracy (%)
	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	
1	86.67	85.65	86.06	85.65	85.65	85.65	85.65
2	82.83	82.75	82.37	82.43	82.43	82.43	82.43
3	83.80	83.61	83.64	83.78	83.78	83.78	83.78
4	84.51	84.63	84.52	84.68	84.68	84.68	84.68
5	85.39	84.69	84.43	85.14	85.14	85.14	85.14
6	83.14	83.31	83.21	82.43	82.43	82.43	82.43
7	83.53	84.22	83.35	83.78	83.78	83.78	83.78
8	84.65	84.00	84.22	84.68	84.68	84.68	84.68
9	83.53	84.22	83.34	83.78	83.78	83.78	83.78
10	82.24	82.29	82.23	82.88	82.88	82.88	82.88
Average	83.75	83.72	83.66	83.75	83.75	83.75	83.75

Table 4: 10-Fold Cross-Validation Performance Metrics of the Definitive Model

Comparison of Performance of Main Model and the Definitive Model:

Both models show similar average performance, with the Main model slightly outperforming the Definitive model. The performance metrics (Precision, Recall, and F1-score) are very consistent across macro and micro averages for both models. The Definitive model shows more consistency across folds, with accuracy ranging from 82.43% to 85.65%. Whereas, the main model has a wider range of performance , with accuracy ranging from 73.50% to 88%. Main model has a notable outlier in Fold 6, with significantly lower performance (73.50% accuracy) compared to other folds. Definitive models don't show any extreme outliers , suggesting more stable performance across different data subsets. Main model achieves the highest single fold performance with 88% accuracy in fold 9. Definitive model's best performance is 85.65% accuracy in Fold 1.

In conclusion, while the main model archives slightly higher average performance, the definitive model demonstrates more consistent performance across folds.

Comparison of the performance of the main model with single test/train split evaluation and k-fold cross validation:

The average performance from k-fold cross validation is similar to single train/test splits results. The accuracy is nearly identical (83.85% vs 83.80%), suggesting that the model's overall performance is consistent. In both cases, the macro and micro metrics are close, but the k-fold results show slightly more variation between macro and micro metrics, suggesting that performance may vary more across classes in different data subsets. The Part 2 results show slightly higher precision than recall (macro). While the k-fold results, recall is slightly higher than precision. This suggests that the model's balance between precision and recall can vary depending on the specific data split. The similarity between part 2 and average k-fold results suggests that the original train/test split was representative of the overall dataset. However, the variation across folds (particularly the outlier in Fold 6) indicates that the model's performance can be sensitive to the specific data it's trained on and evaluated on.

In conclusion, the k-fold method reveals important insights about that model's stability and generalizability. It highlights that the model's performance can vary significantly depending on the specific data it encounters, as apparent from the single split evaluation.

4. Bias Analysis

4.1. Introduction

In this analysis, we examined the bias attributes of **race** and **gender** within our dataset. Our approach involved splitting the original dataset into separate folders based on these group characteristics. For instance, to create the female dataset, we manually moved all images of females from the original dataset into the corresponding classes within the female dataset. The same process was applied to create the male dataset. Similarly, we sorted the images to create separate datasets for Asian, Black, and White individuals. To mitigate bias, we removed images that were not expressive enough and added new images to the Asian, Black, and Male datasets (see figure 1). This resulted in three new datasets: Black Dataset 2.0, Asian Dataset 2.0, and Male Dataset 2.0. We used a script called bias_dataset_label.py to label all images in the bias datasets (Male, Male 2.0, Female, Asian, Asian 2.0, White, Black, Black 2.0) in the format Expression_Race_Index or Expression_Gender_Index. For the definitive dataset, we combined the bias-mitigated datasets: Asian 2.0, Black 2.0, White, Female, and Male 2.0. We used a script called definitive_dataset_label.py to merge these datasets into one and rename the images in the

format Expression_Race_Gender_Index. This comprehensive labeling approach allowed for detailed tracking of each image's expression, race, and gender, facilitating our bias detection and analysis efforts.

4.2. Bias Mitigation Results

Attribute	Group	#Images	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Race	Asian	341	69.23	69.80	69.73	69.74
	Black	536	70.37	67.79	66.79	67.07
	White	1122	81.07	81.34	82.46	81.57
	Total/Average	2000	73.56	72.98	72.99	72.79
Gender	Male	985	73.65	73.38	73.58	73.00
	Female	1015	86.27	86.64	86.70	86.44
	Total/Average	2000	79.96	80.01	80.14	79.72
Overall System Total/Average		2000	86.67	86.73	86.67	86.67

Table 5: Bias Analysis Results of the Main Model Across Different Demographic Group

The bias analysis conducted on the main model revealed disparities in model performance across different demographic groups, as shown in Table 2. The analysis indicated that the model performed significantly better on the White group, with an accuracy of 81.07% and an F1-score of 81.57%, compared to the Black group, which had an accuracy of 70.37% and an F1-score of 67.07%, and the Asian group, which had the lowest performance with an accuracy of 69.23% and an F1-score of 69.74%. Similarly, for the gender attribute, the model performed better on the Female group (accuracy of 86.27% and F1-score of 86.44%) compared to the Male group (accuracy of 73.65% and F1-score of 73.00%). These disparities highlight the need for bias mitigation to ensure more equitable model performance across different demographic groups. One factor that could have contributed to these disparities is the number of images in each group. The White group had the highest number of images (1122), which may have led to better model performance due to a more extensive and diverse dataset. In contrast, the Asian group had the fewest images (341), which might have resulted in lower performance due to limited data representation. Similarly, the Female group had slightly more images (1015) than the Male group (985), potentially contributing to the observed performance differences. Addressing these imbalances is crucial for mitigating bias and improving model fairness.

4.3. Bias Mitigation Steps

To mitigate bias in our model, we took several steps to enhance the dataset and improve the model's performance for underrepresented groups. We focused on adding images only for the underperforming groups (male, Asian, Black). Specifically, we added 249 images in total, ensuring we balanced the classes as much as possible. Here's how we decided on the addition of images:

- Angry Asian male: +50
- Angry Black male: +32
- Focused Asian male: +50
- Happy Asian male: +14
- Happy Black male: +30
- Neutral Asian male: +48
- Neutral Black male: +25

These images were added to both the male 2.0 dataset and the corresponding race-based datasets (Asian 2.0 and Black 2.0). Additionally, we deleted 28 images that weren't expressive enough from the old pictures (not the newly added ones) in the Black, Asian, and male datasets. This process helped create a more balanced set of classes in the different groups, providing more training data to the model to enhance its performance and mitigate bias.

Having a more balanced set of classes and increasing the number of images in the dataset is crucial for a few reasons. First, a balanced dataset helps the model learn equally from all classes, preventing it from being biased towards classes with more examples. This balance is particularly important for underrepresented groups to improve the model's fairness and accuracy across different demographic groups. Second, having more images generally improves the model's ability to generalize and perform well on unseen data by providing it with a richer and more varied set of training examples. By adding images to the underperforming groups and deleting non-expressive images from the Black, Asian, and male datasets, we aimed to create a more robust and equitable model that can better understand and classify facial expressions across diverse populations.

4.4. Comparative Performance Analysis

After retraining the model with the bias mitigation steps, we observed a notable improvement in the performance metrics across different demographic groups. The table below shows the results for the re-trained models:

Attribute	Group	#Images	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Race	Asian	491	82.43	81.99	81.33	81.34
	Black	584	82.95	82.05	82.25	81.91
	White	1122	81.07	81.34	82.46	81.57
	Total/Average	2197	82.15	81.79	82.01	81.61
Gender	Male	1182	84.83	85.10	84.85	84.74
	Female	1015	86.27	86.64	86.70	86.44
	Total/Average	2197	85.55	85.87	85.78	85.59
Overall System Total/Average		2221	86.53	86.72	86.62	86.54

Table 6: Performance Metrics of the Re-trained Model after Bias Mitigation

Comparing these results with the original model's performance (main_model) where we had an overall accuracy of 86.67%, we can see that the accuracy slightly decreased to 86.53% with the definitive model. However, this decrease is insignificant and does not overshadow the key improvement in the balanced performance across different demographic groups. For example:

- The accuracy for the Asian group improved from 69.23% to 82.43%.
- The accuracy for the Black group increased from 70.37% to 82.95%.
- The accuracy for the Male group rose from 73.65% to 84.83%.

These improvements indicate that our bias mitigation efforts successfully enhanced the model's fairness and reliability. Although the overall system accuracy slightly decreased, the more balanced performance across diverse groups signifies a significant step toward reducing bias. This improvement highlights the importance of having a more balanced dataset and adequately representing underrepresented groups, which leads to a more equitable and robust model.

Moreover, the previously underperforming groups (Asian, Black, and Male) now show performance metrics that are comparable to the other groups (White and Female). The mitigated

bias not only improved the performance for these groups but also brought their metrics closer to those of the better-performing groups, thereby achieving a more consistent and fair model performance across all demographics.

5. Conclusion

In conclusion, the meticulous process of data cleaning, labeling, and merging was pivotal in creating a comprehensive and balanced dataset for our CNN model. By combining the FANE and AffectNet datasets, and supplementing them with manually collected images for the Focused expression, we ensured a diverse and representative dataset. Addressing challenges such as image size inconsistencies, duplicate images, and varying lighting conditions through normalization and standardization techniques, we enhanced the dataset's quality and uniformity.

For the process of data training, the main model out performed both variants, achieving the highest overall accuracy and balanced metrics, particularly excelling in recognizing 'Neutral' and 'Happy' expressions. Its superior performance is attributed to the optimal depth of convolutional layers and the effective use of smaller kernel sizes that capture fine details essential for distinguishing subtle facial features. Variant 1, with fewer layers, struggled to capture intricate details, resulting in lower accuracy and higher misclassification rates. Variant 2, despite incorporating larger kernel sizes and additional layers, did not surpass the main model's performance, indicating that a balanced approach is crucial.

Finally, for the process of evaluating the model, the implementation of bias mitigation steps significantly improved model fairness across different demographic groups. By adding new images to the underrepresented and underperforming groups, such as Asian, Black, and male datasets, and ensuring a more balanced class distribution, we addressed the disparities observed in initial model evaluations. The final model demonstrated enhanced performance and reduced bias, as evidenced by more consistent accuracy, precision, recall, and F1 scores across all demographic groups.

6. Reference

[1] Z. A. N. Nabil. “FANE Facial Expressions and Emotion Dataset”, Kaggle. May 24, 2024. https://www.kaggle.com/datasets/furcifer/fane-facial-expressions-and-emotion-dataset

[2] Noam Segal. “AffectNet Training Data", Kaggle. May 24, 2024. https://www.kaggle.com/datasets/noamsegal/affectnet-training-data

[3] E. Landman. “Finding Duplicate Images with Python” towardsdatascience, June 4,2021. https://towardsdatascience.com/finding-duplicate-images-with-python-71c04ec8051

[4] LearnOpenCV. “Image Resizing with OpenCV”, May. 27, 2024. https://learnopencv.com/image-resizing-with-opencv/

[5] R. Gandhi, “Normalize an Image in OpenCV Python”, May 7, 2024. https://www.geeksforgeeks.org/normalize-an-image-in-opencv-python/

[6] J. Gavande, “Bar Plot in Matplotlib” geeksforgeeks, Mar 4, 2021. https://www.geeksforgeeks.org/bar-plot-in-matplotlib/

[7] J. Gavande “Plotting Histogram in Python using Matplotlib” geeksforgeeks, Jan 9, 2024. https://www.geeksforgeeks.org/plotting-histogram-in-python-using-matplotlib/

[5] sanjoy_62, “numpy.concatenate() function | Python” geeksforgeeks, Apr. 22, 2020. https://www.geeksforgeeks.org/numpy-concatenate-function-python/

[6] Microsoft. 2024. *VisualStudio Code Copilot* (version 1.89) [Large language model]. https://code.visualstudio.com/docs/copilot/overview

[7] R. Witté. “Lab Exercise #05: Artificial Neural Networks”, COMP 472. June 10 2024.

[8] R. Witté. “Lab Exercise #06: Introduction to Deep Learning”, COMP 472. June 10 2024.

[9] R. Witté. “Lab Exercise #07: Convolutional Neural Networks (CNNs)”, COMP 472. June 10 2024.

[10] N.S. Aakash. “Image Classification with Convolutional Neural Networks” FreeCodeCamp, Dec. 12 2020. https://www.youtube.com/watch?v=d9QHNkD_Pos

[11] S. Chilamkurthy. “Writing Custom Datasets, DataLoaders and Transforms” Pytorch, 2024. https://pytorch.org/tutorials/beginner/data_loading_tutorial.html#writing-custom-datasets-dataloaders-and-transforms

[12] “sklearn.metrics”, SciKit Learn, 2024. https://scikit-learn.org/stable/api/sklearn.metrics.html

[13] “Datasets & DataLoaders” Pytorch, 2024.
https://pytorch.org/tutorials/beginner/basics/data_tutorial.html#loading-a-dataset

[14] Hope. “10 Common Facial Expressions Explained” ListVerse, Jul. 5 2013.
https://listverse.com/2013/07/05/ten-compelling-origins-of-our-facial-expressions/

[15] H. Parvez. “What the angry facial expression looks like” PsychMechanics, Oct. 12 2020.
https://www.psychmechanics.com/facial-expressions-anger/

[16] A. Cuncic. “How to Read Facial Expressions” Verywellmind, Mar. 28 2023.
https://www.verywellmind.com/understanding-emotions-through-facial-expressions-3024851

[17] C. Collins. “Are Neutral Faces Really Neutral?” Aps, Jul. 29 2016.
https://www.psychologicalscience.org/observer/are-neutral-faces-really-neutral

[18] Jules. “How to Describe Facial Expressions in Writing”,
All Write Alright, 2024. https://allwritealright.com/how-to-describe-facial-expressions-in-writing/

[19] “Unique real-time face generator” Generated.Photos, Jun. 2024.
https://generated.photos/face-generator

Appendix

1. Focused Class Image Batch

Focused Dark-Skinned Men	https://stock.adobe.com/ca/search?filters%5Bcontent_type%3Aphoto%5D=1&filters%5Bcontent_type%3Aillustration%5D=1&filters%5Bcontent_type%3Azip_vector%5D=1&filters%5Bcontent_type%3Avideo%5D=1&filters%5Bcontent_type%3Atemplate%5D=1&filters%5Bcontent_type%3A3d%5D=1&filters%5Bcontent_type%3Aimage%5D=1&k=concentrated+woman&order=relevance&safe_search=1&limit=100&search_page=1&search_type=usertyped&acp=&aco=concentrated+woman&get_facets=0 https://www.istockphoto.com/search/2/image?alloweduse=availableforalluses&istockcollection=&mediatype=photography&phrase=Concentrated%20black%20man&sort=best https://stock.adobe.com/ca/search?filters%5Bcontent_type%3Aphoto%5D=1&filters%5Bcontent_type%3Aillustration%5D=1&filters%5Bcontent_type%3Azip_vector%5D=1&filters%5Bcontent_type%3Avideo%5D=1&filters%5Bcontent_type%3Atemplate%5D=1&filters%5Bcontent_type%3A3d%5D=1&filters%5Bcontent_type%3Aimage%5D=1&k=concentrated+black+man&order=relevance&safe_search=1&limit=100&search_page=1&search_type=usertyped&acp=&aco=concentrated+black+man&get_facets=0
Focused Dark-Skinned Women	https://www.google.com/search?q=Focused+black+woman&sca_esv=8935ef200ca57f75&sca_upv=1&rlz=1C5CHFA_enCA1107CA1109&udm=2&biw=1440&bih=721&sxsrf=ADLYWILYrhSFN9ljYaY2U6ZtGqarzrhNfw%3A1717201318892&ei=pmlaZsWLNoio5NoP_emGkAs&ved=0ahUKEwjF9vq0kbmGAxUIFFkFHf20AbIQ4dUDCBA&uact=5&oq=Focused+black+woman&gs_lp=Egxnd3Mtd2l6LXNlcnAiE0ZvY3VzZWQgYmxhY2sgd29tYW4yBBAjGCdItgpQigNYvAhwAXgAkAEBmAGlAaABzwSqAQM0LjK4AQPIAQD4AQGYAgagAsADwgIFEAAYgATCAgYQABgFGB6YAwCIBgGSBwM1LjGgB8UQ&sclient=gws-wiz-serp https://www.istockphoto.com/search/2/image?alloweduse=availableforalluses&istockcollection=&mediatype=photography&phrase=Concentrated%20black%20woman&sort=best https://stock.adobe.com/ca/search?filters%5Bcontent_type%3Aphoto%5D=1&filters%5Bcontent_type%3Aillustration%5D=1&filters%5Bcontent_type%3Azip_vector%5D=1&filters%5Bcontent_type%3Avideo%5D=1&filters%5Bcontent_type%3Atemplate%5D=1&filters%5Bcontent_type%3A3d%5D=1&filters%5Bcontent_type%3Aimage%5D=1&k=concentrated+black+woman&order=relevance&safe_search=1&search_page=1&search_type=usertyped&acp=&aco=concentrated+black+woman&get_facets=0

Focused Light-Skinned Men	https://www.google.com/search?q=Focused+white+man&sca_esv=8935ef200ca57f75&sca_upv=1&rlz=1C5CHFA_enCA1107CA1109&udm=2&biw=1440&bih=721&sxsrf=ADLYWIL7W2H2Qzalh6Cgtb3EgQ5hueCkLA%3A1717203991996&ei=F3RaZvO6PJSq5NoPjrS2iAU&ved=0ahUKEwizosyvm7mGAxUUFVkFHQ6aDVEQ4dUDCBA&uact=5&oq=Focused+white+man&gs_lp=Egxnd3Mtd2l6LXNlcnAiEUZvY3VzZWQgd2hpdGUgbWFuSPoLUNkEWJIJcAF4AJABAJgBa6ABqwSqAQM1LjG4AQPIAQD4AQGYAgOgAtsBwgIKEAAYgAQYQxiKBcICBRAAGIAEwgIGEAAYBxgewgIGEAAYCBgewgIIEAAYBxgIGB6YAwCIBgGSBwMxLjKgB_II&sclient=gws-wiz-serp https://www.istockphoto.com/search/2/image?alloweduse=availableforalluses&istockcollection=&mediatype=photography&phrase=Concentrated%20man&sort=best https://stock.adobe.com/ca/search?filters%5Bcontent_type%3Aphoto%5D=1&filters%5Bcontent_type%3Aillustration%5D=1&filters%5Bcontent_type%3Azip_vector%5D=1&filters%5Bcontent_type%3Avideo%5D=1&filters%5Bcontent_type%3Atemplate%5D=1&filters%5Bcontent_type%3A3d%5D=1&filters%5Bcontent_type%3Aimage%5D=1&k=concentrated+man&order=relevance&safe_search=1&limit=100&search_page=1&search_type=usertyped&acp=&aco=concentrated+man&get_facets=0
Focused Light-Skinned Women	https://www.google.com/search?q=Focused++woman&sca_esv=8935ef200ca57f75&sca_upv=1&rlz=1C5CHFA_enCA1107CA1109&udm=2&biw=1440&bih=721&sxsrf=ADLYWIIOd5Ok61nkPYtFPVjjkzPEyfC3Bw%3A1717201389281&ei=7WlaZpXZEMmg5NoP58ajyAo&ved=0ahUKEwjVg8PWkbmGAxVJEFkFHWfjCKkQ4dUDCBA&uact=5&oq=Focused++woman&gs_lp=Egxnd3Mtd2l6LXNlcnAiDkZvY3VzZWQgIHdvbWFuMgUQABiABDIFEAAYgAQyBRAAGIAEMgYQABgHGB4yBhAAGAUYHjIGEAAYBRgeMgYQABgIGB4yBhAAGAgYHjIGEAAYCBgeMgYQABgIGB5IpQZQ5AFY5AFwAXgAkAEAmAGGAaABhgGqAQMwLjG4AQPIAQD4AQGYAgGgAo8BmAMAiAYBkgcDMC4xoAf7Bw&sclient=gws-wiz-serp https://www.istockphoto.com/search/2/image?alloweduse=availableforalluses&istockcollection=&mediatype=photography&phrase=Concentrated%20woman&sort=best https://stock.adobe.com/ca/search?filters%5Bcontent_type%3Aphoto%5D=1&filters%5Bcontent_type%3Aillustration%5D=1&filters%5Bcontent_type%3Azip_vector%5D=1&filters%5Bcontent_type%3Avideo%5D=1&filters%5Bcontent_type%3Atemplate%5D=1&filters%5Bcontent_type%3A3d%5D=1&filters%5Bcontent_type%3Aimage%5D=1&k=concentrated+woman&order=relevance&safe_search=1&limit=100&search_page=1&search_type=usertyped&acp=&aco=concentrated+woman&get_facets=0