

COMP 472 ARTIFICIAL INTELLIGENCE

SmartClass A.I.ssistant

Team - OK_13

Chit Chit Myet Cheal Zaw (40110140)

Nadav Friedman (40091001)

Mehdi Chitsaz (40132819)

27 June 2024

Each team member's specialization:

Data Specialist: Chit Chit Myet Cheal Zaw

Training Specialist: Nadav Friedman

Evaluation Specialist: Mehdi Chitsaz

Github repository link - https://github.com/littleSquid00/COMP472_OB_13

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

Chit Chit Myet Cheal Zaw
(40110140)

Nadav Friedman
(40091001)

Mehdi Chitsaz
(40132819)

DATASET

Overview of the existing dataset

For “Face expression recognition dataset”, there are “Train” and “Validation” folders.

The Train folder has 28,821 images and the Validation folder has 7,066 images. The thousands of images that these folders contain are pre-labeled. Each of the labels has its respective folder: angry, disgust, fear, happy, neutral, sad, and surprise. All the images are within their respective folders.

All of the images in this dataset are grayscale, frontal face shots, and 48x48px images.

For the “DAiSEE dataset”, there are “Test”, “Train” and “Validation” folders.

The Test, Train, and Validation folders contain 21, 70, and 22 subfolders respectively. Each of these subfolders represents a unique person, and the subfolder's name is thus a unique ID. Each subfolder identified by this unique ID contains hundreds of folders each representing a clip. Each clip folder contains a single video clip. The video clips are high resolution, coloured, and face the front of the person being filmed. In the background of the video, there are various objects, and sometimes other people. More information regarding the folder structure is present in the DAiSEE dataset's README.md file.

Dataset choices

We have four classes, Neutral, Angry, Engaged and Happy. We used 505 images for each class and that gives a total of 2020 images. In these 2020 images, most of them are frontal face shots.

We took Neutral, Angry and Happy images from “Face expression recognition dataset” found via the Kaggle website. We took Engaged images from the “DAiSEE” dataset.

The reason why I chose “Face expression recognition dataset” is that it is already pre-labelled Neutral, Angry and Happy, but the issue is that it has a lot of bad images with text on them and the resolution is not great. However, we think it is probably still a good choice because it prevents us from having to manually label thousands of images.

We chose the “DAiSEE” dataset because of its Engaged students' images. It is also already pre-labelled for us. The challenge was that it was videos of a different resolution and also they were not grayscale. To overcome this, we had to write code to extract frames from the videos as well as to change the colours and the size.

Provenance Information

“Face expression recognition dataset”

<https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset>

“DAiSEE” dataset

“DAiSEE is a multi-label video classification dataset comprising of 9,068 video snippets captured from 112 users for recognizing the user affective states of boredom, confusion, engagement, and frustration "in the wild". The dataset has four levels of labels namely - very low, low, high, and very high for each of the affective states, which are crowd annotated and correlated with a gold standard annotation created using a team of expert psychologists.”

https://drive.google.com/file/d/1yrk_wyhZ-c7q0Mcyyi888yIFkl_JDELi/view

“UTKFace” dataset

<https://www.kaggle.com/datasets/jangedoo/utkface-new?resource=download>

“UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc. This dataset could be used on a variety of tasks, e.g., face detection, age estimation, age progression/regression, landmark localization, etc.”

<https://susanqq.github.io/UTKFace/>

Bias Mitigation Steps and Changes to the Dataset

During Phase 3 of the project, we discovered our training data was influencing our models to exhibit strong bias (stronger performance) in favor of younger people. Specifically, this manifested itself in significantly stronger performance in properly classifying *Focused* images of younger people in comparison to seniors. To Mitigate this bias, we decided to use two strategies. Firstly, we augmented our training dataset to include more *Focused* images of older people (i.e. people belonging to the ‘Senior’ subgroup). These images were sourced from the UTKFace dataset and were transformed in the same way as all other images included in the training data were in the first phase of the project. Secondly, we looked at the *Focused* images we previously had for the ‘Senior’ subgroup and removed from the training dataset any images that we believed did not obviously represent the *Focused* category.

DATA CLEANING

The techniques and methods

- (1) First, we extracted Engaged images from videos. We got first and last frames from the video by running the command `python ExtractFrames.py`
- (2) Second, we cropped the images using Preview
- (3) Third, we turned them into grayscale by running the following on Linux (ImageMagick is required) `mogrify -colorspace Gray source.jpg`
- (4) Then, we resized the images to 48 x 48px for example by running the following on Linux (ImageMagick is required) `mogrify -resize 48x48! source.jpg`
- (5) Last, we went through the 2000+ images manually to remove those with text on them, those with sunglasses, those that seem to have been mislabeled, and those that aren't real people (e.g: cartoons)

Note: for the facial recognition dataset, only step#5 is needed.

Challenges

For DAiSEE dataset, the script for extracting images from the video was not working (e.g.: `.DS_Store` files were provided - and that broke the script). Additionally, the source/destination directories were not easily configurable, so we had to rewrite parts of it. Going through 2000+ images manually and removing those with text on them were also challenging because it took so much time.

Before-and-After Examples

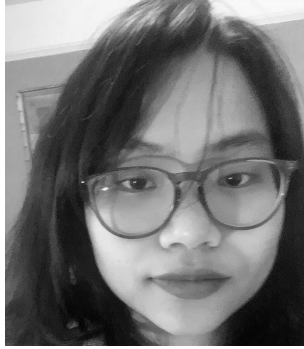
(Initial picture)



(After turning into grayscale)



(After cropping)



(After resizing it into 48x48px)



LABELING

We chose the datasets that have images which are already labelled.

226 engaged pictures were taken from the DAISEE dataset and the remainder were taken from the facial recognition dataset. DAISEE pictures were already labelled as engaged. The facial recognition one is already labelled for happy, angry and neutral classes.

Because DAISEE dataset only has 226 engaged pictures, we took neutral images from the "Face expression recognition dataset". We went through each of them and took images that looked engaged. We labelled them manually.

For the bias analysis - we chose Age and Gender bias attributes.

We took around 1,000 pictures from all datasets and labeled them manually. For age, we categorized the images into three groups: young, middle-aged, and senior. For gender, we classified the images as either male or female.

Additionally, we divided each of these age and gender groups into four emotional categories: angry, focused, happy, and neutral. We chose between 30 - 50 images for each emotional category. There are five sub-categories for age and gender so that brings up to a bit less than 1,000 images. This classification allowed us to analyze the bias across different age and gender groups, as well as their corresponding emotional expressions.

DATA VISUALIZATION

Class Distribution

To better understand the distribution of the images across our classes of facial expressions (Neutral, Angry, Happy, Engaged), we created a bar graph showing the number of images in each class. This is done to show whether any class is over or underrepresented. In our case, we have an equal number of images for each class.

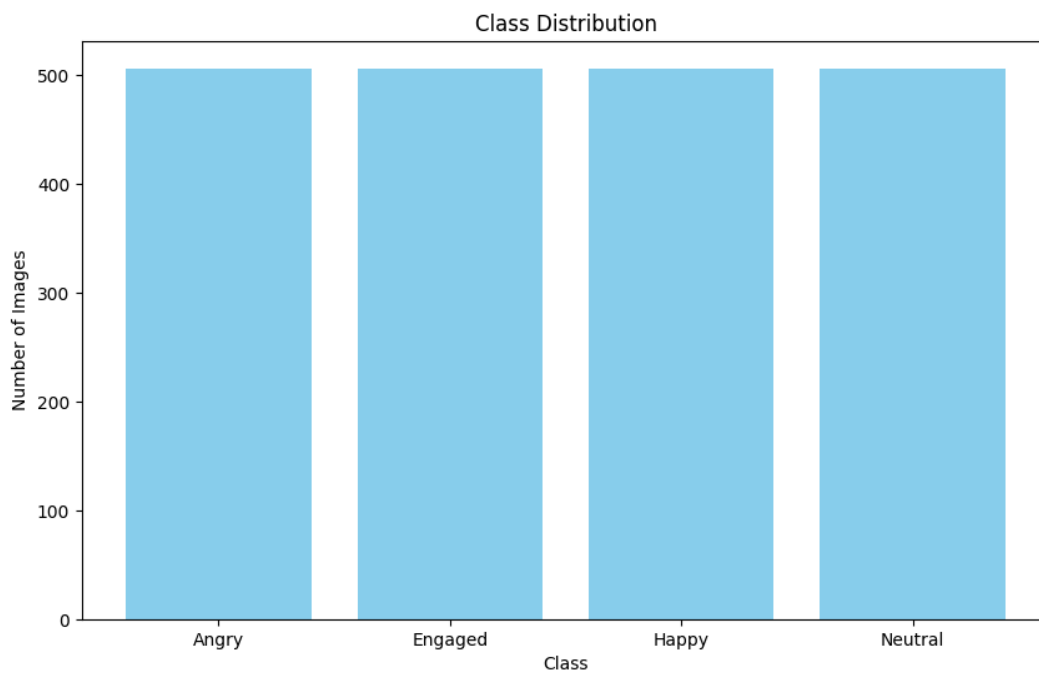


Figure: Class Distribution

The bar chart above shows the number of images in each of our four classes of facial emotions. Each class contains 508 images, meaning our data is balanced and should contribute to more equal training. Of these 508 images in each class, 3 are images of team members, and the other 505 are from the collected data.

Pixel Intensity Distribution

For each class, we plotted the aggregate pixel intensity distribution. All our images were converted to black and white (grayscale) so we plotted a single histogram for each class. These plots help us to visualize the overall brightness and contrast characteristics of the images.

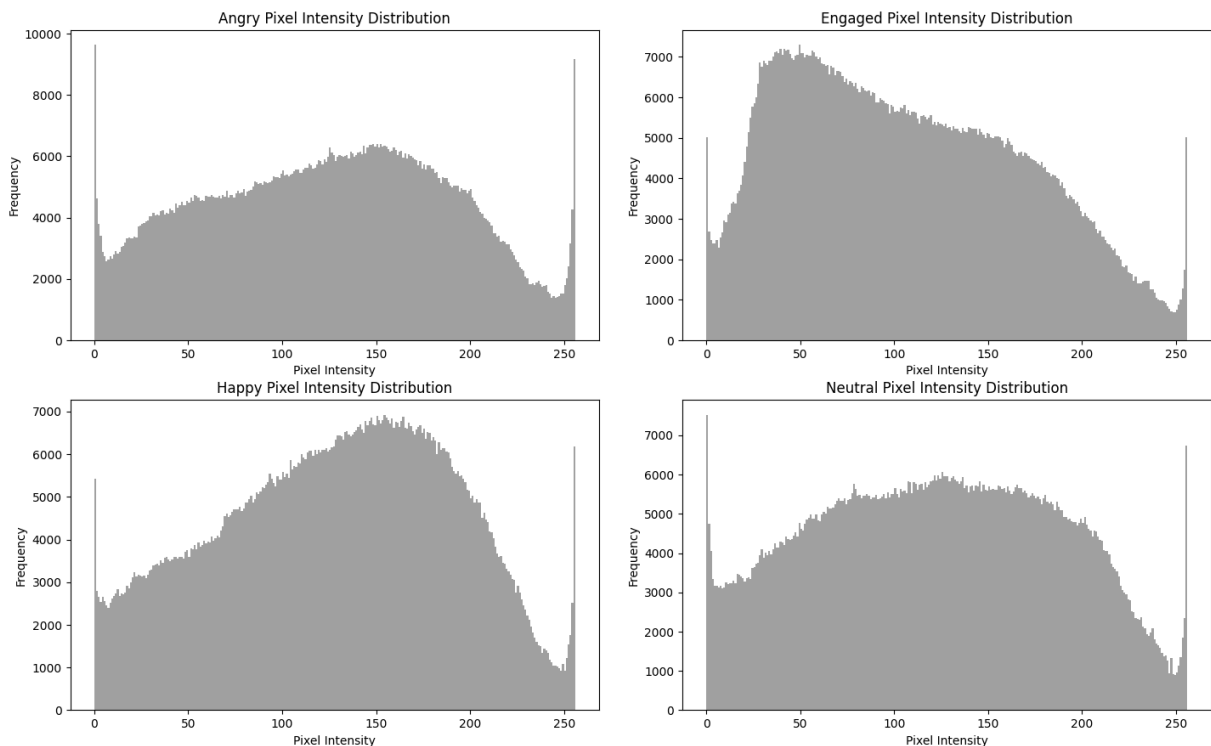


Figure: Pixel Intensity Distribution for each class

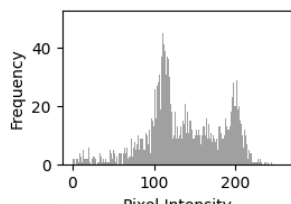
The histograms above display the pixel intensity distributions of the images in each class. They helped us understand the overall trend of brightness and contrast of the images. The fact that they are all in grayscale makes it easier to compare with different classes.

Sample Images

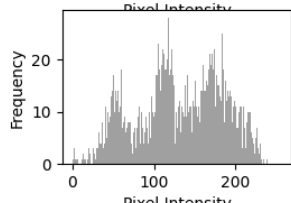
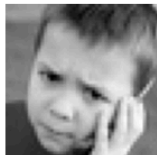
In this section, we present 15 randomly selected images from each class along with their pixel intensity histogram next to them laid in a 5 x 3 grid. The images are randomly chosen every time the DataVisualization.py script is run.

Angry

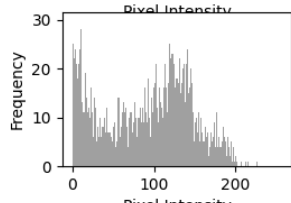
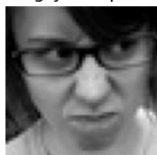
Angry sample 1



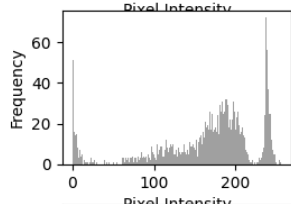
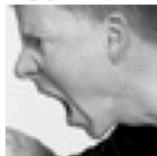
Angry sample 4



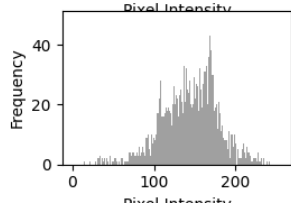
Angry sample 7



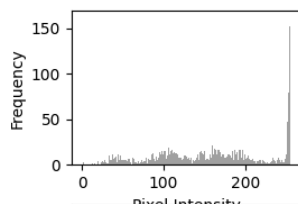
Angry sample 10



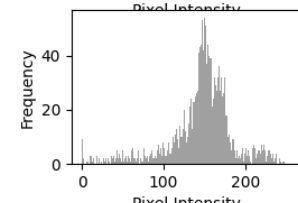
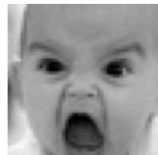
Angry sample 13



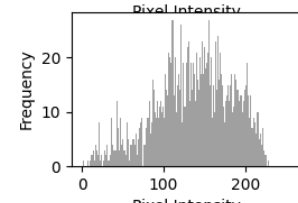
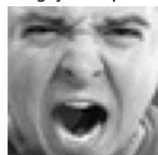
Angry sample 2



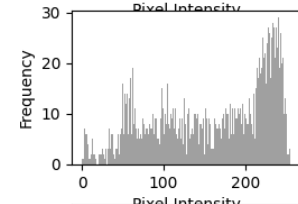
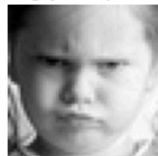
Angry sample 5



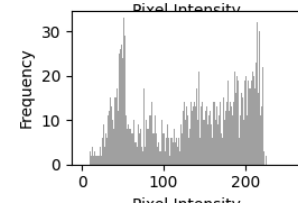
Angry sample 8



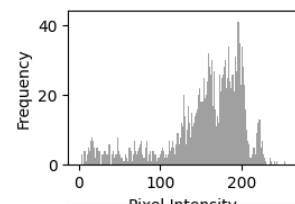
Angry sample 11



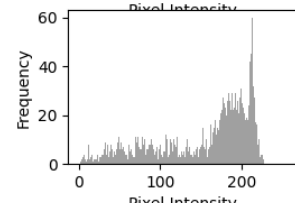
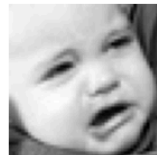
Angry sample 14



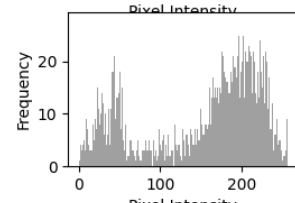
Angry sample 3



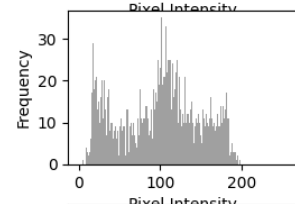
Angry sample 6



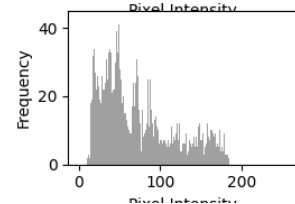
Angry sample 9



Angry sample 12

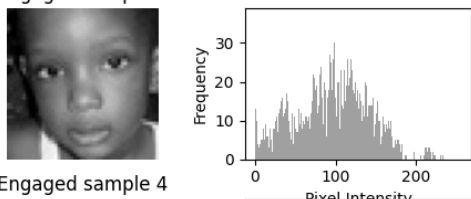


Angry sample 15

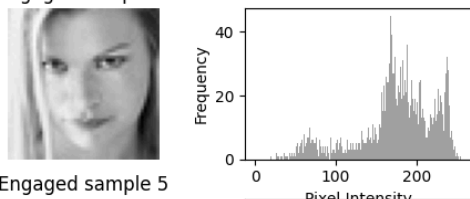


Engaged

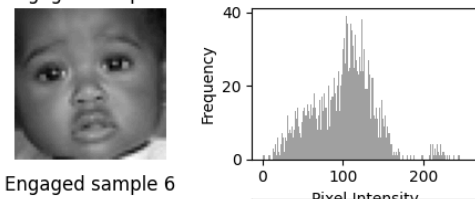
Engaged sample 1



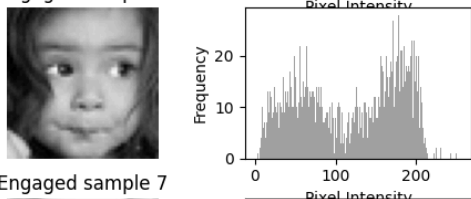
Engaged sample 2



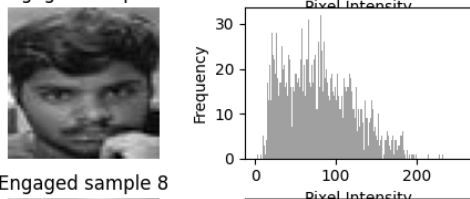
Engaged sample 3



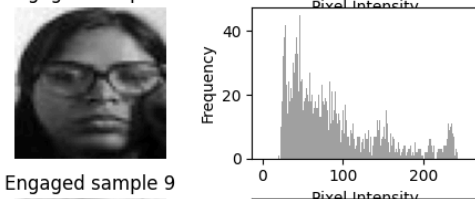
Engaged sample 4



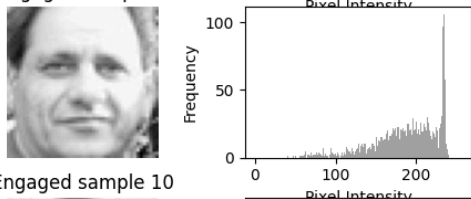
Engaged sample 5



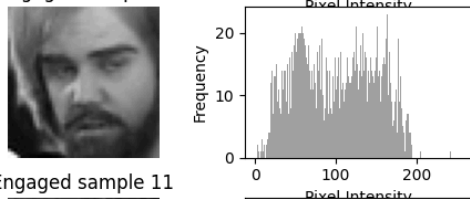
Engaged sample 6



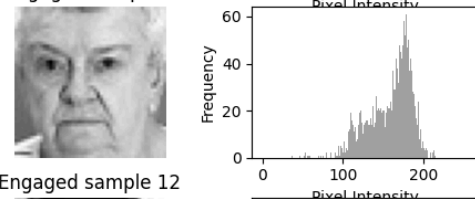
Engaged sample 7



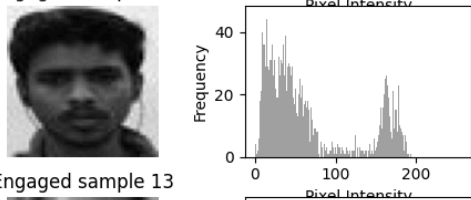
Engaged sample 8



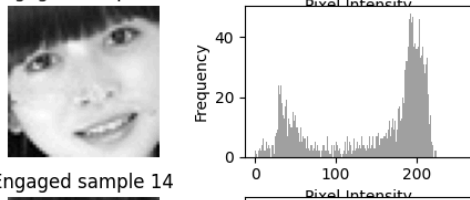
Engaged sample 9



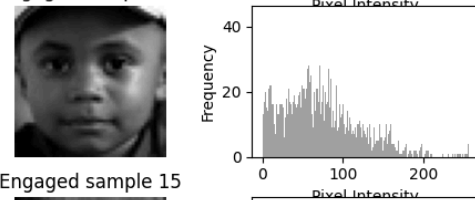
Engaged sample 10



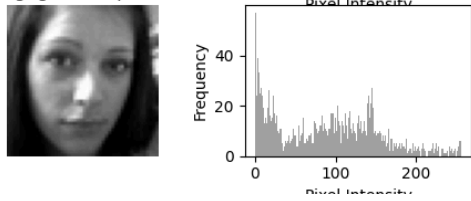
Engaged sample 11



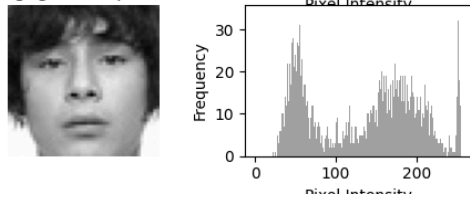
Engaged sample 12



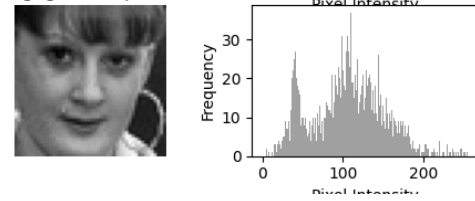
Engaged sample 13



Engaged sample 14

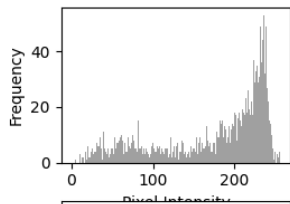


Engaged sample 15

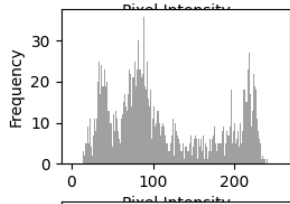


Happy

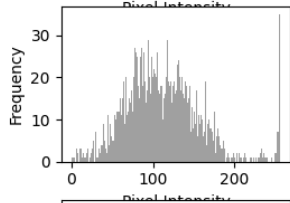
Happy sample 1



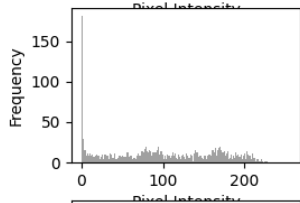
Happy sample 4



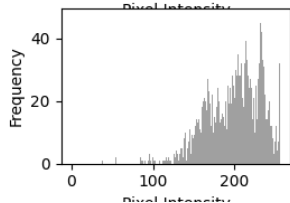
Happy sample 7



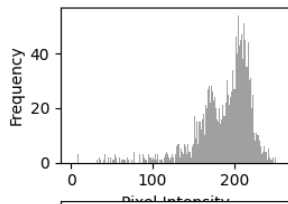
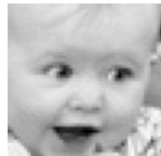
Happy sample 10



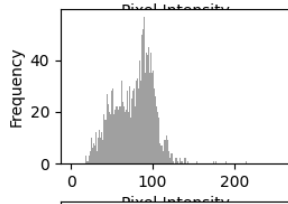
Happy sample 13



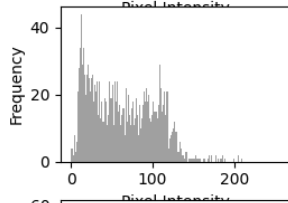
Happy sample 2



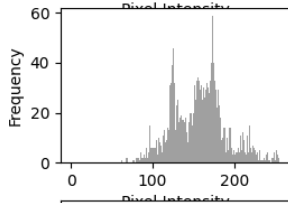
Happy sample 5



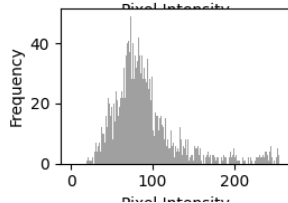
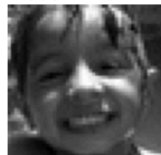
Happy sample 8



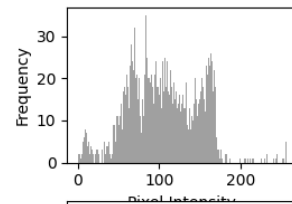
Happy sample 11



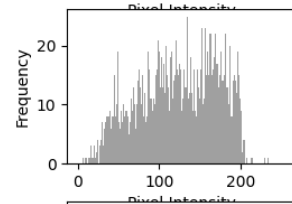
Happy sample 14



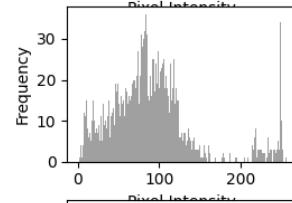
Happy sample 3



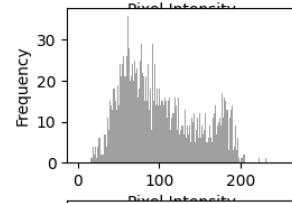
Happy sample 6



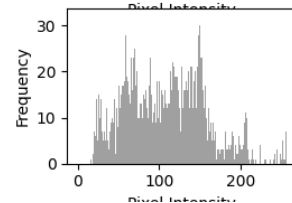
Happy sample 9



Happy sample 12

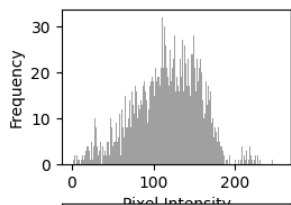
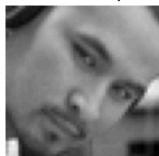


Happy sample 15

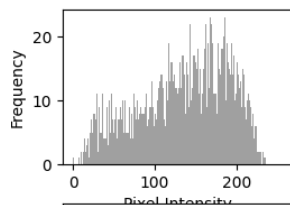


Neutral

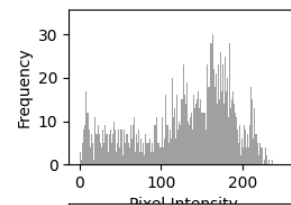
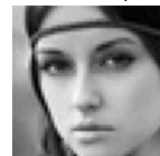
Neutral sample 1



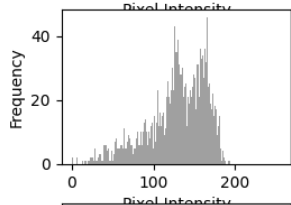
Neutral sample 2



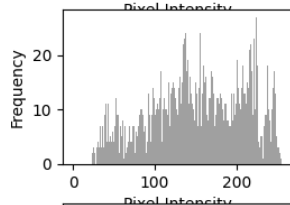
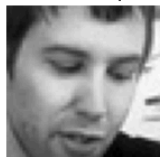
Neutral sample 3



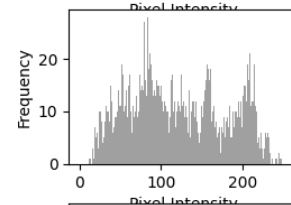
Neutral sample 4



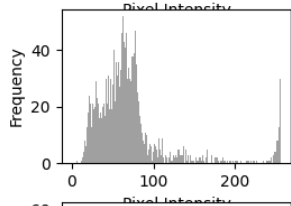
Neutral sample 5



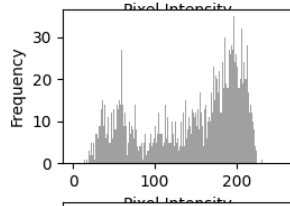
Neutral sample 6



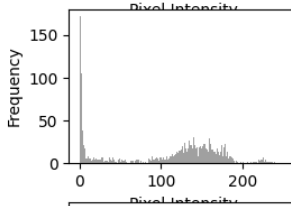
Neutral sample 7



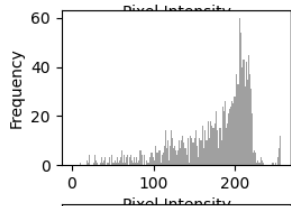
Neutral sample 8



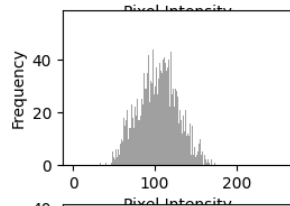
Neutral sample 9



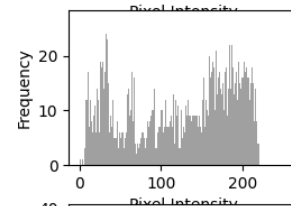
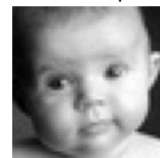
Neutral sample 10



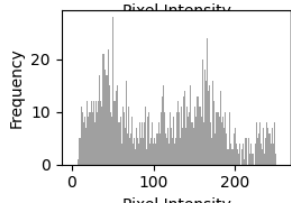
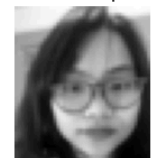
Neutral sample 11



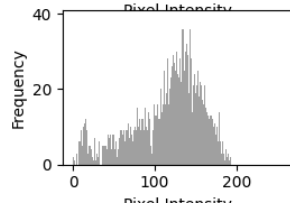
Neutral sample 12



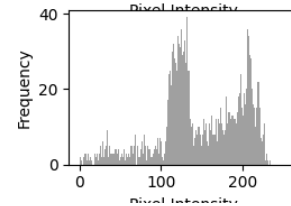
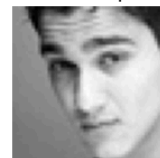
Neutral sample 13



Neutral sample 14



Neutral sample 15



The images and histograms in the grids above confirm that the preprocessing steps were done and that the images were correctly standardized and labeled

CNN ARCHITECTURE

Model Overview and Architecture Details

The main model is a Convolutional Neural Network (CNN) designed for image classification tasks. It consists of convolutional layers followed by fully connected layers.

Input Layer:

Input size: 48x48 RGB images (3 channels).

Convolutional Layer 1:

Input channels: 3 (for RGB images)

Output channels: 32

Kernel size: 3x3

Padding: 1 (to preserve spatial dimensions)

Activation Function:

ReLU (Rectified Linear Unit)

Pooling Layer 1:

Type: Max pooling

Pooling size: 2x2 Stride: 2

Convolutional Layer 2:

Input channels: 32

Output channels: 64

Kernel size: 3x3

Padding: 1 (to preserve spatial dimensions)

Activation Function:

ReLU (Rectified Linear Unit)

Pooling layer2:

Pooling size: 2x2

Stride: 2

Fully connected layer1:

Units: 512

Activation Function:

ReLU (Rectified Linear Unit)

Dropout Layer:

0.5 (for regularization to prevent overfitting)

Fully connected layer2:

Units: 512

Regularization: Dropout layer with a dropout rate of 0.5 to reduce overfitting.

Activation Function: ReLU is used after each convolutional and fully connected layer to introduce non-linearity.

MaxPooling: Used after each ReLU activation to reduce the spatial dimensions and computational complexity.

Variant 1

Variant 1 of the CNN model introduces larger kernel sizes in the convolutional layers.

Convolutional Layer 1:

Filters: 32

Kernel size: 5x5

Padding: 2

Activation function: ReLU (Rectified Linear Unit)

Convolutional Layer 2:

Filters: 64

Kernel size: 5x5

Padding: 2

Activation function: ReLU (Rectified Linear Unit)

Other layers and parameters remain the same as the main model.

Variant 2

Variant 2 adds an additional convolutional layer with smaller kernel sizes.

Convolutional Layer 1:

Filters: 32

Kernel size: 3x3

Padding: 1

Activation function: ReLU (Rectified Linear Unit)

Convolutional Layer 2:

Filters: 64

Kernel size: 3x3

Padding: 1

Activation function: ReLU (Rectified Linear Unit)

Convolutional Layer 3:

Filters: 128

Kernel size: 3x3

Padding: 1

Activation function: ReLU (Rectified Linear Unit)

Pooling layer3:

Type: Max pooling

Pooling size: 2x2

Stride: 2

Fully connected layer1:

Unit: 512

Activation function: ReLU (Rectified Linear Unit)

Dropout: 0.5

Output layer:

Units: Number of classes (4)

Activation function: Softmax

Training Process

The following details outline the training process and methodology used for training three different CNN models: the main model (CNN), CNNVariant1, and CNNVariant2.

Training Methodology:

Batch size: 64

Learning rate: 0.001

Number of epochs: 20

Patience: Early stopping is triggered if the validation loss does not improve for 5 epochs, preventing overfitting and unnecessary computation.

Transform: The input images are subjected to data augmentation using random horizontal flips. This helps in making the model more robust by providing varied representations of the input data.

Normalization: The images are normalized with mean and standard deviation of (0.5, 0.5, 0.5) for each channel, which helps in faster convergence.

Deataset: The data set is loaded from a directory structure using
`./processed_data/alldataset/xxx.jpg`

Splitting: The dataset is split into training (70%), validation (15%), and test sets (15%) using indices.

Optimization Algorithms and Techniques

Adam optimizer: The Adam optimizer `optim.Adam` is used for updating the model parameters. Adam is chosen due to its efficiency and effectiveness in handling sparse gradients and adaptive learning rates.

Mini-batch gradient descent: Training is performed using mini-batches, which allows for efficient use of GPU memory and leads to faster convergence compared to using the entire dataset for each update.

Early stopping: Early stopping is implemented to prevent overfitting. If the validation loss does not improve for 5 consecutive epochs, training is stopped early.

Model Saving: The model with the lowest validation loss is saved during training. This ensures that the best-performing model on the validation set is retained.

EVALUATION

Performance Matrix

Model	Macro			Micro			Accuracy
	P	R	F	P	R	F	
Main Model	0.6372	0.6305	0.6285	0.6305	0.6305	0.6305	0.6305
Variant 1	0.6623	0.6576	0.6572	0.6576	0.6576	0.6576	0.6576
Variant 2	0.6854	0.6712	0.6742	0.6712	0.6712	0.6712	0.6712

Accuracy: Variant 2 has the highest accuracy (0.6712), followed by Variant 1 (0.6576), and then the Main Model (0.6305). Higher accuracy indicates that Variant 2 performs better overall in correctly classifying the emotions compared to the other models.

Precision: Variant 2 also leads in both macro and weighted precision (0.6854). Variant 1 follows with a macro and weighted precision of 0.6623. The Main Model has the lowest precision (0.6372). Higher precision indicates that when the model predicts a certain emotion, it is more likely to be correct, reducing false positives. This is particularly useful in applications where false alarms need to be minimized.

Recall: Variant 2 again leads in recall (0.6712), suggesting it is better at identifying all relevant instances of emotions, reducing false negatives. Variant 1 has a recall of 0.6576, and the Main Model has the lowest recall (0.6305). High recall is crucial in contexts where missing an emotional cue could be detrimental, such as in mental health monitoring where every sign of distress should be detected.

F1-score: balance between precision and recall. Variant 1 follows with an F1-score of 0.6572 (macro and weighted). The Main Model has the lowest F1-score (0.6285). The F1-score is a balanced measure, and higher values suggest a model that performs well overall in both precision and recall.

Classification Report for Main_Model:

	precision	recall	f1-score	support
angry	0.6504	0.6568	0.6536	609
focused	0.7368	0.5526	0.6316	608
happy	0.6278	0.7796	0.6955	608
neutral	0.5338	0.5329	0.5333	608
accuracy			0.6305	2433
macro avg	0.6372	0.6305	0.6285	2433
weighted avg	0.6372	0.6305	0.6285	2433

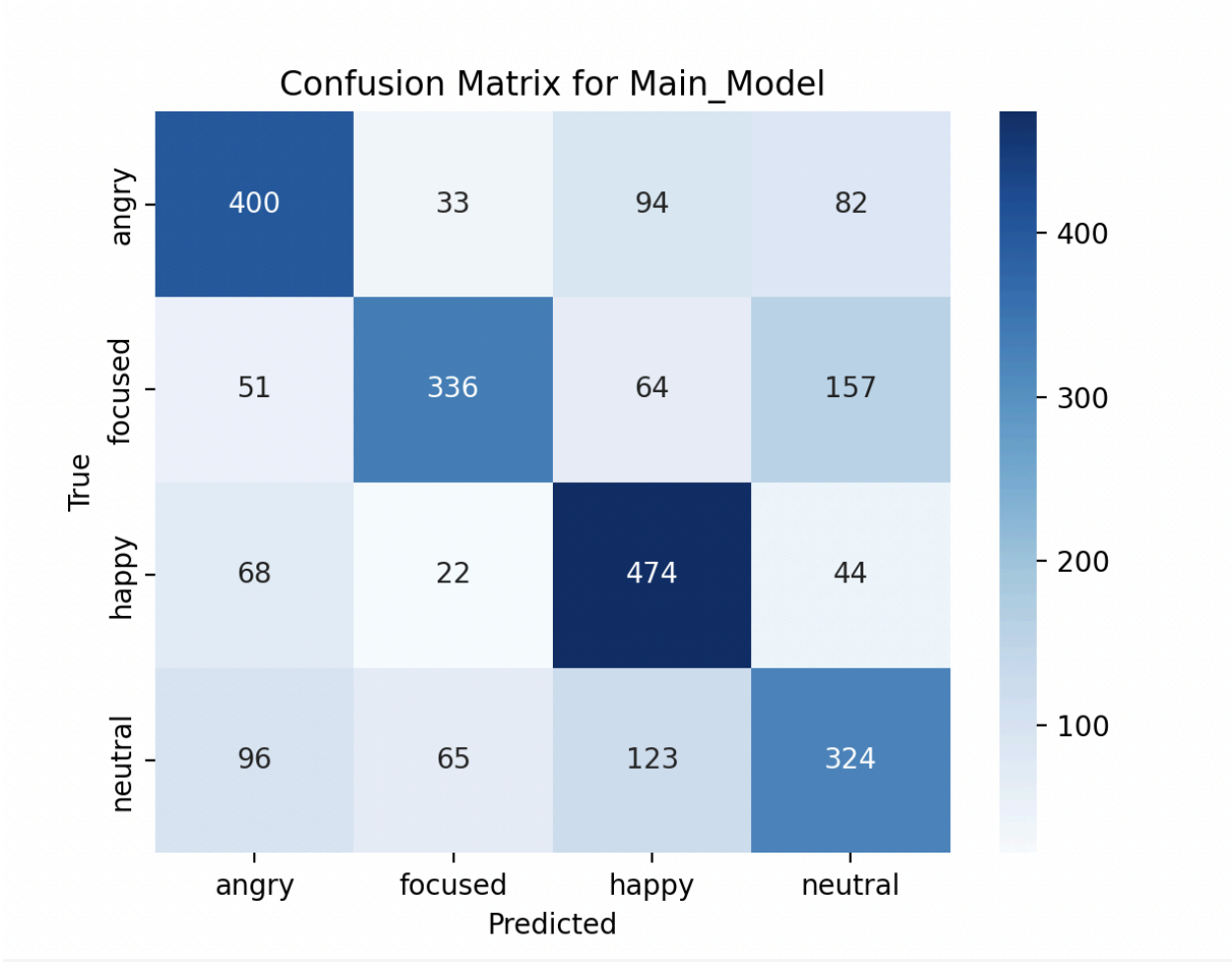
Classification Report for Variant1:

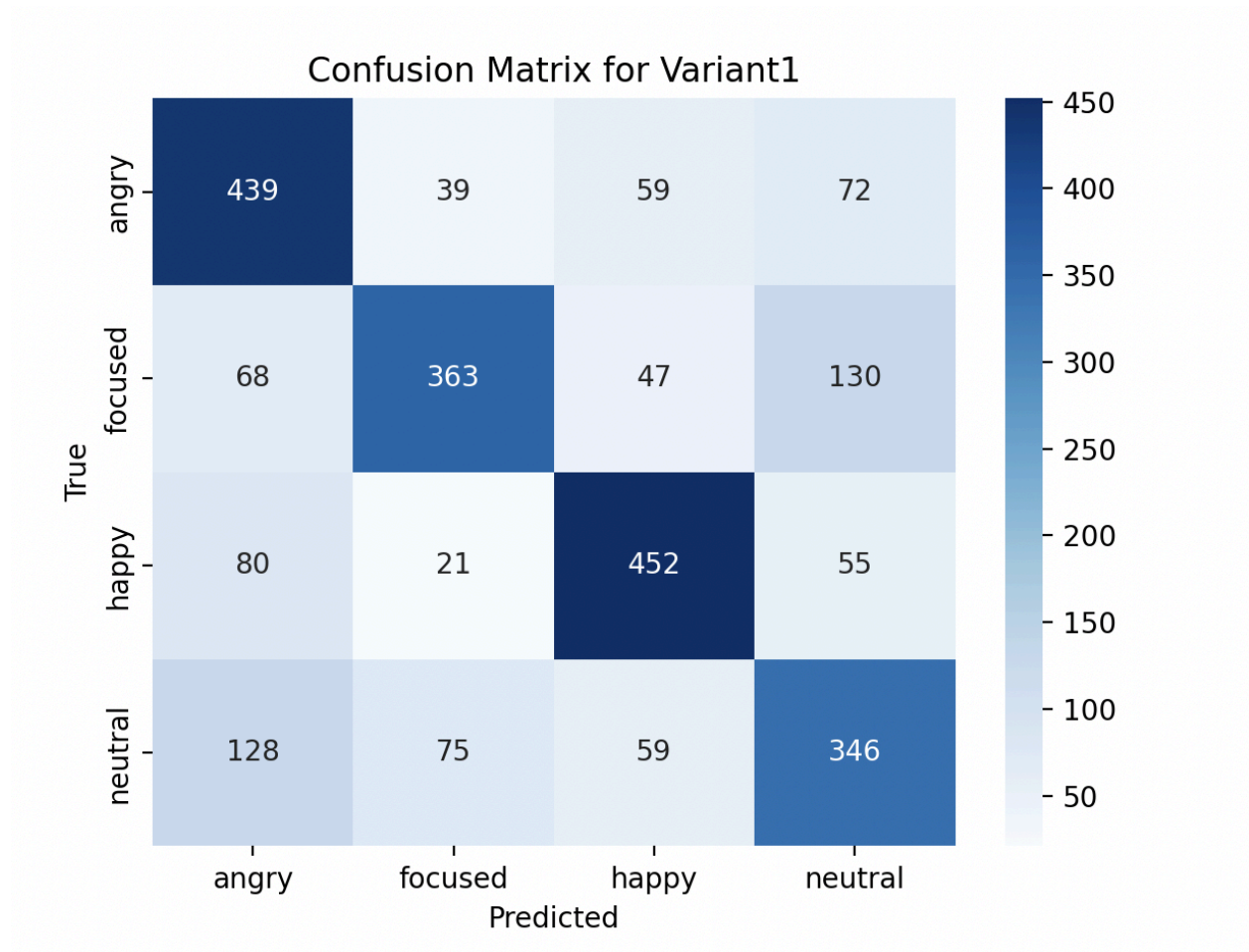
	precision	recall	f1-score	support
angry	0.6140	0.7209	0.6631	609
focused	0.7289	0.5970	0.6564	608
happy	0.7326	0.7434	0.7380	608
neutral	0.5738	0.5691	0.5714	608
accuracy			0.6576	2433
macro avg	0.6623	0.6576	0.6572	2433
weighted avg	0.6623	0.6576	0.6572	2433

Classification Report for Variant2:

	precision	recall	f1-score	support
angry	0.7165	0.6847	0.7003	609
focused	0.6322	0.7434	0.6833	608
happy	0.8497	0.6694	0.7489	608
neutral	0.5434	0.5872	0.5644	608
accuracy			0.6712	2433
macro avg	0.6854	0.6712	0.6742	2433
weighted avg	0.6854	0.6712	0.6742	2433

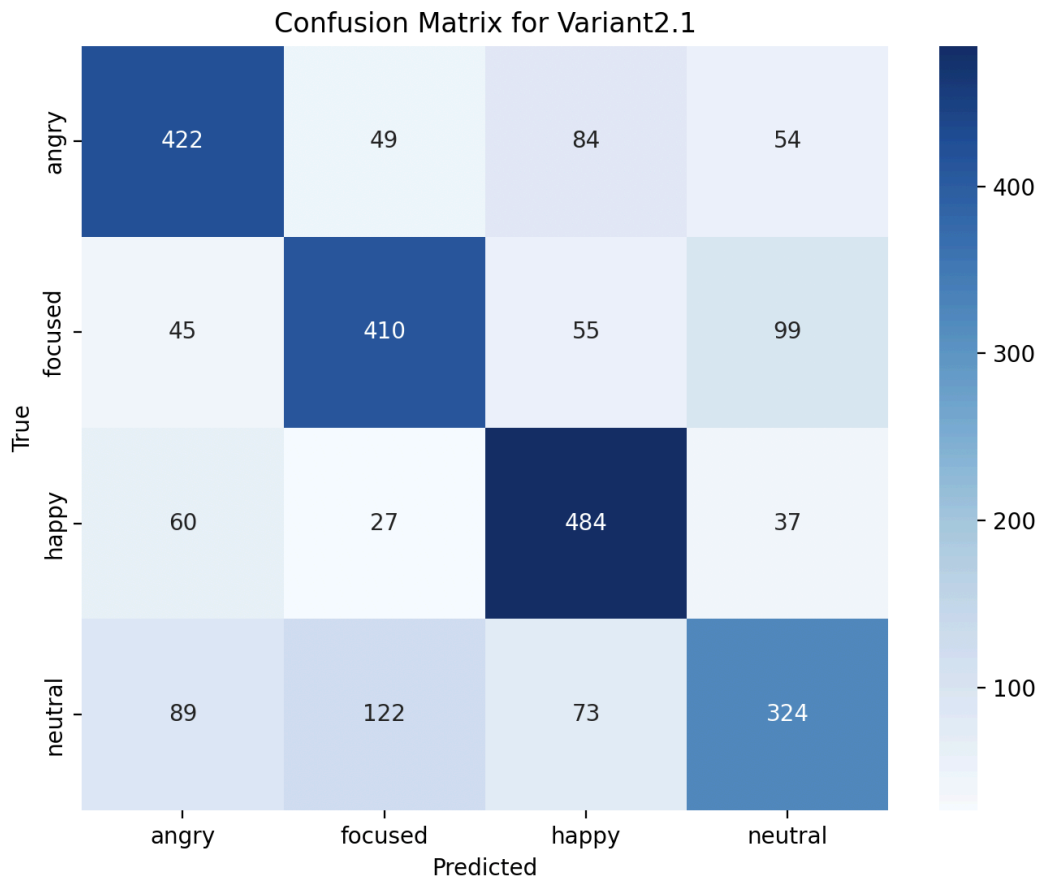
Confusion Matrix Analysis





As you can see from the confusion matrix for the main model, Variant 1 and Variant 2, Class Neutral is often confused with Class Focused, it might be due to their visual similarities or due to the dataset. Class Happy is consistently well-recognized, it could be because of distinctive and easy-to-learn features or better representation in the training data.

The confusion matrix for the saved model from our final architecture in Part III (Variant2.1) is displayed below.



Impact of Architecture Variations

Small kernels: Smaller kernels are excellent at capturing fine details and local patterns. They are effective at detecting small, intricate features like edges, corners, and fine textures, which are crucial for recognizing detailed facial features.

Large kernels: Larger kernels capture broader patterns and more contextual information within a single layer. They are useful for recognizing larger structures and more spatially extended features within the image.

The architecture changes made for variation 2 model (i.e. smaller kernel size) improved performance because a smaller kernel like 3×3 is more effective in detecting edges compared to a larger kernel because it can capture the gradient changes more precisely.

Best model

Variant2 demonstrated the highest overall accuracy (0.6712) and macro average F1-score (0.6742). It showed notable performance across most classes, with particularly high precision and F1-score for the 'happy' class. Variant2 maintained a good balance between precision and recall, leading to higher F1-scores. This balance is crucial for ensuring that the model performs consistently well across all classes. Variant2 excelled in detecting 'focused' and 'happy' emotions, which might be due to better feature extraction or more effective handling of these specific classes in the training process.

Future refinements

Perform extensive hyperparameter optimization using techniques like grid search or Bayesian optimization to fine-tune model parameters for better performance. Use data augmentation to create synthetic samples, particularly for classes that are harder to detect, to help the model generalize better.

K-fold Cross-Validation

In Part II, the training loop used a standard train/test/validation split with early stopping implemented through a patience parameter. For k-fold cross-validation in Part III, we adapted this approach as follows:

- K-Fold Setup: We used 10-fold cross-validation to split the dataset into 10 equal parts, ensuring each part was used once as a test set while the remaining nine parts served as the training and validation set.
- Training and Validation Split: Within each fold, we further split the training set into an 85% training subset and a 15% validation subset to monitor performance and implement early stopping.
- Early Stopping: Early stopping with a patience parameter was maintained as in Part II. If the validation loss did not improve for a specified number of epochs (patience), training was stopped early to prevent overfitting.
- Model Initialization: A new instance of the model was initialized at the start of training for each fold to ensure independence between folds.

We conducted 10-fold cross-validation for both our final model from Part II (referred to as the main version) and the final model developed in Part III.

Main Model from Part II: (Variant2)

Fold	Accuracy	Precision	Recall	F1-Score
0	0.5533	0.5804	0.5520	0.5478
1	0.5328	0.5170	0.5248	0.5120

2	0.5861	0.5985	0.5892	0.5923
3	0.5123	0.5110	0.5273	0.5155
4	0.5514	0.6180	0.5555	0.5458
5	0.6008	0.6013	0.6001	0.5953
6	0.5597	0.5574	0.5521	0.5533
7	0.5802	0.6262	0.5944	0.5850
8	0.5350	0.5978	0.5460	0.5348
9	0.5103	0.5964	0.5287	0.5121
Average	0.5522	0.5804	0.5570	0.5494

Final Model (Variant2.1):

Fold	Accuracy	Precision	Recall	F1-Score
0	0.5574	0.5826	0.5562	0.5569
1	0.5656	0.5918	0.5703	0.5655
2	0.5410	0.5857	0.5476	0.5466
3	0.4959	0.5016	0.5152	0.4963
4	0.5391	0.5350	0.5297	0.5128
5	0.6008	0.6322	0.5985	0.5977
6	0.5062	0.5142	0.4905	0.4955
7	0.4938	0.5177	0.5101	0.5008
8	0.5267	0.5260	0.5363	0.5203
9	0.5514	0.5615	0.5505	0.5490
Average	0.5526	0.5695	0.5571	0.5518

Significant Observations and Trends:

- **Consistency of Performance:** Across the different folds, the model's performance metrics (accuracy, precision, recall, F1-score) showed some variability, which is

expected due to different training and validation splits. However, the overall trends indicate that the model maintains a reasonable level of performance consistency.

- **Early Stopping:** Early stopping was triggered in several folds, indicating that the validation loss plateaued or increased after certain epochs, preventing overfitting and enhancing the model's generalizability.
- **Improved Metrics:** The final model (Variant2.1) showed similar performance metrics on average compared to the main model from Part II, indicating that the architectural changes or optimizations made were beneficial.

Contrast with Initial Train/Test Split Evaluation (Part II):

- **Performance Comparison:** The k-fold cross-validation results for the main model from Part II were compared to the initial train/test split evaluation. The metrics from k-fold cross-validation (accuracy: 0.5522, precision: 0.5804, recall: 0.5570, F1-score: 0.5494) were slightly different from the initial evaluation metrics.
- **Reasons for Discrepancies:** The differences in performance metrics can be attributed to the variability in data segments used for training and validation across different folds. K-fold cross-validation provides a more comprehensive evaluation by ensuring that every data point is used for both training and validation, thereby reducing the bias that can occur in a single train/test split.

Overall, k-fold cross-validation results offer a more robust and reliable estimate of the model's performance, highlighting its strengths and areas for improvement across different segments of the data.

```
Early stopping triggered
CNNVariant2.1 Fold 9 -- Accuracy: 0.5802, Precision: 0.5956, Recall: 0.5859, F1-Score: 0.5794
CNNVariant2.1 Average -- Accuracy: 0.5526, Precision: 0.5695, Recall: 0.5571, F1-Score: 0.5518
Comparison of Models:
Metric      Variant2      Variant2.1
Accuracy    0.5522        0.5526
Precision   0.5804        0.5695
Recall      0.5570        0.5571
F1_score    0.5494        0.5518
(venv) (base) yeezy@Yeezys-MacBook-Pro COMP472 0B 14 %
```

BIAS ANALYSIS

Introduction

Bias analysis for our best performing model (Variant2) was evaluated across two dimensions: Age and gender. The methodology we used was to split each of these attributes into subgroups and then to evaluate the performance of the Variant2 model on each of the subgroups. Finally, to discover any bias exhibited by the model, we compared both the model's performance on each of the subgroups within an attribute. The age attribute was divided into 'Young', 'Middle-aged', and 'Senior'. The gender attribute was divided into 'Male' and 'Female'. The

metrics used to judge the performance of the model within attribute subgroups and as a whole were accuracy, precision, recall, and F1-score.

Bias Detection Results

Variant2 Model Bias Analysis						
Attribute	Group	#Images	Accuracy	Precision	Recall	F1-Score
Age	Young	200.00	66.50%	69.11%	66.50%	66.40%
	Middle-aged	200.00	59.00%	64.44%	59.00%	53.22%
	Senior	174.00	46.55%	37.77%	43.24%	39.16%
	Total/Average	191.33	57.35%	57.11%	56.25%	52.93%
Gender	Male	200.00	51.00%	53.35%	51.00%	47.42%
	Female	200.00	56.50%	52.15%	56.50%	51.91%
	Total/Average	200.00	53.75%	52.75%	53.75%	49.67%
Overall System Total/Avg		195.67	55.55%	54.93%	55.00%	51.30%

The table above displays results of the Bias Detection step for the Variant2 model. There are several interesting observations to note. Firstly, the difference between the model's performance in the male subgroup and the female subgroup is pretty negligible indicating that the Variant2 model is unbiased across genders. However, the Variant2 model does perform differently across the age subgroups. In particular, model performance across all metrics seems to decrease as age increases. The model performs well above its average performance on younger people (66.50% accuracy), above average on middle-aged people (59% accuracy), and well below average for seniors (46.55% accuracy). Furthermore, the precision, recall and F1-score metrics for the 'Senior' subgroup were also quite poor when compared to the overall system averages for those metrics. Therefore, we concluded that this model shows strong bias favoring younger individuals.

Classification Report for Model on Age - Senior:				
	precision	recall	f1-score	support
Angry	0.4923	0.7111	0.5818	45
Focused	0.1000	0.0294	0.0455	34
Happy	0.5469	0.7000	0.6140	50
Neutral	0.3714	0.2889	0.3250	45
accuracy			0.4655	174
macro avg	0.3777	0.4324	0.3916	174
weighted avg	0.4001	0.4655	0.4199	174

Classification Report for Model on Age - Young Age:				
	precision	recall	f1-score	support
Angry	0.6053	0.4600	0.5227	50
Focused	0.8857	0.6200	0.7294	50
Happy	0.7455	0.8200	0.7810	50
Neutral	0.5278	0.7600	0.6230	50
accuracy			0.6650	200
macro avg	0.6911	0.6650	0.6640	200
weighted avg	0.6911	0.6650	0.6640	200

To discover the root of this bias, we analyzed the performance of the Variant2 model in the 'Senior' and 'Young' subgroups. The results, shown above, clearly illustrate that the Variant2 model is exceptionally poor at classifying images of seniors in the *Focused* class when compared to classifying *Focused* images of younger people. The reason for this discrepancy in model performance makes intuitive sense. Students are the people that display focused and/or engaged faces the most in their everyday life. Furthermore, students are almost always young people. This likely means our Variant2 model training data did not possess enough images of seniors in the *Focused* class.

Bias Mitigation Steps

As mentioned above, we came to the conclusion that our Variant2 model exhibited strong bias (stronger performance) in favor of younger people. Specifically, this manifested itself in significantly stronger performance in properly classifying *Focused* images of younger people in comparison to seniors. To Mitigate this bias, we decided to use two strategies. Firstly, we augmented our training dataset to include more *Focused* images of older people (i.e. people belonging to the 'Senior' subgroup). These images were sourced from the UTKFace dataset and were transformed in the same way as all other images included in the training data were in the first phase of the project. Secondly, we looked at the *Focused* images we previously had for the 'Senior' subgroup and removed from the training dataset any images that we believed did not obviously represent the *Focused* category.

Comparative Performance Analysis

After making the aforementioned changes to the training dataset, we trained the Variant2 model on this new training dataset. The resulting model was named Variant2.1. We then assessed the bias of Variant2.1 using the same process we did for Variant2. The results of this bias detection process are displayed below.

Variant2.1 Model Bias Analysis						
Attribute	Group	#Images	Accuracy	Precision	Recall	F1-Score
Age	Young	200	60.00%	59.75%	60.00%	59.37%
	Middle-aged	200	58.50%	57.71%	58.50%	57.80%
	Senior	174	53.45%	51.44%	52.76%	51.20%
	Total/Average	191.33	57.32%	56.30%	57.09%	56.12%
Gender	Male	200	52.50%	53.06%	52.50%	52.40%
	Female	200	56.50%	56.21%	56.50%	56.08%
	Total/Average	200	54.50%	54.64%	54.50%	54.24%
Overall System Total/Avg		195.67	55.91%	55.47%	55.79%	55.18%

In comparison to the bias results obtained by Variant2, Variant2.1 displays a much more balanced performance for all metrics across all age subgroups. In particular, the performance of Variant2.1 for the ‘Senior’ subgroup far exceeds the performance of Variant2 for that subgroup. Importantly, the performance of Variant2.1 on the ‘Senior’ subgroup is now much more in line with the other age subgroups. This indicates that the steps we took to mitigate the bias of Variant2 were successful.

In addition, Variant2.1 maintained the unbiased results for the gender subgroup that Variant2 possessed. In terms of overall system performance, all metric results are extremely similar at around 55%. These results were all negligibly better than the overall results for Variant2. However, since Variant2.1 is a less biased model than Variant2 it is ultimately the more attractive model.

REFERENCES

A. Gupta et al.(n.d.), "DAiSEE dataset" Source. [Online]. Available:
https://drive.google.com/file/d/1yrk_wyhZ-c7q0Mcyi888yIFkI_JDELI/view

J.Hunter et al. (n.d), "Matplotlib Documentation" Source. [Online]. Available:
<https://matplotlib.org/stable/contents.html>

J.Oheix. (2019), "The facial expression recognition dataset" Source.[Online]. Available:
<https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset>

"OpenCV Documentation" Source. [Online]. Available: <https://docs.opencv.org/master/>

S. Subedi. (2018), "UTKFace" Source. [Online]. Available:
<https://www.kaggle.com/datasets/jangedoo/utkface-new?resource=download>