COMP 472
Artificial Intelligence Project Assignment
Part 1 & 2

Team name: **AK_18**

Data Specialist: **Nadim Khalife** (Student ID: 40188245)
Training Specialist: **Jungsoo Lee** (Student ID: 40174025)
Evaluation Specialist**: Victor-Thyreth Ouy** (Student ID: 40208821)

Project Repository: https://github.com/jungsoolee1/COMP-472-Project

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

Signatures and ID Numbers:

Jungsoolee Date: 2024-06-15
Jungsoo Lee 40174025

Date: 2024-06-15
Victor-Thyreth Ouy

Date: 2024-06-15
Nadim Khalife

I. Dataset

A. Overview

The dataset used for the artificial intelligence project is extracted from the FER 2013 (Facial Expression Recognition 2013) dataset. Here are the key details:

- **Total Number of Images**: The FER 2013 dataset contains 35,887 grayscale images of faces.
- **Selected Images**:
  - Approximately 400 images per class for training the model.
  - 100-200 images per class for testing the model.
- **Classes Used**: Neutral, angry, engaged, and happy.
- **Image Size**: All selected images are 48 x 48 pixels.
- **Diversity**:
  - Includes people of different ages, backgrounds, and ethnicities.
  - Ensures the dataset is robust and unbiased, working well across various backgrounds.
- **Image Characteristics**:
  - Most images are frontal face shots, suitable for training models focused on frontal face analysis.
  - Some images have unorthodox framing and positions, beneficial for testing to determine which angles the program evaluates best.

B. Justification for Dataset Choice

- Abundant Facial Expressions: The dataset contains a substantial number of images of various facial expressions, allowing us to train the model effectively.
- Consistent Image Size: The images are uniformly sized at 48x48 pixels, simplifying the data preparation process.
- Aligned Emotions: The emotions depicted in the images (neutral, angry, engaged, and happy) align well with our project's goal of detecting students' emotions.
- Diverse Subjects: The dataset includes people of different ages, backgrounds, and ethnicities, making the model more robust and unbiased.

C. Dataset Challenges

- Resolution Limitations: The 48x48 pixel resolution might miss finer details in facial expressions. This limitation can affect the model's ability to capture subtle changes in facial features that are critical for distinguishing between similar emotions.
- Similar Expressions: Differentiating between neutral and engaged expressions is challenging due to their similarities. Both expressions often appear similar, with

minimal differences in facial features, making it difficult to categorize them accurately.

- Detail Loss: The fixed resolution may limit the detail in facial expressions, potentially impacting the model's training effectiveness. Higher resolution images could capture more intricate details that are essential for precise emotion recognition.

D. Provenance information

Image Batch	Source	Licensing Type
All Images	FER2013 Dataset Kaggle	Public Domain / Open Source

II. Data Cleaning

A. Techniques and Methods Applied for Standardizing the Dataset

The primary goals of our data cleaning is to standardize the resolution of images, adjust their brightness, contrast, and sharpness, and ensure consistent naming conventions. The cleaned dataset provides a uniform foundation for effective machine learning model training.

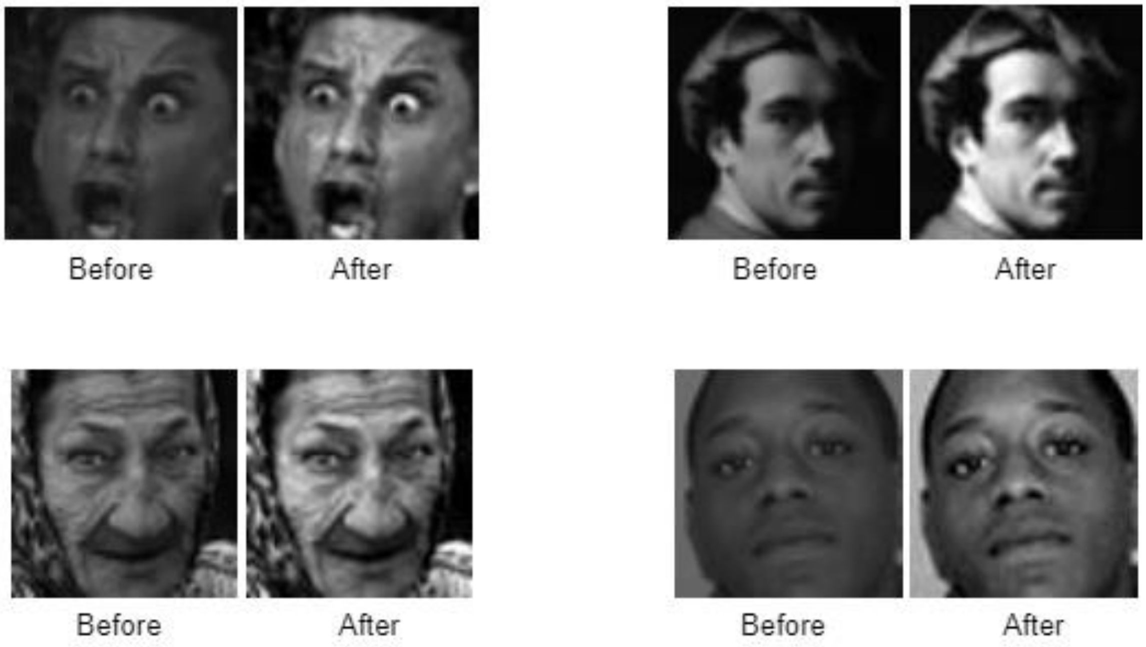


Figure 1: Comparison before and after cleaning dataset images

1. **Resizing and Conversion to Grayscale**:
    - Resizing: All images are resized to a standard resolution (64x64 pixels). This standardization ensures that each image has the same dimensions, facilitating efficient processing and model training.
    - Conversion to B&W: Images are converted to black and white (grayscale). This simplifies the data by reducing it to a single channel, which can reduce computational complexity and focus the model on essential features without the variability introduced by color. Additionally, the dataset that we found was already set in grayscale.
2. **Brightness Adjustment**:
    - High Brightness: Images that are too bright are adjusted to reduce their brightness. This is achieved by calculating the brightness level and reducing it proportionally if it exceeds a certain threshold of 150.
    - Low Brightness: Conversely, images that are too dark are adjusted to increase their brightness. This ensures that all images have a similar brightness level, enhancing the model's ability to learn from them.]

3. **Contrast Adjustment**:
   - High Contrast: Images with excessively high contrast are adjusted to reduce their contrast. This helps in preventing certain features from being overly pronounced, which can mislead the model.
   - Low Contrast: Images with very low contrast are adjusted to increase their contrast. This helps in highlighting features that might otherwise be too subtle for the model to detect.
4. **Sharpness Adjustment**:
   - High Sharpness: Images with high sharpness are adjusted to reduce their sharpness. This helps in smoothing out overly crisp edges that might create noise in the data.
   - Low Sharpness: Images with low sharpness are adjusted to enhance their sharpness. This ensures that the important features and edges within the images are clear and distinguishable.
5. **Naming**:
   - Each image is renamed using a consistent naming convention that includes the directory names and a unique identifier. This helps in maintaining a clear and organized dataset structure, which is crucial for effective data management and retrieval.

B. Challenges Encountered and Solutions

1. **Balancing Brightness and Contrast Adjustments**:
   - Challenge: Determining the right thresholds and adjustment levels for brightness, contrast, and sharpness to enhance image quality. We did not want to add a set fixed increase/decrease size since this would not be proportional to some images on either extremes.
   - Solution: The script uses proportional adjustments based on how much the image's properties deviate from the thresholds. It takes into account how much brighter/darker an image is compared to the threshold and increases/decreases based on that amount.
2. **Handling Existing Cleaned Dataset Directory**:
   - Challenge: If a cleaned dataset directory already exists, running the cleaning process again would result in conflicts.
   - Solution: The script first checks if the cleaned dataset directory exists and removes it if it does. This ensures that the cleaning process starts fresh each time, avoiding conflicts and ensuring the directory contains only the latest cleaned data.
3. **Ensuring Unique File Names**:
   - Challenge: When renaming and saving cleaned images, ensuring that each image has a unique name to prevent overwriting.
   - Solution: The script generates unique file names based on the parent directory, subdirectory, and a count. This systematic approach ensures that each image name is unique and follows a consistent pattern.

III. Labeling

The FER 2013 dataset is already pre-labeled, which simplified the process of labeling the images with their corresponding emotions. However, we had to meticulously reviewed the dataset to retain only the most accurate images that matched their designated emotions and discarded those that did not. This was crucial for achieving better results when training the model, especially since we are using a dataset of 400 images. Each team member was responsible for one emotion category, and we labeled the images accordingly until all four categories were completed. The dataset was then divided into two folders: one for training and one for testing, with each folder containing the four emotions: Happy, Engaged, Neutral, and Angry.

One significant challenge we encountered was labeling the neutral and engaged emotions, as these facial expressions appear very similar. The FER 2013 dataset includes a neutral folder with 1223 images, containing both purely neutral expressions and those with more focused, engaged looks. We worked to separate this folder into two distinct categories: neutral and engaged. For the happy and angry folders, we selected images that most accurately represented these emotions. Some facial expressions were difficult to categorize; for instance, a subtle smile might be mistaken for a neutral expression.

To ensure accuracy, we focused on categorizing each facial expression as appropriately as possible. For example, faces with wide smiles or laughs were categorized as happy, while faces with eyes open wider than usual were placed in the engaged category, indicating a more focused demeanor. Finally, to ensure that the faces were represented accurately, we cross-checked each other's work. This process helped identify and correct any inconsistencies in our dataset, ensuring that the facial expressions were correctly labeled.

IV. Dataset Visualization

A. Class Distribution:

The following is a bar graph showing the number of images in each class. The following data visualization was done using Matplotlib and Pillow [1], [2].

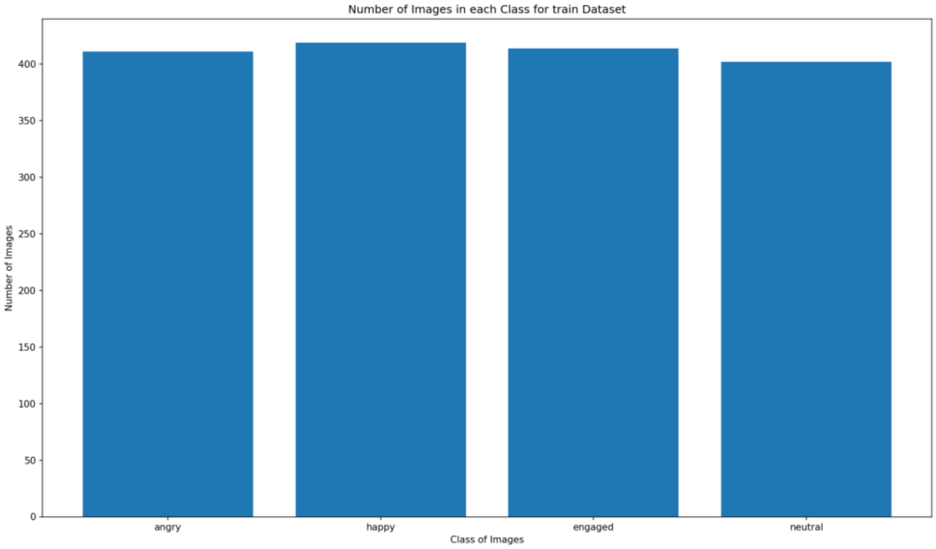


Figure 2: Number of images in each class for the training dataset

The bar graph above shows a near equal number of images for each class. We can thus conclude that none of the classes are overrepresented or underrepresented.

B. Pixel Intensity Distribution:

The following are histograms of the aggregated pixel intensity for each class. Since the images are in grayscale, the pixel intensity will measure the pixel's brightness ranging from 0 to 255, where 0 represents black and 255 is white. The frequency label on the y-axis indicates the number of pixels with the associated pixel intensity.

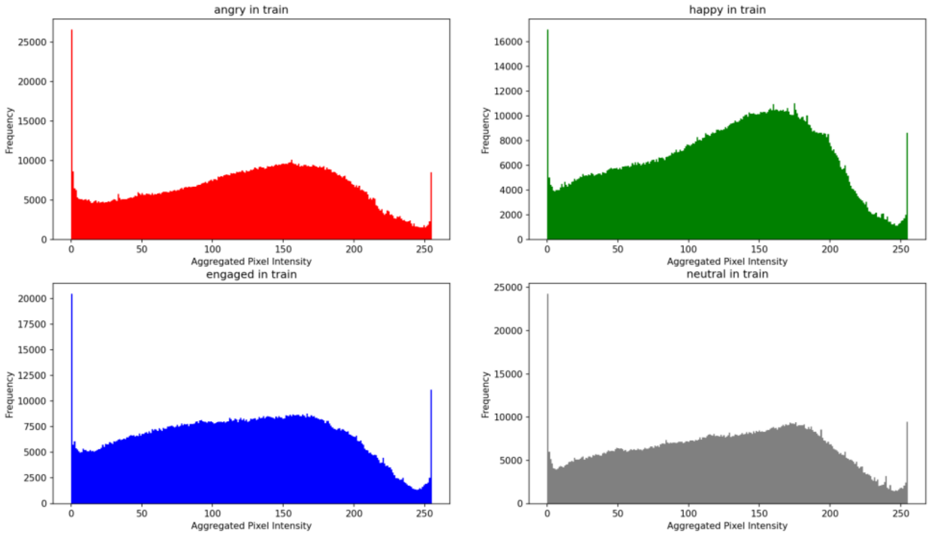


Figure 3: Histograms of aggregated pixel intensity for each class

C. Sample Images:

The following are pixel intensity histograms of 15 randomly sampled images for each class.

Neutral Class:

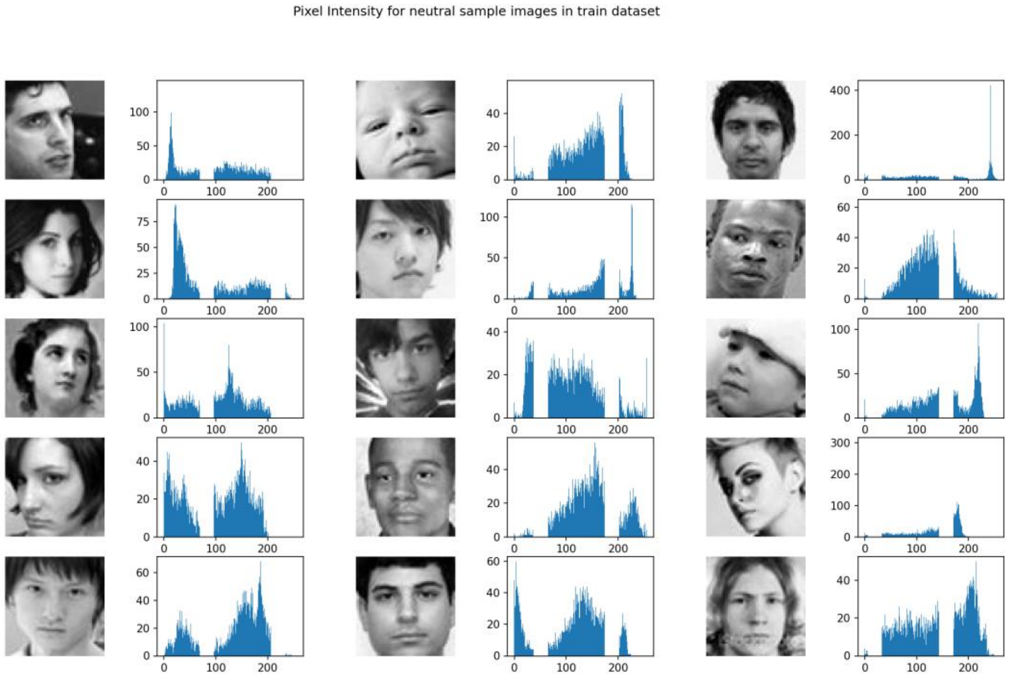


Figure 4: Pixel Intensity Histograms of Images in Neutral Class

Engaged Class:

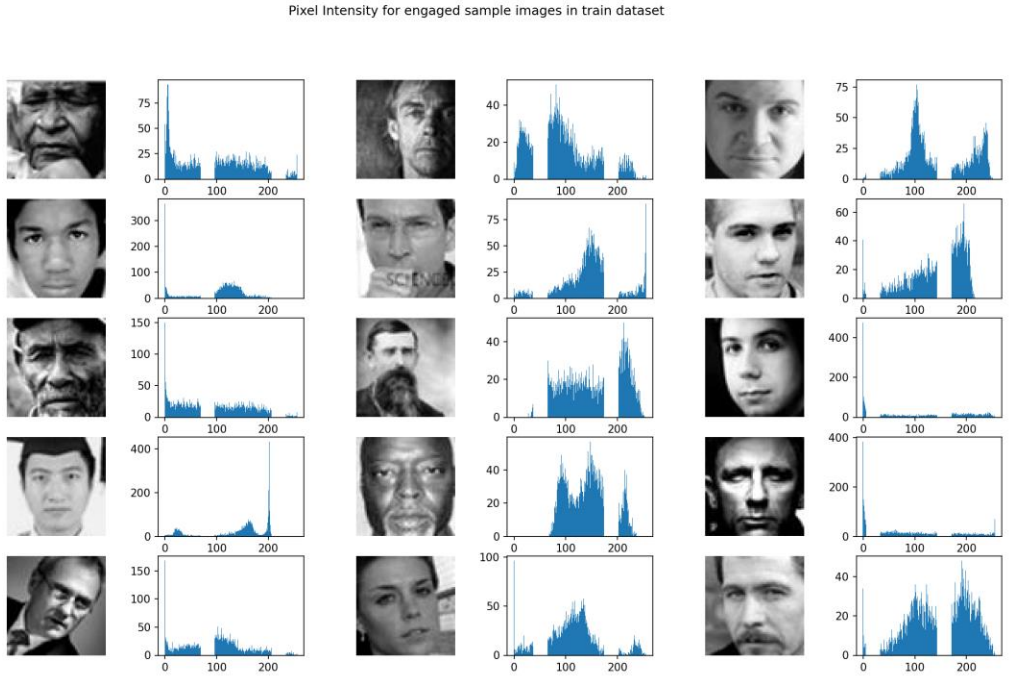


Figure 5: Pixel Intensity Histograms of Images in Engaged Class

Happy Class:

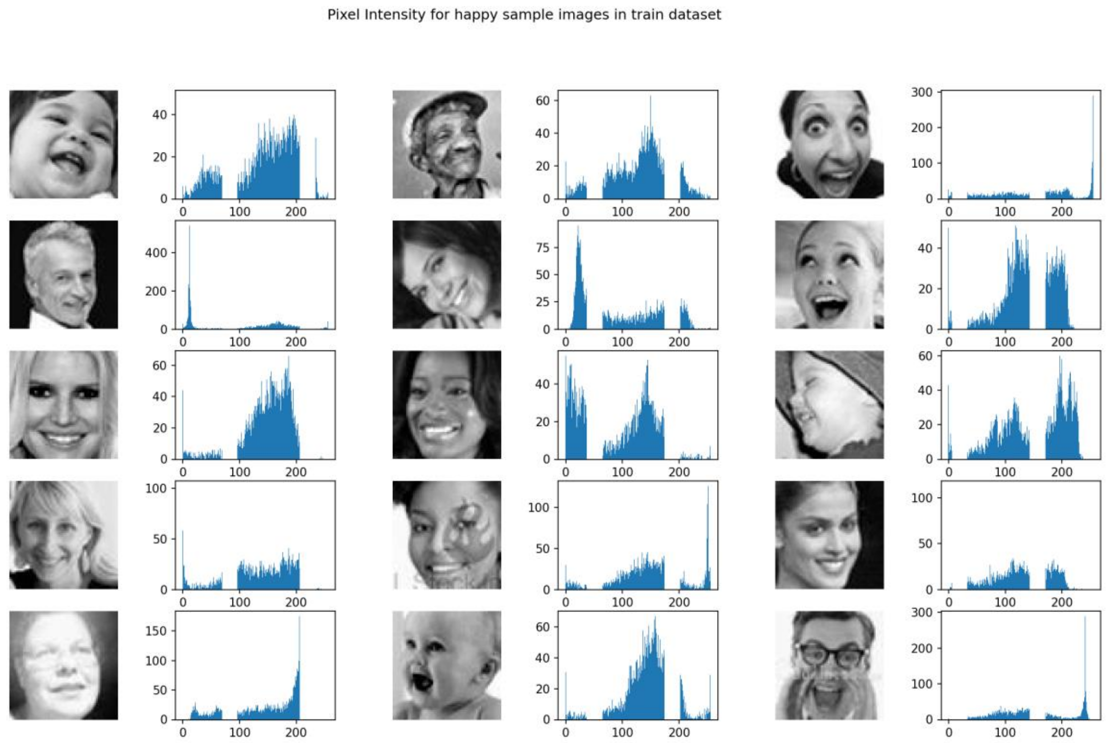


Figure 6: Pixel Intensity Histograms of Images in Happy Class

Angry Class:

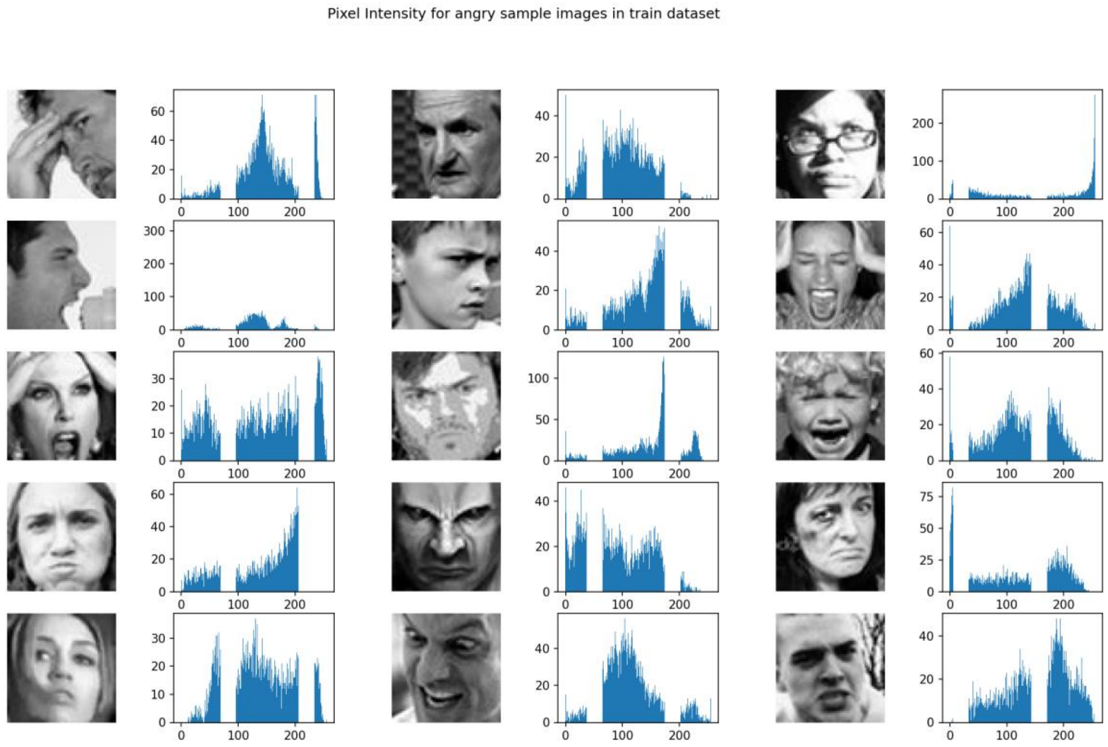


Figure 7: Pixel Intensity Histograms of Images in Angry Class

Part 2

I. CNN Architecture

Main Model:

The main model consists of 6 convolutional layers, each followed by batch normalization to stabilize the learning process and ensure convergence. Each convolutional layer uses the Leaky ReLU activation function, which helps the model learn complex patterns by allowing a small gradient when the unit is not active. Following every other convolutional layer, a max pooling layer is used to down-sample the spatial dimensions, retaining the most significant features while discarding less useful patterns.

The network then connects to a fully connected layer to produce the output class predictions. Before this layer, a dropout layer is added to prevent overfitting by randomly setting a fraction of the input units to zero at each update during training. The convolutional layers are designed with a stride of 1 to ensure detailed pixel-level feature extraction. The model increases the number of output channels progressively: 32 channels in the first two layers, 64 channels in the third and fourth layers, 128 channels in the fifth layer, and 256 channels in the sixth layer. This gradual increase allows the model to capture more detailed and complex features as it goes deeper.

<u>Variants:</u>

- Variant 1:
    - This variant uses only two convolutional layers instead of six. This reduces the model's depth, making it less capable of capturing complex features and patterns. The shallower architecture results in a decrease in accuracy by approximately 10% compared to the main model, highlighting the importance of depth in capturing detailed features.
- Variant 2:
    - This variant uses 2x2 kernels instead of 3x3 kernels in the convolutional layers. The smaller kernel size affects the model's ability to extract features efficiently, requiring more layers to achieve comparable performance. As a result, this model achieves around 1% lower accuracy than the main model. The smaller kernels capture fine details well but may lead to underfitting if not enough layers are used to capture the necessary features.

Training Process:

The training process involved initially setting the number of epochs to 10 and the learning rate to 0.01. However, this configuration did not yield satisfactory accuracy, as the model required more epochs to train the dataset accurately. We adjusted the training setup to 20 epochs and reduced the learning rate to 0.001 to allow for finer adjustments during training.

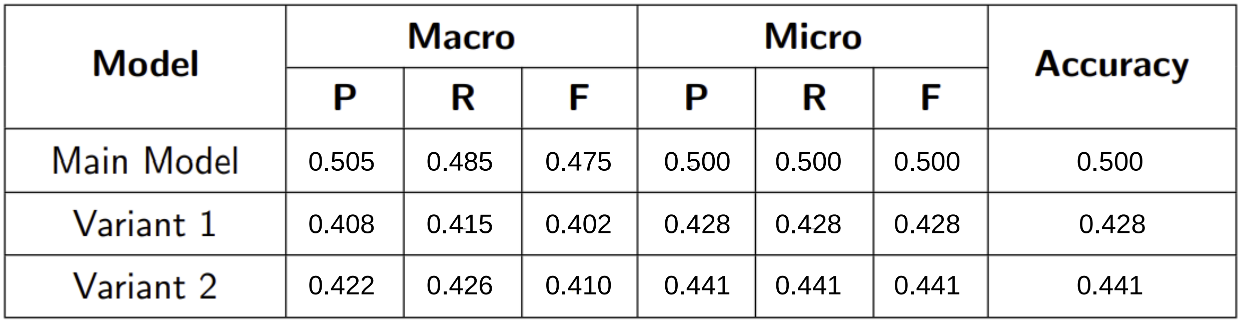
The Adam optimizer was used to adapt the learning rate for each parameter, providing efficient training through mini-batch gradient descent. We implemented early stopping with a patience of 3 epochs to prevent overfitting; training was halted if the validation loss did not improve after 3 consecutive epochs.

The model's performance was evaluated based on the validation set, with the model showing the lowest validation loss saved as the best model. This approach ensured that the model selected was the one that generalized best to unseen data. After training, the model was evaluated on the test set to determine its final accuracy.

Therefore, the main model's deep architecture, with its progressive increase in channels and use of regularization techniques like dropout and batch normalization, allowed it to capture detailed and complex features effectively. Variant 1, with fewer layers, demonstrated the impact of model depth on performance, while Variant 2 showed how kernel size affects feature extraction and model accuracy. The training process, involving careful tuning of epochs, learning rate, and the use of the Adam optimizer with early stopping, was crucial in achieving optimal model performance.

II. Evaluation.

Performance Metrics



Model	Macro			Micro			Accuracy
	P	R	F	P	R	F	
Main Model	0.505	0.485	0.475	0.500	0.500	0.500	0.500
Variant 1	0.408	0.415	0.402	0.428	0.428	0.428	0.428
Variant 2	0.422	0.426	0.410	0.441	0.441	0.441	0.441

Figure 8: Table performance metrics of the main model and its two variants

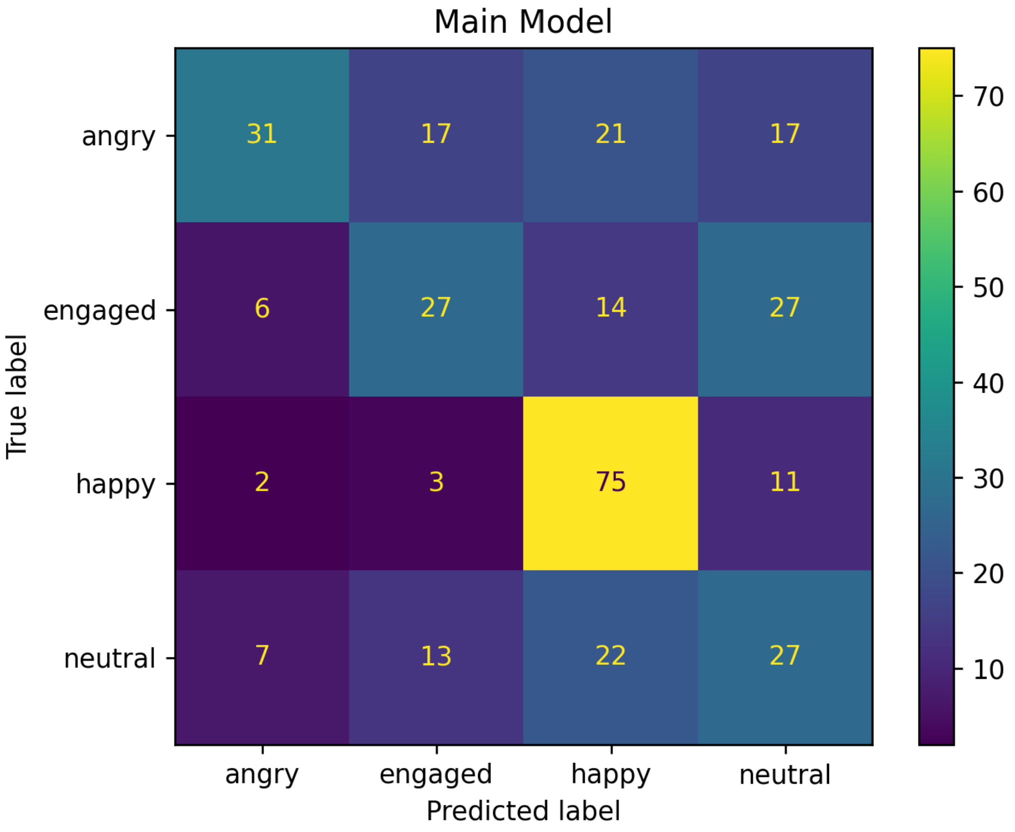
The performance metrics of the Main Model and its two variants reveal significant differences in their ability to classify facial expressions accurately. The Main Model demonstrates balanced and robust performance with a macro precision of 0.505, macro recall of 0.485, and macro F1-score of 0.475. Its micro metrics and accuracy are all 0.500, indicating a consistent ability to correctly identify true positives while minimizing false positives and false negatives. This balance is crucial for facial image analysis applications, where both types of errors can significantly impact the reliability of the results.

In contrast, Variant 1 shows notably lower performance, with macro precision and recall of 0.408 and 0.415, respectively, and a macro F1-score of 0.402. The micro metrics and accuracy are all 0.428. This variant's lower precision suggests a higher rate of false positives, while its lower recall indicates more false negatives. Consequently, Variant 1 may frequently misclassify facial expressions, making it less suitable for applications requiring high accuracy and reliability.

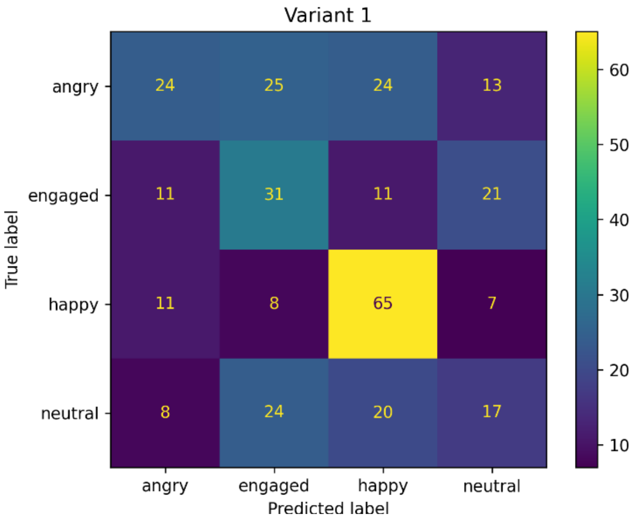
Variant 2 performs better than Variant 1 but still falls short of the Main Model's performance, with a macro precision of 0.422, macro recall of 0.426, and a macro F1-score of 0.410. The micro metrics and accuracy are all 0.441. While it offers a better balance between precision and recall than Variant 1, it still underperforms compared to the Main Model. This makes Variant 2 a more viable alternative if computational complexity is a concern, though it sacrifices some accuracy. Overall, the Main Model stands out as the most reliable for facial image analysis, offering the best balance between precision and recall, which is essential for ensuring accurate and consistent classification of facial expressions.

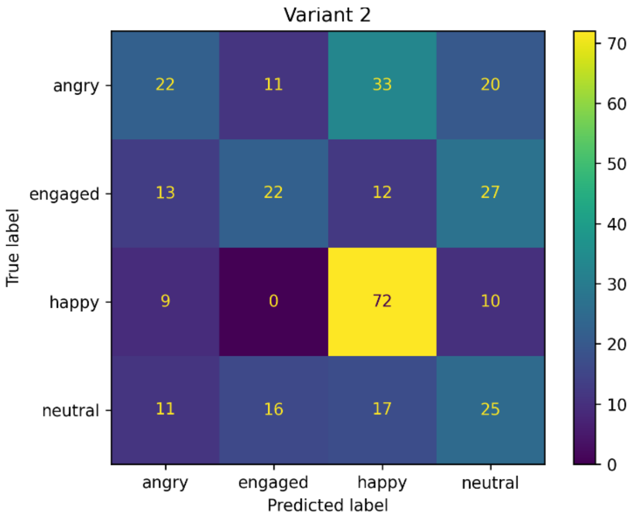
Confusion Matrix Analysis

Confusion Matrix for Main Model:



Confusion Matrix for Variant 1 and Variant 2:





Main Model:

- **Most Frequently Confused Classes:**
  - **Angry and Engaged:** 17 instances
  - **Angry and Neutral:** 17 instances
  - **Engaged and Neutral:** 27 instances

  These confusions could be due to similar facial features in expressions, leading the model to misclassify. The neutral expression, being a common baseline, might be incorrectly predicted when the features are not distinctly recognized.

- **Well-Recognized Classes:**
  - **Happy:** 75 correct predictions
  - **Engaged:** 27 correct predictions

  Distinct features such as smiling in happy expressions are easier for the model to identify. The model's balanced architecture allows it to capture these features effectively.

Variant 1:

- **Most Frequently Confused Classes:**
  - **Angry and Engaged:** 25 instances
  - **Angry and Neutral:** 24 instances
  - **Neutral and Engaged:** 24 instances

  Reducing the number of convolutional layers likely limits the model's ability to learn complex features. As a result, similar expressions are more frequently confused due to insufficient depth to capture detailed nuances.

- **Well-Recognized Classes:**
  - **Happy:** 65 correct predictions
  - **Engaged:** 31 correct predictions

  Despite the shallower architecture, some expressions with distinct and pronounced features like happiness are still relatively well-recognized. However, the overall reduction in layers impacts the depth of feature extraction.

Variant 2:

- **Most Frequently Confused Classes:**
  - **Angry and Engaged:** 33 instances
  - **Neutral and Engaged:** 27 instances
  - **Angry and Neutral:** 20 instances

  Smaller kernels might lead to a higher focus on fine details but miss broader context, resulting in higher misclassification rates for expressions with subtle differences.

- **Well-Recognized Classes:**
  - **Happy:** 72 correct predictions

  - **Engaged:** 22 correct predictions

Smaller kernels help in capturing fine-grained details which might benefit recognizing clear and distinct expressions like happy and engaged. However, the broader context required for nuanced expressions might be lost, affecting overall performance.

Impact of Architectural Variations:

Depth (Number of Convolutional Layers):

Reflecting on how the depth influenced performance, we can see a clear impact on the model's ability to capture detailed features. The main model, with 6 convolutional layers, achieved a test phase accuracy of 50%, significantly higher than variant 1, which had only 2 convolutional layers and achieved an accuracy of 42%. This indicates that a higher depth allows the model to capture more detailed features and recognize patterns better, resulting in improved accuracy. However, deeper networks also carry a higher risk of overfitting due to their increased complexity. Regularization techniques were thus necessary to ensure the model generalized well to unseen data.

Kernel Size Variations:

The variations in kernel size also affected the model's recognition abilities, particularly regarding finer versus broader facial features. The main model, using a kernel size of 3, was more adept at identifying broader features of the sample images, allowing it to achieve better overall accuracy. In contrast, variant 2, which utilized a smaller kernel size of 2, was better at capturing finer details. This difference highlights the trade-off between capturing fine details and broader features, and suggests that a combination of kernel sizes might improve the model's ability to recognize both types of features.

Conclusions and Forward Look:

Summary of Primary Findings:

The main model performed the best among the three different models tested, achieving a test phase accuracy of 50%. This superior performance is attributed to its 6 convolutional layers, which enabled it to capture more features and patterns. To mitigate the risk of overfitting associated with a higher number of layers, regularization techniques such as dropout were employed to enhance generalization.

Suggestions for Future Refinements:

For future refinements in model architecture or training strategies, it is suggested to incorporate variations in kernel sizes across different convolutional layers. This approach would allow the model to process inputs at multiple scales, improving its ability to identify both finer and broader aspects of the images. Additionally, increasing the number of layers could help the model detect more complex features, though it would be crucial to employ additional regularization techniques to prevent overfitting.

References

[1] Matplotlib, “Matplotlib: Python plotting — Matplotlib 3.1.1 documentation,” Matplotlib.org, 2012. https://matplotlib.org/

[2] A. C. (PIL F. Author), “Pillow: Python Imaging Library (Fork),” PyPI. https://pypi.org/project/Pillow/

[3] P.-L. Carrier and A. Courville, "Facial Expression Recognition 2013 (FER2013)," University of Montreal. Available: https://www.kaggle.com/datasets/msambare/fer2013. [Accessed: May 30, 2024].