Team Name

AK-18

Team Member

Mina Jaberi

Student ID: 40176988

Specialization: Data Specialist

Mahshad Mahdavi Moghadam

Student ID: 40295634

Specialization: Training Specialist

Mina Mahmoud Roshanzamir

Student ID: 40253186

Specialization: Evaluation Specialist

Link of Project:

https://github.com/Minajab/AppliedAI_EmotionDetection/tree/main

1. Dataset

i. Overview

In this section, our aim is to provide a comprehensive overview of the dataset that we used for the emotion recognition project.

The dataset was compiled from multiple sources due to the unavailability of all required classes within a single publicly accessible dataset. Specifically, we obtained data for the 'neutral' and 'angry' emotion classes from a publicly available dataset. However, the remaining two classes were sourced from a specific online stock photo.

In our dataset, we have a total of 2,115 images, which are categorized into four distinct classes. Table 1 shows the distribution of instances of each class.

Class	Number of Instances
Neutral	555
Focused	532
Tired	402
Angry	626
Total	2115

Table 1 Number of Instances in Each Class

The data for our project has been collected from two distinct sources: FER-2013 and an online stock of photos[1]. To manage the scale of the dataset, we opted to select images randomly from both sources. FER-2013 alone contains an extensive collection of 35,887 grayscale images, which exceeds the scope of our project.

This dataset exhibits distinctive characteristics that significantly contribute to its suitability for our research:

- Facial Alignment:

  Most of the images in the dataset have been automatically captured to ensure that the face is centered and occupies a consistent amount of space within the image. This feature enhances the dataset's suitability for facial analysis and related research objectives.

- Diverse Backgrounds:

  In addition to consistent facial alignment, the dataset features images collected from online stock photos, offering a variety of backgrounds. This diversity in backgrounds enriches our dataset, providing a broad range of environmental contexts for our analysis.

[1] https://www.pexels.com/

ii. Dataset Selection Justification

In this section, we provide an explanation for our choice of utilizing the FER2013 and Pexels photo stock datasets in our research project. These datasets offer unique advantages and address specific needs in our study.

FER2013 Dataset: The FER2013 dataset plays a pivotal role in our research for the following reasons:

- Dataset Size and Suitability: FER2013 is a relatively large dataset, making it well-suited for the training of neural network algorithms. Its size allows for robust model development.
- Public Availability: This dataset is publicly available, enhancing accessibility for both researchers and students. This openness aligns with our commitment to transparency and collaboration.
- Research Validity: FER2013 has been utilized in numerous research papers, with demonstrated success in facial expression recognition. Its usage in prior studies affirms its reliability and suitability for our research objectives.

However, it's important to acknowledge certain limitations in the FER2013 dataset:

- Resolution Challenge: The images in FER2013 are of relatively low resolution (48x48 pixels). This can pose a challenge as it may not adequately capture fine-grained facial features, potentially limiting our model's ability to recognize subtle emotional expressions.
- Class Limitations: FER2013 has limitations in terms of the classes it includes. Notably, it lacks complex facial expressions such as 'tired' and 'focused.' These constraints prompted us to seek additional resources for these intricate emotional expressions.

Pexels Photo Stock: The choice of the Pexels Photo Stock dataset is driven by the following considerations:

- Resource Versatility: Pexels Photo Stock serves as a valuable resource for projects requiring diverse, high-quality visuals. Its extensive collection of images provides a rich source of visual content, essential for our research.
- Accessibility: Pexels Photo Stock is readily accessible, contributing to the efficiency of our project.

However, it is important to recognize certain challenges associated with Pexels Photo Stock:

- Content Overuse: The popularity of Pexels may result in some images being overused across various projects. This overuse can impact the uniqueness of the content we utilize.
- Differing Image Quality: Images in Pexels Photo Stock vary in quality, which may necessitate careful selection to ensure consistency in the dataset.

In conclusion, our choice of the FER2013 and Pexels Photo Stock datasets is driven by their respective strengths and the specific needs they fulfill in our research. Acknowledging both their advantages and limitations allows us to make informed decisions and utilize the datasets effectively in our project.

iii. Provenance Information

In Table 2, we provide complementary information about our collected dataset.

Batch Image	Source	Dataset Reference	License	Open Access
Neutral	FER2013[2]	https://www.kaggle.com/datasets/msambare/fer2013	https://opendatacommons.org/licenses/dbcl/1-0/	Public Domain
Angry	FER2013	https://www.kaggle.com/datasets/msambare/fer2013	ttps://opendatacommons.org/licenses/dbcl/1-0/	Public Domain
Tired	Pexels	https://www.pexels.com/	https://www.pexels.com/license/	Attribution not required
Focused	Pexels	https://www.pexels.com/	https://www.pexels.com/license/	Attribution not required

Table 2 Information of Each Image Batch

[2] Carrier, Pierre-Luc, Aaron Courville, Ian J. Goodfellow, Medhi Mirza, and Yoshua Bengio. "FER-2013 face database." Universite de Montral 3 (2013).

2. Data Cleaning Phase

A set of clearance functions has been applied to the dataset, which is listed in order below.

1. **Deleting background ():** To enhance the model's precision, background detection, and elimination is essential. In the FER-2013 dataset, the focus is solely on the faces, whereas the pictures obtained from another source exhibit the opposite scenario. As a result, to improve the model's accuracy for two distinct emotions, we have implemented this function for the 'focused' and 'bored' classes. An example output for this function can be found below.

Figure 1 Bored class. The leftmost image is the original, and the rightmost one is the processed version.

2. **Rotation ():** During testing, the input image can exhibit various degrees of rotation. Therefore, to make the model more robust, it is valuable to introduce slight rotations within a subset of the data. In the rotation function, for each training image, a random number between zero and one should be generated. If the random value is less than 0.5, the image should be rotated. The degree of rotation should be a random number between -30 and 30 degrees. Here is an example output:

Figure 2 Angry class. The leftmost image is the original, and the rightmost one is the processed version.

3. **Saturation adjustment ():** This function includes a parameter known as the 'saturation factor.' When the value of this factor is greater than one, it increases the purity of colors in an image. This function has been applied to the 'bored' and 'focused' classes, which contain colorful images with a saturation factor of 1.5 (resulting in a slight alteration). An example output is as below.

Figure 3 Bored class. The leftmost image is the original, and the rightmost one is the processed version.

4. **Grayscale ():** The 'angry' and 'neutral' classes consist of grayscale images, while the remaining classes have colorful images. Therefore, the grayscale transformation has been applied to the colorful classes. The benefits of transforming the images to grayscale are as follows:
   a. Grayscale images contain only one channel, as opposed to RGB images, which have three channels. Consequently, converting to grayscale results in reduced memory usage and a faster training procedure.
   b. Grayscale images retain essential information while eliminating potentially distracting color information. The training model's focus is solely on the intensity and luminance of the pixels, simplifying the feature extraction and training stages.

Figure 4 Focused class. The leftmost image is the original, and the rightmost one is the processed version.

5. **Contrast Regulation ():** After applying the grayscale function to all emotional classes, contrast regulation should be performed. The rationale behind this is that emotional expressions are often conveyed through facial features, and ensuring

consistent and controlled contrast can help the model better focus on these features. The 'contrast-factor' parameter determines the intensity of contrast and has been set to 1.1, resulting in a slight increase in contrast.

Figure 5 Neutral class. The leftmost image is the original, and the rightmost one is the processed version.

6. **Morphology (), Denoise (), and Deblock ():** The images collected from the FER-2013 dataset are notably noisy. When zoomed in, artifacts are easily visible. Therefore, three functions—Morphology (), Denoise (), and Deblock ()—have been applied in sequence.

   Morphological operations involve a chain of dilation and erosion, which helps reduce or eliminate artifacts and blocks. Dilation regularizes and smooths boundaries, while erosion is effective at removing small noise or isolated pixels in an image [3].

   For denoising, the fast mean denoising method has been used. Unlike some noise reduction techniques that blur or smooth images, non-local means denoising aims to preserve important details and structures in the image. As a result, it is a good choice for denoising the FER-2013 images, which may not be of high quality.

   In the deblocking process, we have two steps: the application of the bilateral filter and Gaussian blur. The bilateral filter reduces compression artifacts, often visible as blocky patterns while preserving edges and fine details in the image. Since relying solely on the bilateral filter was not sufficient for mitigating the blocks, we subsequently applied Gaussian blur, making the artifacts less pronounced and visually disruptive. Example outputs after applying these three deblocking functions are as follows.

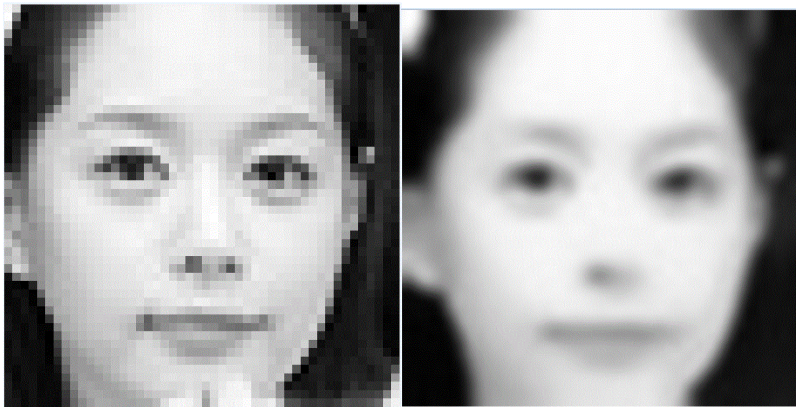

Figure 6 Neutral class. The leftmost image is the original, and the rightmost one is the processed version.

7. **Resizing the image ():** We calculated the maximum width and height among the images from different classes and then adjusted the size of the images to match the maximum width and height.

3. Labeling:

While we collected our dataset from different resources, all our data was pre-labeled.

- FER2013 Dataset: For the classes related to emotions (Angry, Neutral), we utilized the dataset, which was publicly available and had pre-labeled emotional expressions.
- Pexels Photo Stock: For classes such as Tired and Focused, we sourced images from the Pexels Photo Stock website. These images were categorized based on the presence of tired or focused expressions.

The utilization of the FER2013 dataset and images from Pexels Photo Stock was instrumental in labeling our dataset accurately and efficiently.

4. Visualization phase

i. Class Distribution Analysis

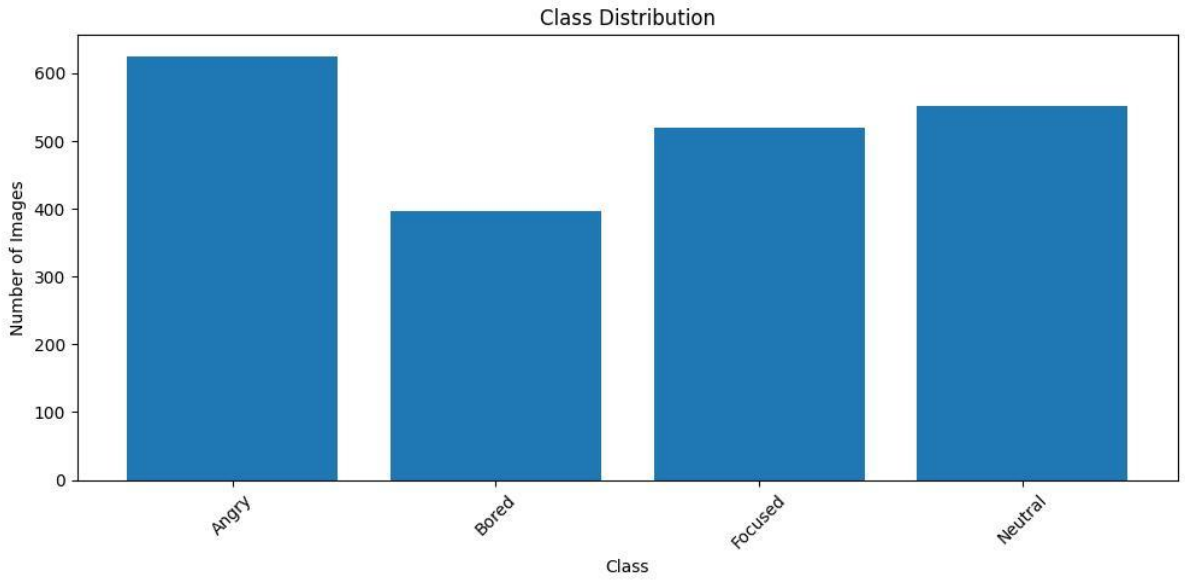

figure 7 (class distribution)

- **Methodology**: We utilized the class_dist function, which leveraged Matplotlib to generate a bar graph showcasing the number of images in each class.

- **Findings (Refer to Figure 7)**:
  - We have four distinct emotion classes: Angry, Bored, Focused, and Neutral.
  - The Angry class contains the highest number of images, highlighting ample data for this emotion.

- The Bored class has fewer images, hinting at the possible need for data augmentation to balance this class.
- Focused and Neutral classes are almost equal in terms of the number of images they contain.
- An imbalance in dataset distribution can lead to a model bias, making this visualization crucial for identifying such concerns beforehand.

ii. Sample Image Visualization

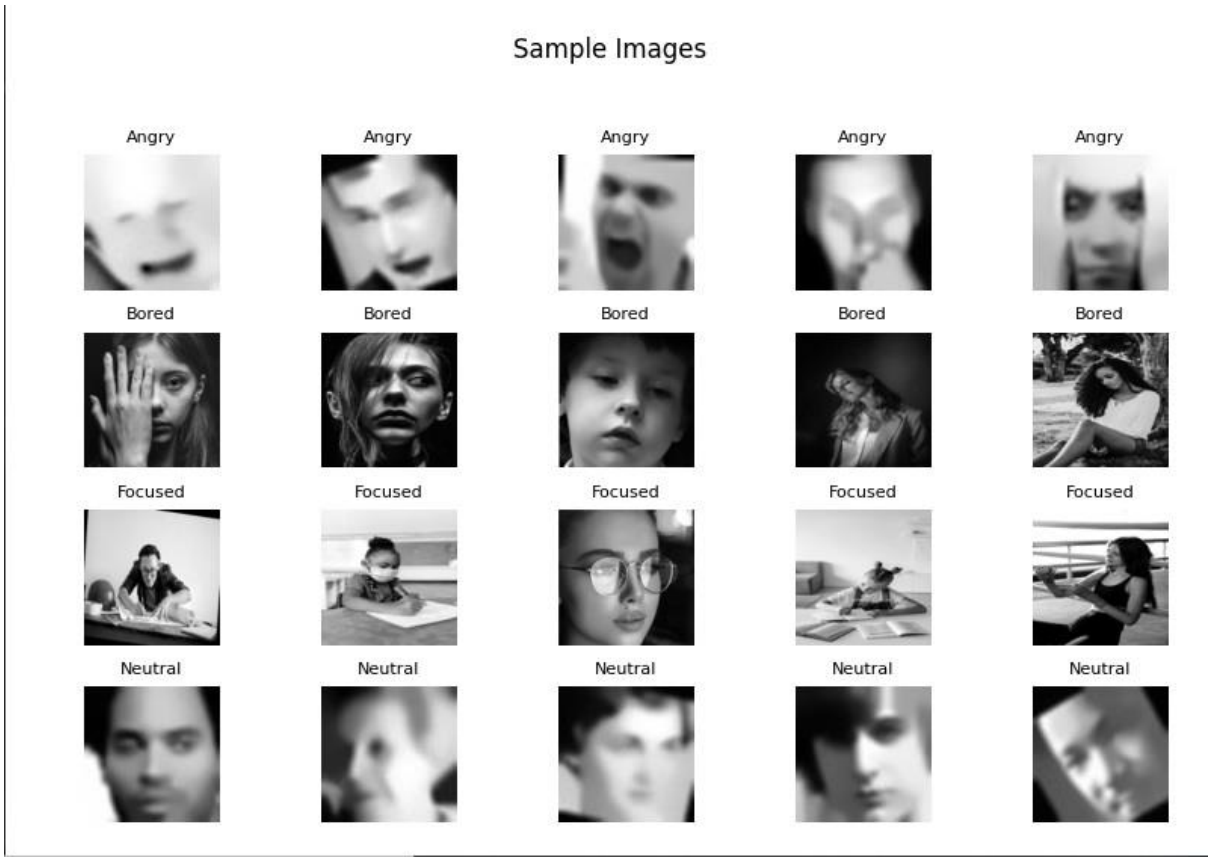

Figure 8 (sample images)

- **Methodology**: We employed the sample_imgs function, which is designed to showcase a set of images from various classes in a 5x5 grid format.

- **Findings (Refer to Figure 8)**:
  - **Quality and Clarity**: There is a variation in the quality of images; some are sharp, while others are blurred.
  - **Variability**: The dataset showcases a range of subjects in terms of people and their poses, emphasizing the dataset's diversity.
  - **Potential Anomalies**: It is essential to periodically check these images to identify any mislabeling or inconsistencies, as these can affect model performance.

iii. Pixel Intensity Distribution Analysis

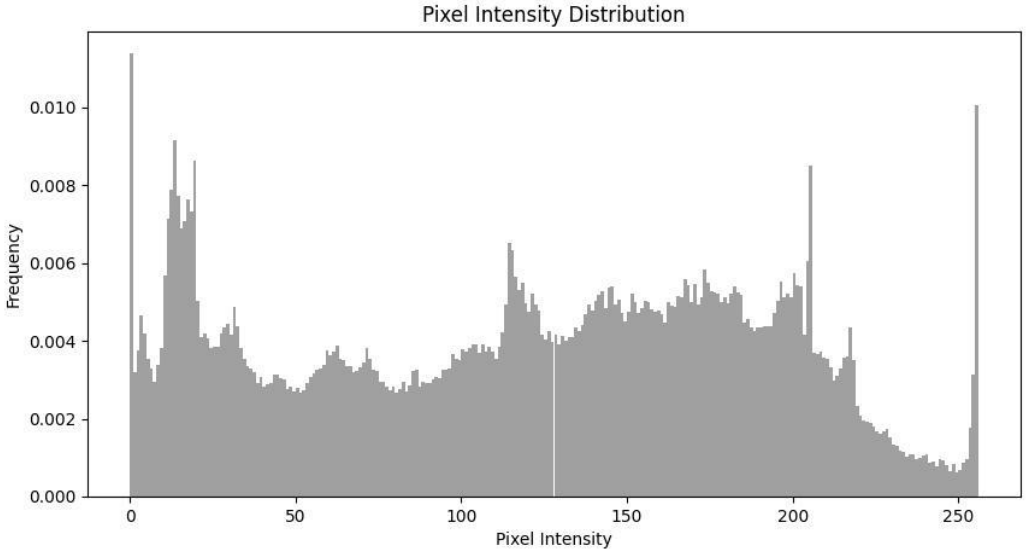

Figure 9 (pixel intensity distribution)

- **Methodology**: We introduced the pixel_dist function, aimed at displaying a histogram to depict the distribution of pixel intensities across a list of images.

- **Findings (Refer to Figure 9)**:
    - **Dark Pixels**: A notable peak around the 0-50 intensity range indicates many dark pixels in the images, possibly due to shadows, dark backgrounds, or objects.
    - **Mid-Tone Pixels**: The 100-150 range displays a balanced distribution of pixels, likely representing objects in typical lighting conditions.
    - **Bright Pixels**: A significant peak near the 250-intensity range suggests a considerable number of very bright or white pixels, which could be due to overexposed areas or flash reflections.

- **Insights**:
    - **Variations in Lighting**: The wide range of intensities indicates that the dataset contains images with varied lighting conditions, which is beneficial for training models.
    - **Potential Anomalies**: The distinct peaks and troughs in the histogram can help in spotting anomalies like overexposed or underexposed images. In such cases, data augmentation might be necessary.

CNN architecture:

1. Model Overview and Architecture Details:

The deep learning model comprises two parts: convolutional layers for detecting image features and a multilayer perceptron (MLP) for classification. The MLP receives the output of the last convolutional layer as a flattened vector and predicts the emotional class. The architectural specifications of the convolutional neural network for the main model are as follows:

- The model consists of three convolutional layers, each with 64, 128, and 256 neurons, respectively.
- In each convolutional layer, the steps of padding, filtering, batch normalization, applying the activation function, and maximum pooling should be followed in order.
- There is no padding in the main model.
- The filters should be applied with a kernel size of three and a step size of one.
- Apply the non-linear activation function ReLU to the normalized output of the convolutional operation.
- Apply the max-pooling algorithm with a moving window size and a stride of two.
- After passing through the convolutional layers and obtaining the 2D recognized features, they should be flattened into one vector and fed into the last block, which is an artificial neural network for predicting the emotion label.
- The MLP contains four layers. The input vector to the two middle layers and the output layer should be normalized, and then the ReLU function is applied as the activation function for the neurons.

Regarding controlling the number of parameters, it is known that having many parameters leads to overfitting. To avoid this, 50% of the weights in each layer will be discarded only in the training phase.

To understand the impacts of changing the kernel size and learning depth, the following modifications have been applied to the main architecture of the model to create different variations.

Table 1 Different variations of the model

Models	Kernel size	Dimensions of the convolutional layers
Main model	**3**	**Three convolutional layers with 64, 128, and 256 neurons, respectively.**
Variation one	2	Three convolutional layers with 64, 128, and 256 neurons, respectively.
Variation two	5	Three convolutional layers with 64, 128, and 256 neurons, respectively.
Variation three	3	Two convolutional layers with 64, and 128 neurons, respectively.
Variation four	3	Four convolutional layers with 64, 128, 256, and 512 neurons, respectively.

2. Training Process:

The dataset, which contains images of four emotional classes (angry, bored, focused, and neutral), should be divided into three groups: training, validation, and test datasets. The 'train_test_split' function from scikit-learn is used for this purpose, allocating 70%, 15%, and 15% to the training, test, and evaluation datasets, respectively. In this phase, the training and validation datasets are utilized.

For the training procedure, various hyperparameters must be considered during the model's design. These parameters are as follows:

- Number of epochs: In each epoch, the entire training dataset should be seen once. Consequently, the epoch number indicates how many times the entire training dataset will be observed. For this model, the epoch value is set to 100. At the end of each epoch, the model's performance is measured using the validation dataset.
- Number of batches: This parameter specifies how many batches the entire training dataset should be divided into during each epoch. The weights are updated after calculating the output error for one batch. In this model, the batch size has been set to 10.
- Learning rate: The learning rate determines the speed of weight adjustments. For this model, the learning rate is set to 0.01, providing a balance between rapid and gradual changes.
- Activation function of the neurons: ReLU, a non-linear function, is chosen as the activation function. One advantage of this function compared to sigmoid is that it leads to sparse activations, meaning that the output of many neurons is zero. This is beneficial for the efficiency and resource utilization of the neural network.
- The chosen loss function is categorical cross-entropy. This loss function aims to minimize the difference between the true and predicted occurrence probabilities of different classes.

The output of this phase includes two models saved in a specified path after running the stage: the best model resulting in the optimal performance for the validation data within 100 epochs and the final model after 100 epochs. The best weights contribute to the evaluation part and serve as the output of this phase.

The following images display the loss function versus the number of epochs for both the training and validation datasets, showing results for the main model and various variations.

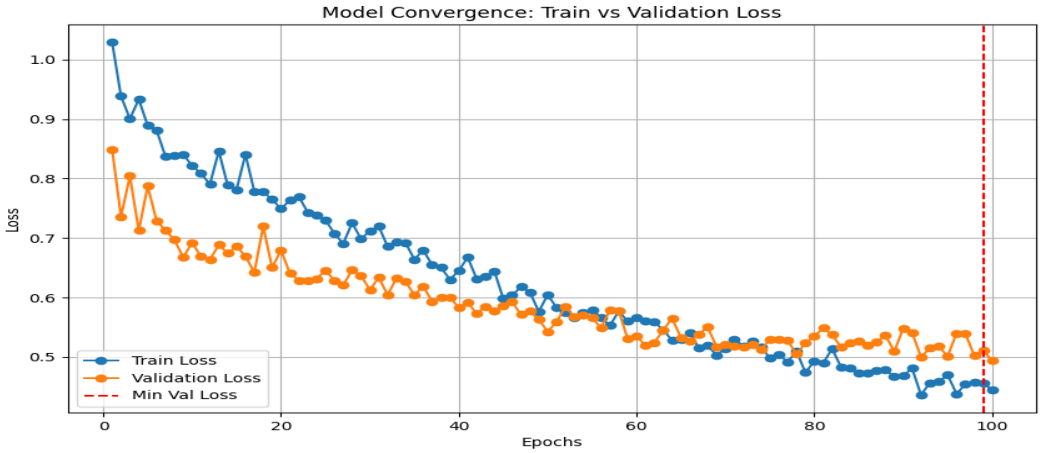

Figure 1 Loss/Epochs for the main model (kernel size:3, the dimension of convolutional layers: 64, 128, and 256)

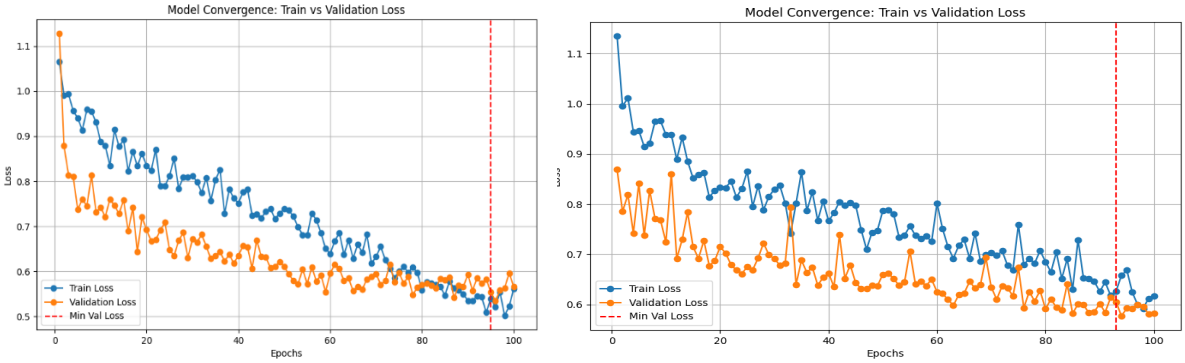

Figure 2 Loss/Epochs for variations 1 (left side image) and 2 (right side image)

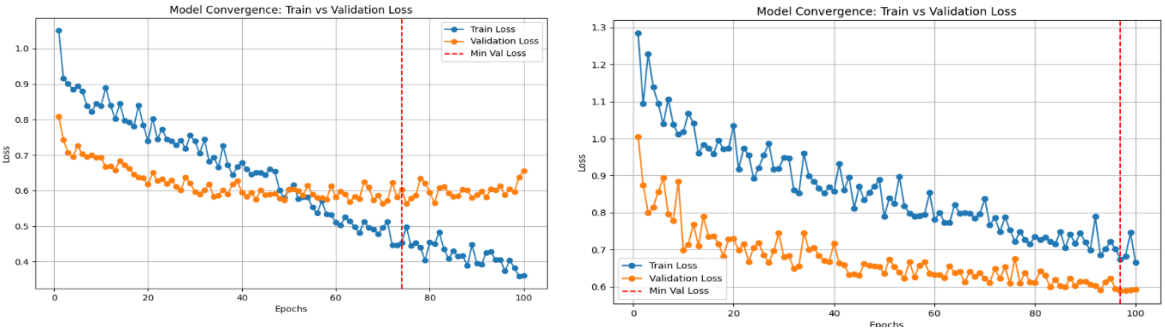

Figure 3 Loss/Epochs for variations 3 (left side image) and 4 (right side image)

Evaluation part II:

In this section, we assess the evaluation results of two variations in addition to the main model. As demonstrated in the previous section, we explored two kernel sizes (one larger and one smaller than the kernel size of the main model) and two learning depths (one greater and one smaller than the number of convolution layers in the main model).

Based on the figures presented above, the best model with a kernel size of two exhibits a lower loss value compared to the kernel size of five. Regarding learning depth, having four convolution layers is more effective than two layers. Consequently, variations one and four are chosen among the four examined variations.

1. Performance metrices

In this section, we present a comprehensive analysis of the performance metrics, including accuracy, precision, recall, and F1-measure, for the Main Model and its two variants, as outlined in the predefined table. Through these metrics, we aim to provide a nuanced understanding of each model's efficacy. By comparing and contrasting the performance of the Main Model with its variants, we offer insights into their respective strengths and weaknesses. For example, if one model exhibits higher recall but lower precision in the context of facial image analysis, we delve into the implications of this trade-off. This comparative evaluation serves to elucidate the distinctive characteristics of each model, contributing to a more informed interpretation of their performance in specific domains or tasks.

Table 2 Comparison of the different variations

	Macro			Micro			
Config	**P**	**R**	**F1-measure**	**P**	**R**	**F1-measure**	**Accuracy**
Main model	79.63%	76.69%	75.19%	74.61%	74.61%	74.61%	74.61%
Variation one	68.45%	66.53%	65.93%	65.38%	65.38%	65.38%	65.38%
Variation four	64.52%	62.84%	62.52%	63.78%	63.78%	63.78%	63.78%

The main model has the highest scores for precision and recall among the three models. High precision indicates that it is often correct when it classifies a facial expression. High recall means it is good at identifying most expressions correctly.

Variant one has lower precision and recall. This means it is not as accurate in classifying expressions and it also misses more expressions than the main model.

Variant four has the lowest recall, which means it fails to identify many expressions correctly. However, its precision is a little better than Variant one, suggesting that when it does make a classification, it is right more often than Variant one. Despite this, it still does not perform as well as the main model.

2. Confusion matrices

In this section, we showcase the evaluation metrics for our model. Initially, we present confusion matrices, revealing how our model performed in classifying different categories. We then analyze

instances where classes were commonly confused, exploring potential reasons tied to the dataset or model specifics. Additionally, we highlight successful classes, speculating on the factors contributing to their accurate classification. This section aims to provide a straightforward assessment of our model's strengths and areas for improvement.

Table 3 Precision and recall for the main model on the test dataset.

Class	Recall	Precision
Angry	39.5%	84.33%
Bored	91.33%	90.13%
Focused	92.02%	58.36%
Neutral	83.87%	85.71%

Table 4 Main Model

		Prediction			
		Angry	Bored	Focused	Neutral
Real	Angry	70	0	107	0
	Bored	0	137	0	13
	Focused	13	0	150	0
	Neutral	0	15	0	78

Table 5 Precision and recall for the variation 1l on the test dataset.

Class	Recall	Precision
Angry	38.70%	60%
Bored	64.40%	86.36%
Focused	92.30%	76.59%
Neutral	70.73%	50.87%

Table 6 Variation 1

		Prediction			
		Angry	Bored	Focused	Neutral
Real	Angry	36	0	1	56
	Bored	0	38	21	0
	Focused	0	6	72	0
	Neutral	24	0	0	58

Table 7 Precision and recall for the variation 4 on the test dataset.

Class	Recall	Precision
Angry	75.26%	58.82%
Bored	52.54%	70.45%
Focused	83.33%	69.89%
Neutral	40.24%	58.92%

Table 8 Variation 4

		Prediction			
		Angry	Bored	Focused	Neutral
Real	Angry	70	0	0	23
	Bored	0	31	28	0
	Focused	0	13	65	0
	Neutral	49	0	0	33

Reasons behind misclassification?

Several points extracted from the misclassification analysis are listed below.

- In terms of data points that are not classified properly, it is observed that for the 'angry' dataset in both the main and the first variation models, only approximately 40% of the actual angry test images are correctly labeled. This may be attributed to the nature of the data, as both the 'angry' and 'neutral' datasets have been obtained from the FER dataset. The issue with this dataset lies in its significant noise and low resolution. Despite the cleaning phase's efforts to mitigate noise, the problem persists even with a smaller kernel size (an attempt to learn more detailed features). The same issue is observed with the neutral dataset, and increasing the number of layers has not provided a solution.
- For the 'bored' dataset, with the variation four models, it can be seen that roughly half of the real bored images can be recognized. Regarding the other classes, except for the class of 'angry,' it can be seen that raising the number of learning layers does not improve performance. The figure related to the loss shows that by increasing the number of layers, the best performing model was trained for at most 10 epochs significantly decreases the difference between the true and predicted occurrence probabilities (roughly the same as the main model); however, the precision metrics worsen. The reason behind that could be overfitting, even when randomly discarding 50% of the weights.

Motives behind perfect classifications?

Good classifications are highlighted in green, and these accurate predictions occur with the main model and the first variation, both of which have three convolutional layers and kernel sizes of three and two, respectively. However, increasing the depth of learning (variation number four) results in uninteresting outcomes due to overfitting. It may be beneficial to experiment with increasing the number of layers concurrently with adjusting the ratio of discarded weights to address this issue.

3. Architectural Variations

Depth Impact on CNN Performance:

The main model and Variant 4 differ in depth. The main model has three convolutional layers with sizes 64, 128, and 256, while Variant 4 adds an additional layer, making it four layers deep with sizes 64, 128, 256, and 512.

From the results, main model, with one less convolutional layer than Variant 4, actually performs better. This suggests that the extra layer in Variant 4 does not necessarily help the model capture more useful features; in fact, it could be memorizing the training data., which we refer to as overfitting. Overfitting happens when a model learns the details and noise in the training data to the extent that it negatively impacts the performance of new data. This seems to be the case with Variant 4, where the additional depth might have made the model too complex for the task at hand.

On the other hand, the main model, with one fewer layer, seems to have a good balance between learning the important features and not overfitting, as indicated by its superior performance metrics.

Kernel Size Variations:

- The smaller kernel size in Variant 1 means that the filter is looking at fewer pixels at a time. This can be good for picking up finer details in the image because it's focusing on a small area. However, this can also be a disadvantage because it might miss the bigger picture or broader features. In the context of facial recognition, a smaller kernel might be better at detecting fine details like small changes in expression, but it might not be as good at understanding the overall emotion being expressed.
- The main model's larger kernel size of 3 means it's taking in more pixels at a time. This can help the model recognize broader features better because it's seeing more of the image at once. It's less focused on the tiny details and more on the overall patterns. For facial expressions, this might mean it's better at recognizing general states like 'happy' or 'sad' because it's looking at the whole face rather than getting stuck on the details.

4. Architectural Variations

Summarizing the primary findings:

The main model achieved the highest precision and recall values, indicating that it was the most accurate in classifying the facial expressions correctly. It managed to maintain a good balance between detecting finer details and broader patterns within the images, which is crucial for recognizing complex facial expressions accurately.

The reasons for the main model's superior performance could be attributed to its ability to capture just the right amount of detail without overfitting, thanks to its optimal kernel size and the number of layers. This balance allowed the model to generalize better to new data, making it more reliable for real-world applications where it's essential to understand students' engagement and emotions accurately.

In contrast, Variant 1, with a smaller kernel size, did not perform as well, potentially because it focused too much on finer details and missed the broader context of facial features. Variant 4, despite having an additional layer, did not improve upon the main model's performance, suggesting that the extra depth may have led to overfitting, where the model learned the training data too well but failed to predict new examples correctly.

Future refinements:

To enhance the performance of our facial expression recognition model, we suggest implementing several key refinements. First, incorporating dropout layers and regularization techniques will be crucial to prevent overfitting, a common challenge in more complex models. Adjusting the learning rate, perhaps with the assistance of learning rate schedulers, could significantly improve the efficiency and effectiveness of our model's training.

Enhancing our approach to data augmentation can aid in generalizing the model to a broader range of data, ensuring it performs well in diverse real-world scenarios. Exploring hybrid architectures, such as combining the strengths of convolutional neural networks with recurrent neural networks, might provide deeper insights, especially for dynamic data like facial expressions.

Integrating attention mechanisms could enhance the model's accuracy by focusing on the most informative parts of facial images, such as the eyes and mouth. It's also worth exploring the balance between the depth and width of our network, potentially adjusting the number of layers and neurons to find an optimal structure.

Utilizing transfer learning, especially by fine-tuning a pre-trained model on our specific dataset, could offer significant performance enhancements, particularly when our training data is limited. Systematic hyperparameter optimization and rigorous testing through methods like k-fold cross-validation are essential to ensure the model's robustness and consistency. Using a diverse dataset for testing will also help in reducing bias and improving the model's applicability.

Finally, maintaining a focus on ethical considerations and actively working to mitigate biases in our model is crucial. This not only ensures technical soundness but also upholds the ethical integrity of our application, an increasingly important aspect in the field of artificial intelligence.

Evaluation part III:

Confusion Matrices of the Best Performing Model Before Bias Mitigation:

The evaluation of the best-performing model before implementing bias mitigation techniques involved the analysis of confusion matrices across ten test folds. These matrices provide insights into the model's performance on a class-wise basis.

Among the test folds, the highest recognition accuracy was observed in test fold seven. This indicates that the model excelled in its predictive capabilities when confronted with the data in this specific fold. A noteworthy observation is the lowest precision in recognizing class I, which occurred in test fold six. This implies that the model struggled to accurately identify instances of class I in this fold. Precision is a crucial metric as it reflects the ability of the model to avoid false positives in predictions.

In the remaining test folds, the model demonstrated a consistent level of recognition performance. This suggests that, overall, the model maintained a stable predictive ability across most of the test scenarios. A distinctive aspect of the analysis is the identification of non-uniform data distribution, particularly noticeable in test folds six and seven. This highlights potential variations or imbalances in the dataset that impact the model's performance differently across these specific folds.

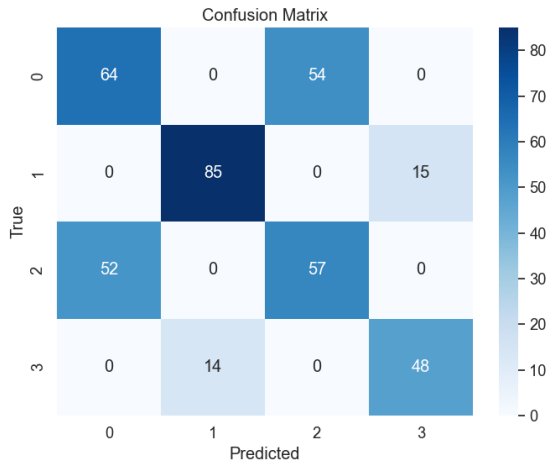

Figure 4 K-fold one

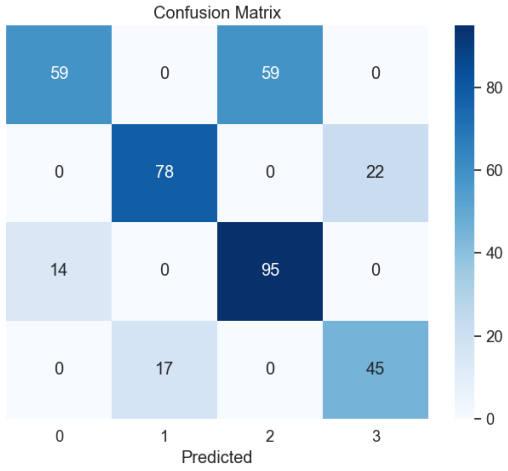

Figure 5 K-fold two

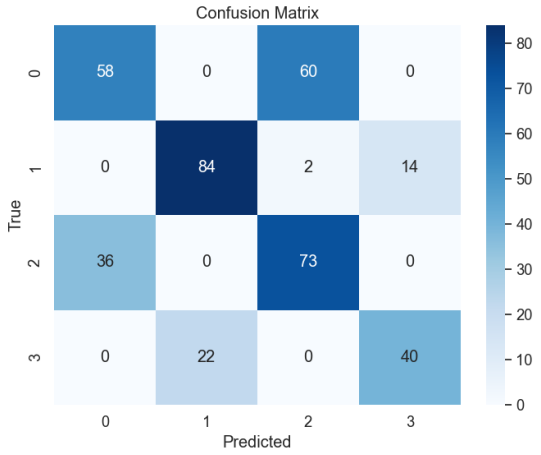

Figure 6 K-fold three

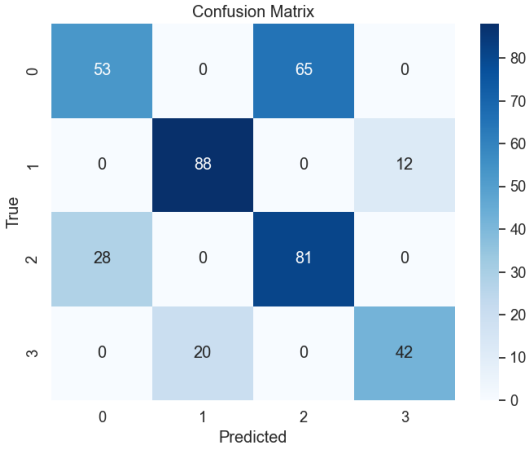

Figure 7 K-fold four

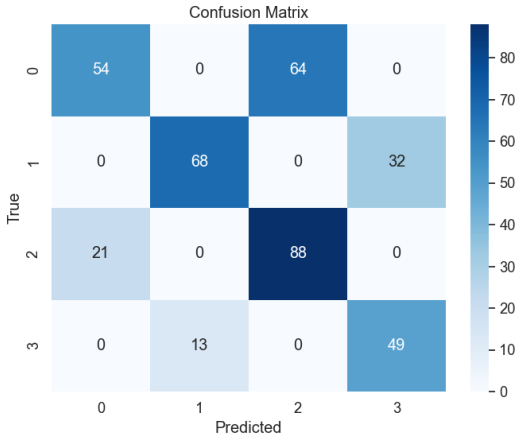

Figure 8 K-fold five

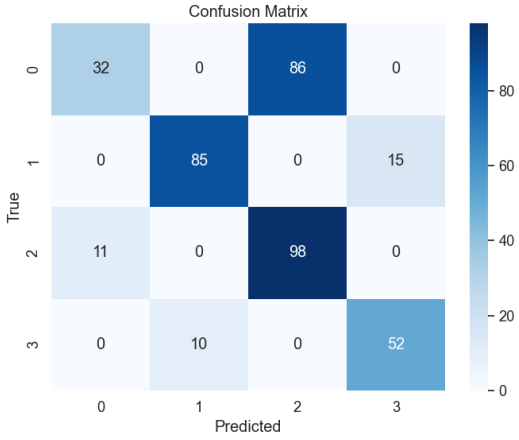

Figure 6 K-fold six

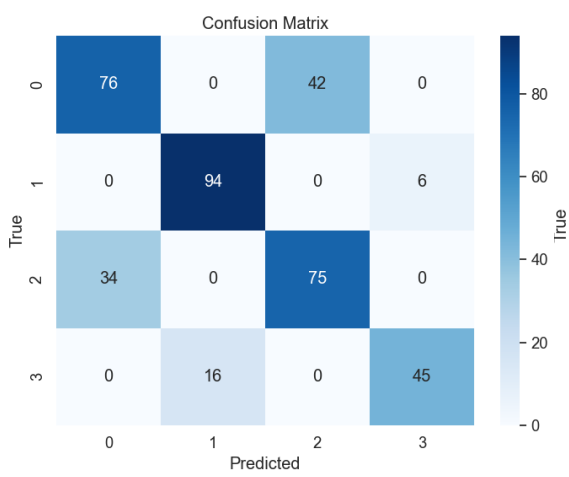

Figure 7 K-fold seven

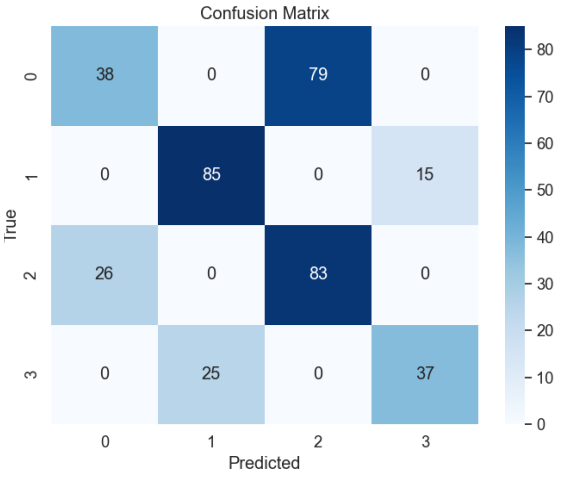

Figure 8 K-fold eight

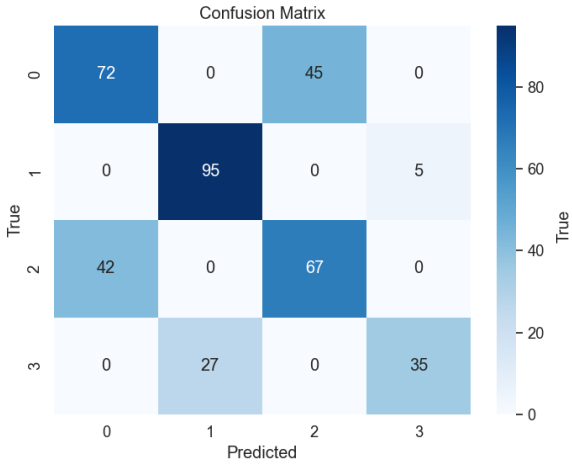

Figure 9 K-fold nine

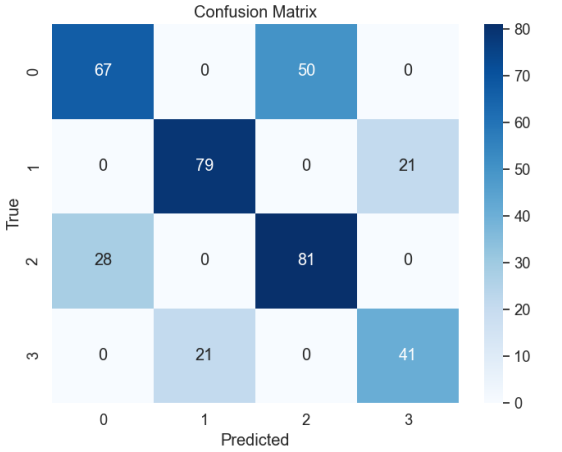

Figure 10 K-fold ten

Performance Results of Each Fold:

In this section, we present a detailed examination of the performance of our best performing machine learning models across ten folds, both before and after the application of bias mitigation technique. The results are shown in Tables 1 and 2, presenting the model's micro and macro precision, recall, F1-score and accuracy. Table 1 elucidates the model's performance before bias mitigation, while Table 2 reflects the outcomes after implementing mitigation strategies. Our discussion delves into significant observations and trends discerned across these folds, emphasizing the consistency of the model's performance as a key focal point. This analysis serves as a foundation for understanding the nuances of our models' predictive capabilities and gauging the efficacy of bias mitigation interventions.

Table 9 Performance Details of the Best Model Part II

	Micro			Macro			
Fold	**P**	**R**	**F**	**P**	**R**	**F**	**Accuracy**
1	0.6529	0.6529	0.6529	0.6714	0.6723	0.6718	0.6529
2	0.7120	0.7120	0.7120	0.7294	0.7193	0.7094	0.7120
3	0.6555	0.6555	0.6555	0.6727	0.6616	0.6626	0.6555
4	0.6786	0.6786	0.6786	0.7004	0.6874	0.6845	0.6786
5	0.6658	0.6658	0.6658	0.6858	0.6838	0.6676	0.6658
6	0.6863	0.6863	0.6863	0.7369	0.7147	0.6861	0.6863
7	0.7474	0.7474	0.7474	0.7672	0.7524	0.7572	0.7474
8	0.6262	0.6262	0.6262	0.6475	0.6332	0.6227	0.6262
9	0.6932	0.6932	0.6932	0.7208	0.6861	0.6929	0.6932
10	0.6907	0.6907	0.6907	0.6937	0.6917	0.6895	0.6907
Average	0.6809	0.6809	0.6809	0.7026	0.6902	0.6844	0.6809

The next table shows the performance details of the best model from part three after data mitigation.

Table 10 Performance Details of the Best Model Part III (After Bias Mitigation)

	Micro			Macro			
Fold	**P**	**R**	**F**	**P**	**R**	**F**	**Accuracy**
1	0.6259	0.6259	0.6259	0.6413	0.6221	0.6243	0.6259
2	0.6412	0.6412	0.6412	0.6396	0.6357	0.6302	0.6412
3	0.5419	0.5419	0.5419	0.5609	0.5401	0.5455	0.5419
4	0.6488	0.6488	0.6488	0.6550	0.6497	0.6431	0.6488
5	0.6412	0.6412	0.6412	0.6410	0.6371	0.6330	0.6412
6	0.6564	0.6564	0.6564	0.6563	0.6571	0.6557	0.6564
7	0.7251	0.7251	0.7251	0.7344	0.7249	0.7263	0.7251
8	0.6946	0.6946	0.6946	0.7013	0.6955	0.6941	0.6946
9	0.5384	0.5384	0.5384	0.5619	0.5365	0.5364	0.5384
10	0.6461	0.6461	0.6461	0.6463	0.6442	0.6451	0.6461
Average	0.6360	0.6360	0.6360	0.6438	0.6343	0.6334	0.6360

In terms of the consistency across different folds, it is evident that most folds exhibit a similar distribution of data, resulting in approximately the same performance. However, two folds stand out, yielding the best and worst results, identified as fold 7 and fold 9, respectively.

Upon comparing the performance results of the optimal model from the previous section with the outcomes updated after applying the K-fold algorithm, a noticeable decrease of approximately 6 percentage points in the average F-measure is observed across test folds 1 to 10. In the earlier phase of learning, 70% of the data was allocated to the training set, with 15% for validation and another 15% for testing. The data was shuffled before sampling, reducing the likelihood of non-uniform selection from various emotional classes. However, the observed decline in performance may not be solely attributed to non-uniform selection but is more likely a consequence of the mismatch in the number of collected data points from different classes. Specifically, the "angry" class constitutes the majority of the data, while the "bored" class contributes the least, with the "focused" and "neutral" classes falling in between. Consequently, the K-fold procedure unveils and accentuates this imbalance.

Bias analysis:

Introduction:

The features selected for bias analysis include age categorized into three groups: young, middle-aged, and senior, as well as gender divided into female and male subsets. Consequently, 24 label categories have been derived based on gender, age, and emotional class. During the data collection phase of the project, the dataset was labeled based on emotion. Gender and age labels were manually assigned to enhance assessment precision. Subsequently, a CSV file was created with four columns, encompassing gender, age, emotional class, and the image file name. This CSV file was prepared to facilitate proper data engineering. To identify biases, the number of data points in each of the aforementioned 24 groups was computed, revealing imbalances in the data distribution.

Bias detection results:

After counting the amount of data in each of the aforementioned 24 groups, this initial analysis has revealed a significant imbalance, particularly in the age group of seniors.

The following table displays the performance results of the best model from part two, categorized by different genders and ages. The previously mentioned prepared CSV file was utilized in this step to differentiate between various types of data.

Table 11 Performance Examination of Different Groups for the Biased Model

Group	Accuracy	P	R	F
Young	0.684	0.591	0.526	0.524
Middle-aged	0.727	0.609	0.565	0.555
Senior	0.666	0.5	0.312	0.380
Average	0.693	0.566	0.467	0.486
Male	0.687	0.595	0.552	0.545
Female	0.604	0.535	0.535	0.737
Average	0.645	0.565	0.543	0.641
Overall System AVG	0.673	0.566	0.498	0.548

From the above table, the most significant decrease is observed in the recall for the senior age group, which impacts the system's performance on this age group. Consequently, it has become evident that there is an imbalance in the senior age group. Subsequently, for further investigation, the distribution of data based on gender, age, and emotional class was extracted to add appropriate data where needed. Following the results of this process, given that the average distribution of data is around 50 images per group of age, gender, and emotional class, groups with fewer than 50 images were identified. These groups were then subjected to the step of removing imbalance to decrease the variance in the data.

Bias mitigation step:

To compensate for the lack of data in the identified groups, augmentation functionalities were applied to the images of these underrepresented groups. Within this function, there is a loop with a condition to

surpass the threshold of 50 images. Within this loop, one image from the group is randomly selected, and among the implemented augmentation functions, one is randomly chosen. The selected image is then passed to this function as input. The augmentation functions include rotating the selected image by a random angle between -30 and 30 degrees, horizontally flipping the image, zooming in by cropping the image with a random zoom factor, and shifting the image along width and height.

Comparative Performance Analysis:

After addressing the data deficiency in specific groups through data augmentation, the test and train procedures were rerun, and the performance results for these particular data groups are as follows.

Table 12 Performance Examination of Different Groups for the Unbiased Model

Group	Accuracy	P	R	F
Young	0.583	0.589	0.587	0.582
Middle-aged	0.625	0.629	0.613	0.602
Senior	0.630	0.606	0.608	0.585
Average	0.613	0.608	0.603	0.590
Male	0.585	0.598	0.590	0.583
Female	0.642	0.629	0.632	0.621
Average	0.614	0.613	0.611	0.602
Overall System AVG	0.614	0.617	0.610	0.605

Upon examining the F-score, which reflects both precision and recall, it can be acknowledged that the performance has improved, and the bottleneck observed within the senior age group has been alleviated. The improved results in the second trained model with mitigated data are expected, given the increase in data within groups previously lacking sufficient samples. The augmentation function has been applied randomly, contributing to the diversification of the data.

Confusion Matrix for The Best Model After Bias Mitigation:

The presented visuals illustrate the confusion matrices of our top-performing model across test folds one to ten. A discernible pattern emerges as the peak recognition accuracy is attained in test fold seven, contrasting with the lowest precision noted in recognizing class I, identified in fold nine. Impressively, recognition performance maintains a remarkable consistency across the remaining folds. However, a distinctive observation is the non-uniform distribution of data, particularly evident in folds seven and nine. These insights underscore the nuanced intricacies of our model's behavior, shedding light on areas of exceptional proficiency and those requiring attention, thus informing potential refinements for enhanced performance.

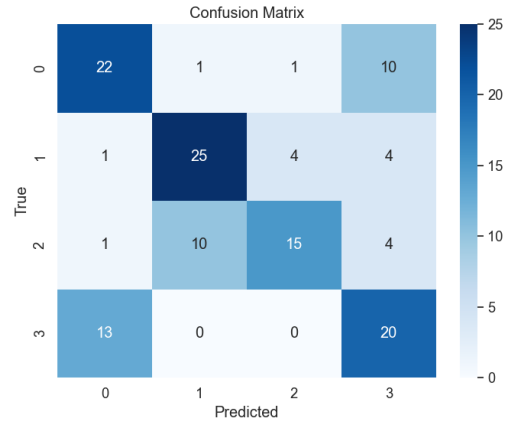

Figure 9 K-fold one

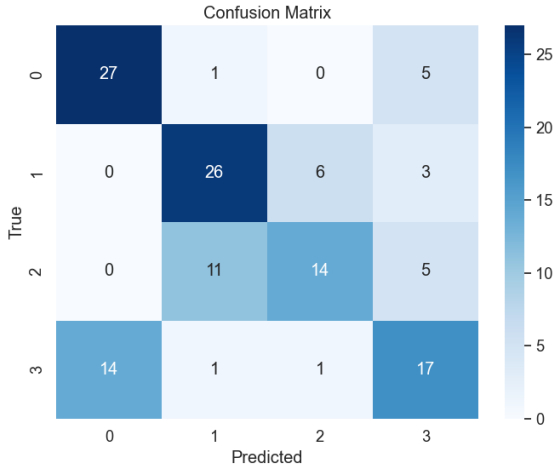

Figure 10 K-fold two

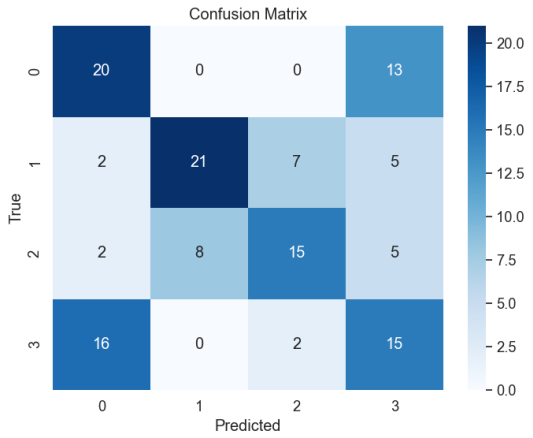

Figure 11 K-fold three

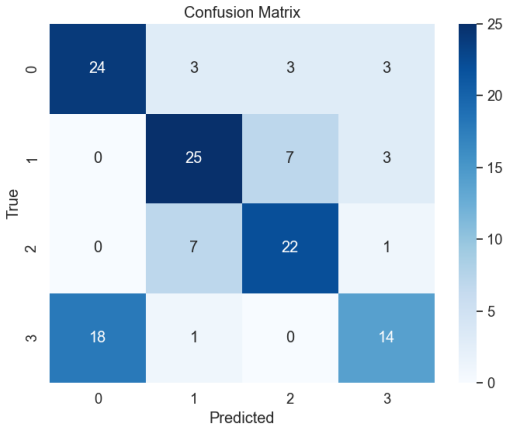

Figure 12 K-fold four

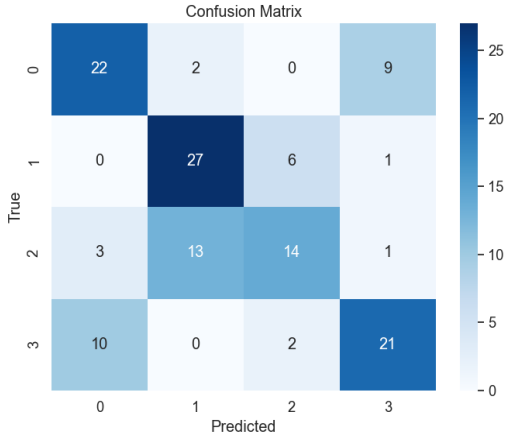

Figure 13 K-fold five

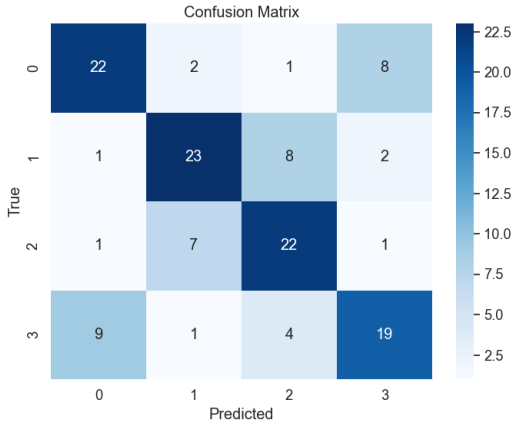

Figure 6 K-fold six

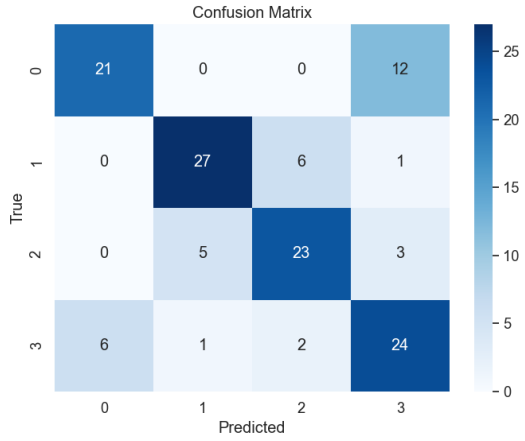

Figure 7 K-fold seven

Figure 8 K-fold eight

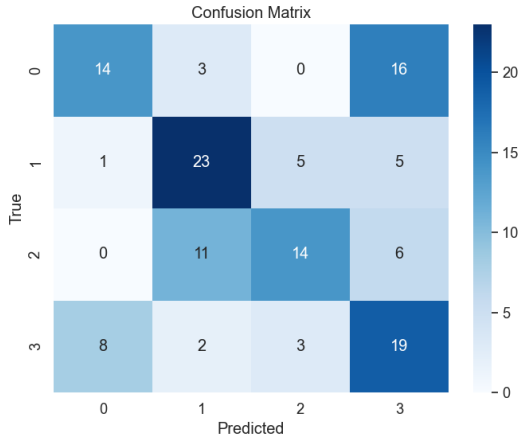

Figure 9 K-fold nine

Figure 10 K-fold ten

Reference

[1] "Pexels," [Online]. Available: https://www.pexels.com/. [Accessed: October 15, 2023].

[2] P.-L. Carrier, A. Courville, I. J. Goodfellow, M. Mirza, and Y. Bengio, "FER-2013 face database," Université de Montréal, 2013.

[3] Maulion, M. (2021, January 31). Morphological Operations: A Cleaning Technique in Image Processing. Medium. https://mattmaulion.medium.com/morphological-operations-a-cleaning-technique-in-image-processing-155baed8fcd1