COMP 472 Project Part 2

Summer 2024

Due Date: June 15, 2024

Submitted by Group AK-2

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

Name	**ID**	**Role**	**Signature**	**Date**
Nadia Beauregard	40128655	Training Specialist & Developer	Nadia Beauregard	**14/05/2024**
Marina Girgis	40168639	Evaluation Specialist & Developer	Marina	**15/06/2024**
Karina Sanchez-Duran	40189860	Data Specialist & Developer	Karina SD	**14/05/2024**

Link to Github Repo:

https://github.com/KarinaSandur/COMP472_SmartClass_A.I.ssistant

Originality Forms

Faculty of Engineering and Computer Science
Expectations of Originality

This form sets out the requirements for originality for work submitted by students in the Faculty of Engineering and Computer Science. Submissions such as assignments, lab reports, project reports, computer programs and take-home exams must conform to the requirements stated on this form and to the Academic Code of Conduct. The course outline may stipulate additional requirements for the course.

1. Your submissions must be your own original work. Group submissions must be the original work of the students in the group.
2. Direct quotations must not exceed 5% of the content of a report, must be enclosed in quotation marks, and must be attributed to the source by a numerical reference citation[1]. Note that engineering reports rarely contain direct quotations.
3. Material paraphrased or taken from a source must be attributed to the source by a numerical reference citation.
4. Text that is inserted from a web site must be enclosed in quotation marks and attributed to the web site by numerical reference citation.
5. Drawings, diagrams, photos, maps or other visual material taken from a source must be attributed to that source by a numerical reference citation.
6. No part of any assignment, lab report or project report submitted for this course can be submitted for any other course.
7. In preparing your submissions, the work of other past or present students cannot be consulted, used, copied, paraphrased or relied upon in any manner whatsoever.
8. Your submissions must consist entirely of your own or your group's ideas, observations, calculations, information and conclusions, except for statements attributed to sources by numerical citation.
9. Your submissions cannot be edited or revised by any other student.
10. For lab reports, the data must be obtained from your own or your lab group's experimental work.
11. For software, the code must be composed by you or by the group submitting the work, except for code that is attributed to its sources by numerical reference.

You must write one of the following statements on each piece of work that you submit:
For individual work: **"I certify that this submission is my original work and meets the Faculty's Expectations of Originality",** with your signature, I.D. #, and the date.
For group work: **"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality",** with the signatures and I.D. #s of all the team members and the date.

A signed copy of this form must be submitted to the instructor at the beginning of the semester in each course.

I certify that I have read the requirements set out on this form, and that I am aware of these requirements. I certify that all the work I will submit for this course will comply with these requirements and with additional requirements stated in the course outline.

Course Number:	COMP 472	Instructor:	Dr. René Witte
Name:	Karina Sanchez-Duran	I.D. #	40189860
Signature:	KarinaSD	Date:	June 14, 2024

[1] Rules for reference citation can be found in "Form and Style" by Patrich MacDonagh and Jack Bordan, fourth edition, May, 2000, available at http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf.
Approved by the ENCS Faculty Council February 10, 2012

Faculty of Engineering and Computer Science
Expectations of Originality

This form sets out the requirements for originality for work submitted by students in the Faculty of Engineering and Computer Science. Submissions such as assignments, lab reports, project reports, computer programs and take-home exams must conform to the requirements stated on this form and to the Academic Code of Conduct. The course outline may stipulate additional requirements for the course.

1. Your submissions must be your own original work. Group submissions must be the original work of the students in the group.
2. Direct quotations must not exceed 5% of the content of a report, must be enclosed in quotation marks, and must be attributed to the source by a numerical reference citation[1]. Note that engineering reports rarely contain direct quotations.
3. Material paraphrased or taken from a source must be attributed to the source by a numerical reference citation.
4. Text that is inserted from a web site must be enclosed in quotation marks and attributed to the web site by numerical reference citation.
5. Drawings, diagrams, photos, maps or other visual material taken from a source must be attributed to that source by a numerical reference citation.
6. No part of any assignment, lab report or project report submitted for this course can be submitted for any other course.
7. In preparing your submissions, the work of other past or present students cannot be consulted, used, copied, paraphrased or relied upon in any manner whatsoever.
8. Your submissions must consist entirely of your own or your group's ideas, observations, calculations, information and conclusions, except for statements attributed to sources by numerical citation.
9. Your submissions cannot be edited or revised by any other student.
10. For lab reports, the data must be obtained from your own or your lab group's experimental work.
11. For software, the code must be composed by you or by the group submitting the work, except for code that is attributed to its sources by numerical reference.

You must write one of the following statements on each piece of work that you submit:
For individual work: **"I certify that this submission is my original work and meets the Faculty's Expectations of Originality"**, with your signature, I.D. #, and the date.
For group work: **"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality"**, with the signatures and I.D. #s of all the team members and the date.

A signed copy of this form must be submitted to the instructor at the beginning of the semester in each course.

I certify that I have read the requirements set out on this form, and that I am aware of these requirements. I certify that all the work I will submit for this course will comply with these requirements and with additional requirements stated in the course outline.

Course Number:	COMP 472	Instructor:	Dr. René Witte
Name:	Nadia Beauregard	I.D. #	40128655
Signature:	Nadia Beauregard	Date:	June 14, 2024

[1] Rules for reference citation can be found in "Form and Style" by Patrich MacDonagh and Jack Bordan, fourth edition, May, 2000, available at http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf.
Approved by the ENCS Faculty Council February 10, 2012

Faculty of Engineering and Computer Science
Expectations of Originality

This form sets out the requirements for originality for work submitted by students in the Faculty of Engineering and Computer Science. Submissions such as assignments, lab reports, project reports, computer programs and take-home exams must conform to the requirements stated on this form and to the Academic Code of Conduct. The course outline may stipulate additional requirements for the course.

1. Your submissions must be your own original work. Group submissions must be the original work of the students in the group.
2. Direct quotations must not exceed 5% of the content of a report, must be enclosed in quotation marks, and must be attributed to the source by a numerical reference citation[1]. Note that engineering reports rarely contain direct quotations.
3. Material paraphrased or taken from a source must be attributed to the source by a numerical reference citation.
4. Text that is inserted from a web site must be enclosed in quotation marks and attributed to the web site by numerical reference citation.
5. Drawings, diagrams, photos, maps or other visual material taken from a source must be attributed to that source by a numerical reference citation.
6. No part of any assignment, lab report or project report submitted for this course can be submitted for any other course.
7. In preparing your submissions, the work of other past or present students cannot be consulted, used, copied, paraphrased or relied upon in any manner whatsoever.
8. Your submissions must consist entirely of your own or your group's ideas, observations, calculations, information and conclusions, except for statements attributed to sources by numerical citation.
9. Your submissions cannot be edited or revised by any other student.
10. For lab reports, the data must be obtained from your own or your lab group's experimental work.
11. For software, the code must be composed by you or by the group submitting the work, except for code that is attributed to its sources by numerical reference.

You must write one of the following statements on each piece of work that you submit:
For individual work: **"I certify that this submission is my original work and meets the Faculty's Expectations of Originality"**, with your signature, I.D. #, and the date.
For group work: **"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality"**, with the signatures and I.D. #s of all the team members and the date.

A signed copy of this form must be submitted to the instructor at the beginning of the semester in each course.

I certify that I have read the requirements set out on this form, and that I am aware of these requirements. I certify that all the work I will submit for this course will comply with these requirements and with additional requirements stated in the course outline.

Course Number:	Comp 472	Instructor:	Dr. René Witte
Name:	Marina Girgis	I.D. #	40168639
Signature:	Marina	Date:	2024-06-15

[1] Rules for reference citation can be found in "Form and Style" by Patrich MacDonagh and Jack Bordan, fourth edition, May, 2000, available at http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf.
Approved by the ENCS Faculty Council February 10, 2012

Dataset

Dataset Overview

Overview of the Angry dataset:
The total number of images is 670, there are 2 classes: the training images with 442 images and the testing images with 228 images. The images chosen are mostly frontal face shots, but there are some side face images also.

Overview of the focused dataset:
The total number of images is 504, there are 2 classes: the training images with 403 images and the testing images with 101 images. The images chosen are mostly frontal face shots, but there are some side face images also with different backgrounds. Some of the images have a little part of another person's face. For the sake of variety, we chose people from different gender and age groups.

Overview of the happy dataset:
The total number of images is 516, there are 2 classes: the training images with 405 images and the testing images with 111 images. The images chosen are mostly frontal face shots, some people have more happier reactions than others, with different happy images, we cover a lot of smiles types.

Overview of the neutral dataset:
The total number of images is 559, there are 2 classes: the training images with 421 images and the testing images with 138 images. There are different backgrounds, whether it is in nature or coloured, we made sure that the images chosen have the full face pointed to the camera to have a full coverage of the neutral expressions.

Data Collection

1) **Facial Expressions Training:**

Source: Kaggle [1]

Description: An extensive collection of facial images labeled with different affective states (emotions) is called AffectNet. It has been specially processed for neural network applications, with an emphasis on real-world limitations like data quality and memory consumption. Every image has been scaled to 96 × 96 pixels, which is the fixed resolution. This guarantees consistency in the dimensions of the images, which is necessary for neural network inference and training.

Relevance: For the purpose of training and assessing machine learning models, particularly deep learning models like convolutional neural networks (CNNs), AffectNet offers an abundance of resources. It can be used by researchers to test out novel algorithms and methods in emotional computing and computer vision.

Difficulties: Although 96x96 pixels is a workable fixed resolution for memory constraints, it may be insufficient for some applications that need finer details. This may hinder models' ability to identify nuanced facial expressions.

2) **Selected Pictures from the Web for "focused" images:**

Source: A Variety of Websites [2]-[4]

Description: We carefully picked and downloaded pictures from several educational websites, online learning platforms, and stock photo sources to portray attentive and engaged students. These pictures show pupils engaged in learning activities and paying close attention, demonstrating their level of involvement.

Relevance: The model must be trained using these images in order to identify the engaged/focused class, which is necessary for our project. Usually, standard facial expression datasets do not have a good representation of this class.

Difficulties: There are a number of difficulties associated with manually sourcing these photos, such as guaranteeing demographic variety and preserving consistency in image quality. Furthermore, each photograph must be labeled and verified to guarantee that it truly depicts a focused and engaged condition.

Dataset	Link	License	Rating
Facial Expressions Training Data [1]	https://www.kaggle.com/datasets/noamsegal/affectnet-training-data?select=anger	Attribution-NonCommercial-ShareAlike	10/10 for usability
Pexels focused images [2]	https://www.pexels.com/search/focused/	Free to use	N/A
Freepik [3]	https://www.freepik.com/search?ai=excluded&format=search&last_filter=query&last_value=&query=&type=photo	Free to use	N/A
Unsplash [4]	https://unsplash.com/	Free to use	N/A

Table 1: Data Collection Information

Data Cleaning

One technique used to clean the data was to convert all the images in the entire dataset to the same format. The team chose to convert all images to JPEG to maximize space and because most of the images taken from existing datasets were already in JPEG format. To complete this conversion, a Python script called PNGtoJPEGConverter.py was created. The script takes a folder path as input and then converts all PNG files in the folder to JPEG [5].

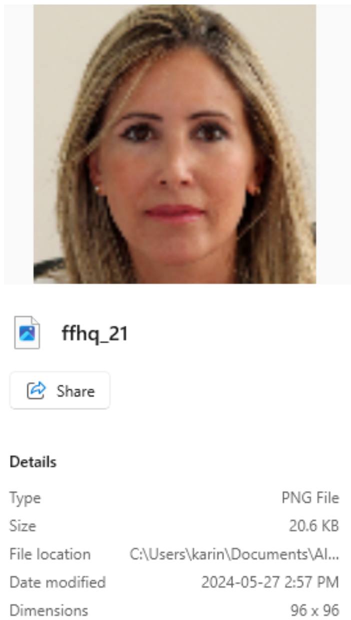
Another technique used to standardize the dataset was to resize the images so they all have the exact same dimensions. To resize all the images in the dataset, a Python script called resizeImages.py was created that takes a folder path as input and then resizes all JPEG images in the folder to 150 x 150 [6]. Most images taken from the Kaggle dataset were 96 x 96 and most images taken from other sources were around 500 x 500. The decision to resize all the images to 150 x 150 was made to enlarge the 96 x 96 images to a more reasonable size without allowing them to become too pixelated. It was also thought that shrinking the images from around 500 x 500 to 150 x 150 wouldn't affect image quality that much.

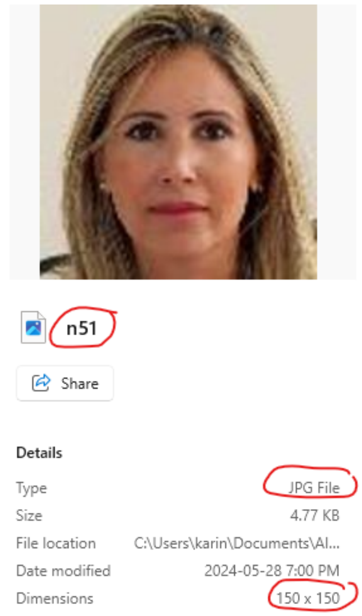
To standardize the data, all images in a folder were renamed to a consistent format. To rename all the images in a folder, a Python script called renameImages.py was created [7]. The Python script takes a folder path and name as input and renames all the images with the inputted name as the prefix and a number as the suffix in increasing order (e.g. angry1, angry2, etc..).

Another method we used to clean the dataset was to manually inspect every image in each class. Every team member manually looked over roughly 1700 pictures in the dataset and cropped out images to have a better view of the person's face and/or to remove distracting things in the background that could confuse the AI.

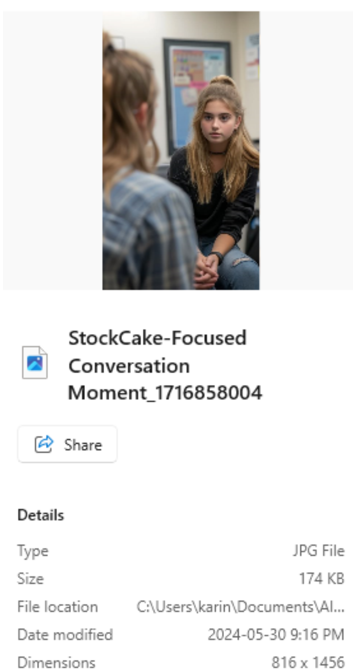
The main difficulty we encountered was that the datasets from Kaggle were too big (approximately 3000-5000 images per class). Thus, the datasets were too big for us to manually inspect every image as planned (to crop them and ensure the facial expression matched the label) in the time given and the files were too big to upload to GitHub. Thus, the team decided to use only a subset of the data found on Kaggle, convert all the images to JPEG (as mentioned above) and place all the files in zip folders to save time and space.

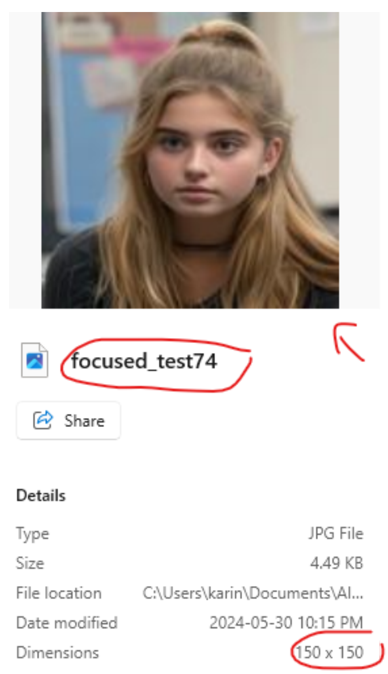
See examples of data cleaning below:

<table>
<tr><th>‘Neutral’ Image Before Cleaning</th><th>‘Neutral’ Image After Cleaning</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

<table>
<tr><th>‘Focused’ Image Before Cleaning</th><th>‘Focused’ Image After Cleaning</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Labeling

The images for the neutral class, the angry class and the happy class were mostly taken from the Kaggle dataset mentioned above and were already placed in folders labeled neutral, angry and happy, respectively.

There was no existing dataset for a ‘focused facial expression’ or at least none the team could find. Thus, team members each found approximately 170 images of people with a focused facial expression from a variety of sources. The focused images found amongst the team were combined and placed in a zip folder labeled focused.

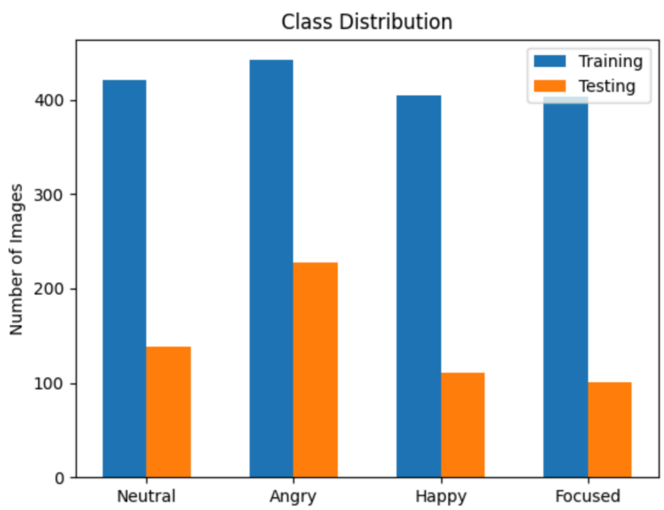
The images for ‘focused’ were handpicked from a variety of sources by each teammate. Thus, the team was certain that the images in the ‘focused’ folder were properly labeled (i.e: placed in the correct folder). However, the images for neutral, angry and happy were taken from Kaggle and the team decided it was necessary to verify each image was labeled correctly (placed in the right folder).

Given the time limitation, the team decided to only take a subset of the data found on Kaggle and thus took approximately 1000 images for neutral, 1000 images for angry and 1000 images for happy. Each team member inspected the images in one of the classes in order to ensure the images were properly labeled. In essence, team members checked that the images in the ‘angry’ folder were in fact all angry, all the images in the ‘happy’ folder were all happy and all the images in the ‘neutral’ folder were all neutral. Otherwise, the image was simply removed from the folder. In the end, each class contained around 500 - 600 images. Afterwards, the images in each class were subdivided into three: training images, validation images and testing images. The distribution for the classes can be seen in the *Class Distribution* graph below.

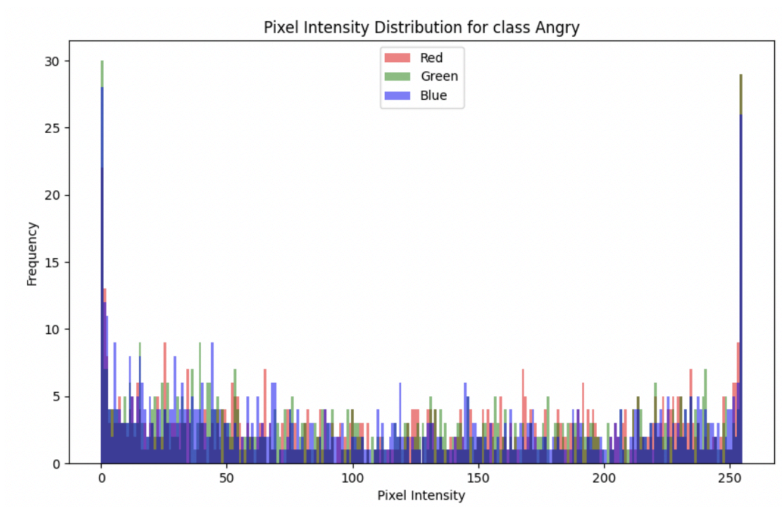
Dataset Visualization

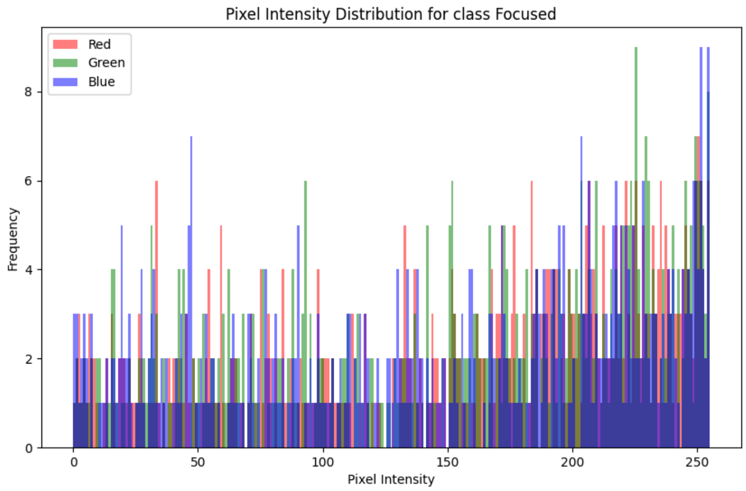
The following graphs were created using Matplotlib [8].

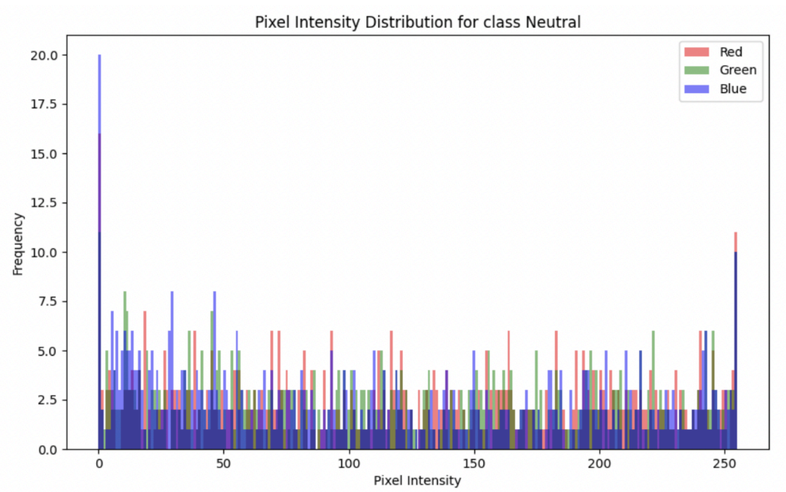
1) Class Distribution:

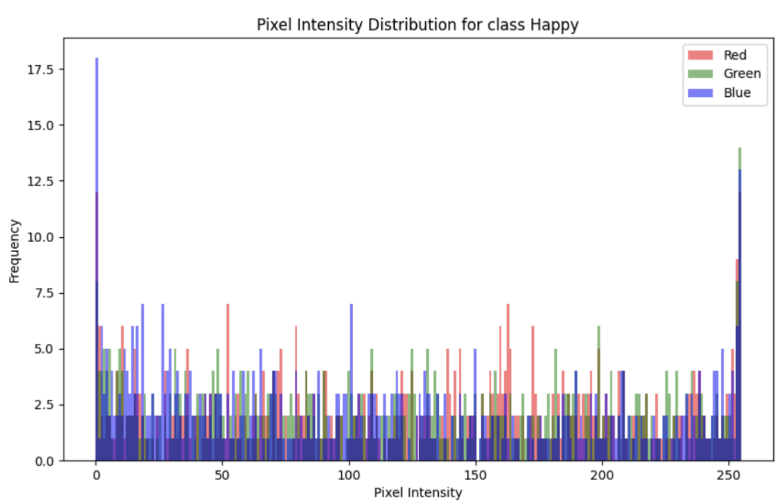


2) Pixel Intensity Distribution:



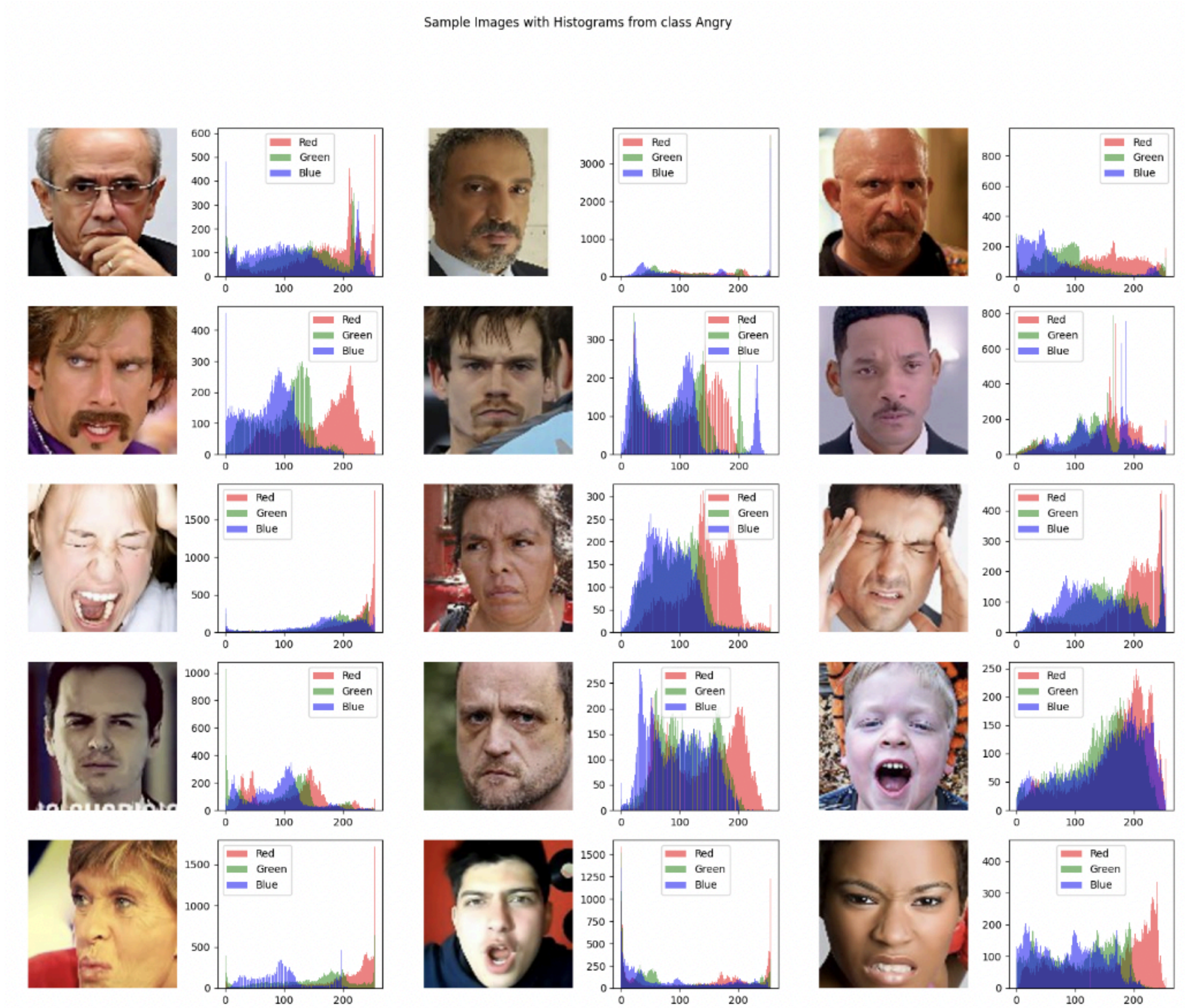

Pixel Intensity Distribution for class Focused
Red
Green
Blue
Frequency
Pixel Intensity
0
2
4
6
8
0
50
100
150
200
250


Pixel Intensity Distribution for class Neutral
Red
Green
Blue
Frequency
Pixel Intensity
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
50
100
150
200
250

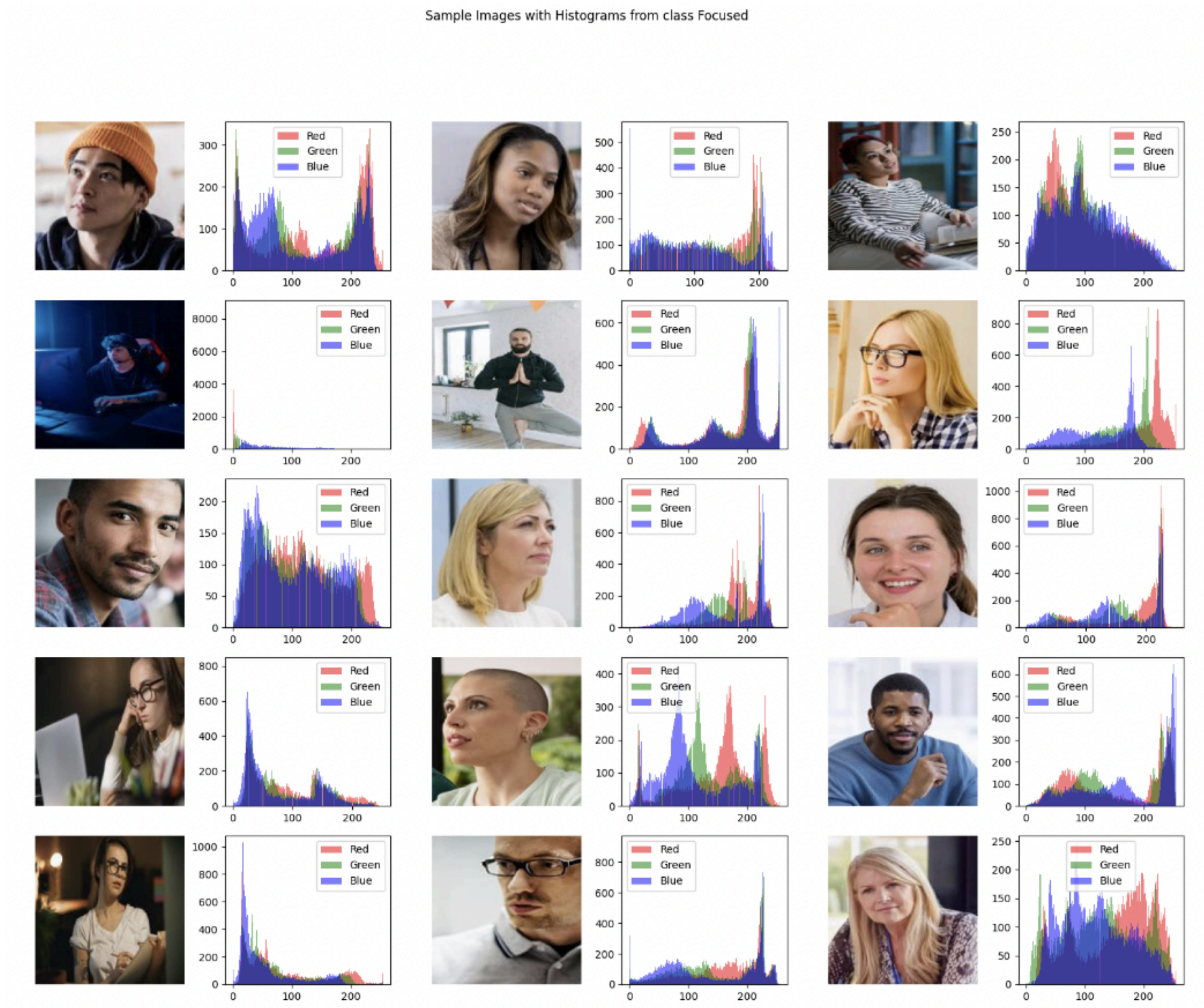

Pixel Intensity Distribution for class Happy
Red
Green
Blue
Frequency
Pixel Intensity
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
50
100
150
200
250

3) Sample Images:

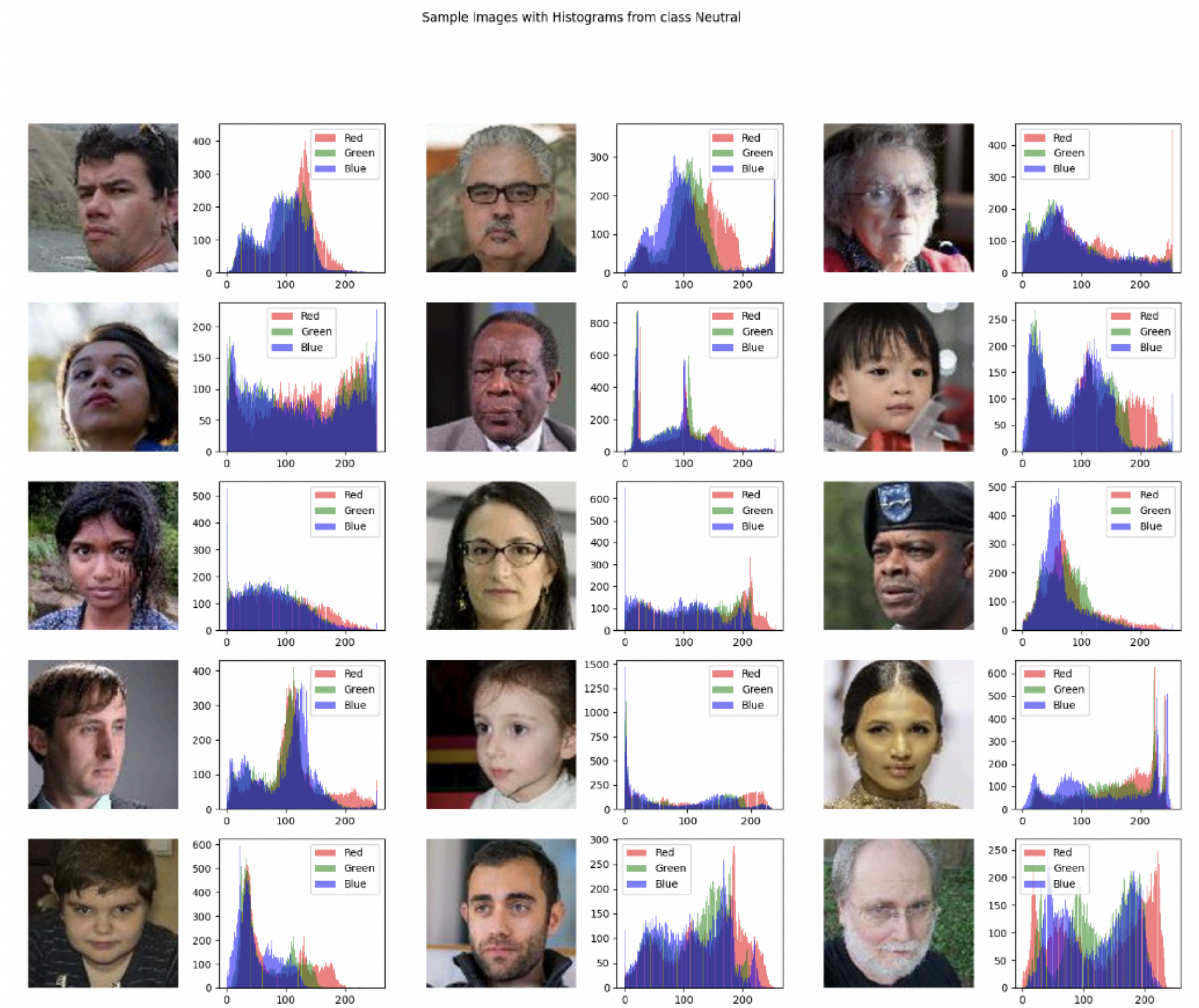
Class Angry:



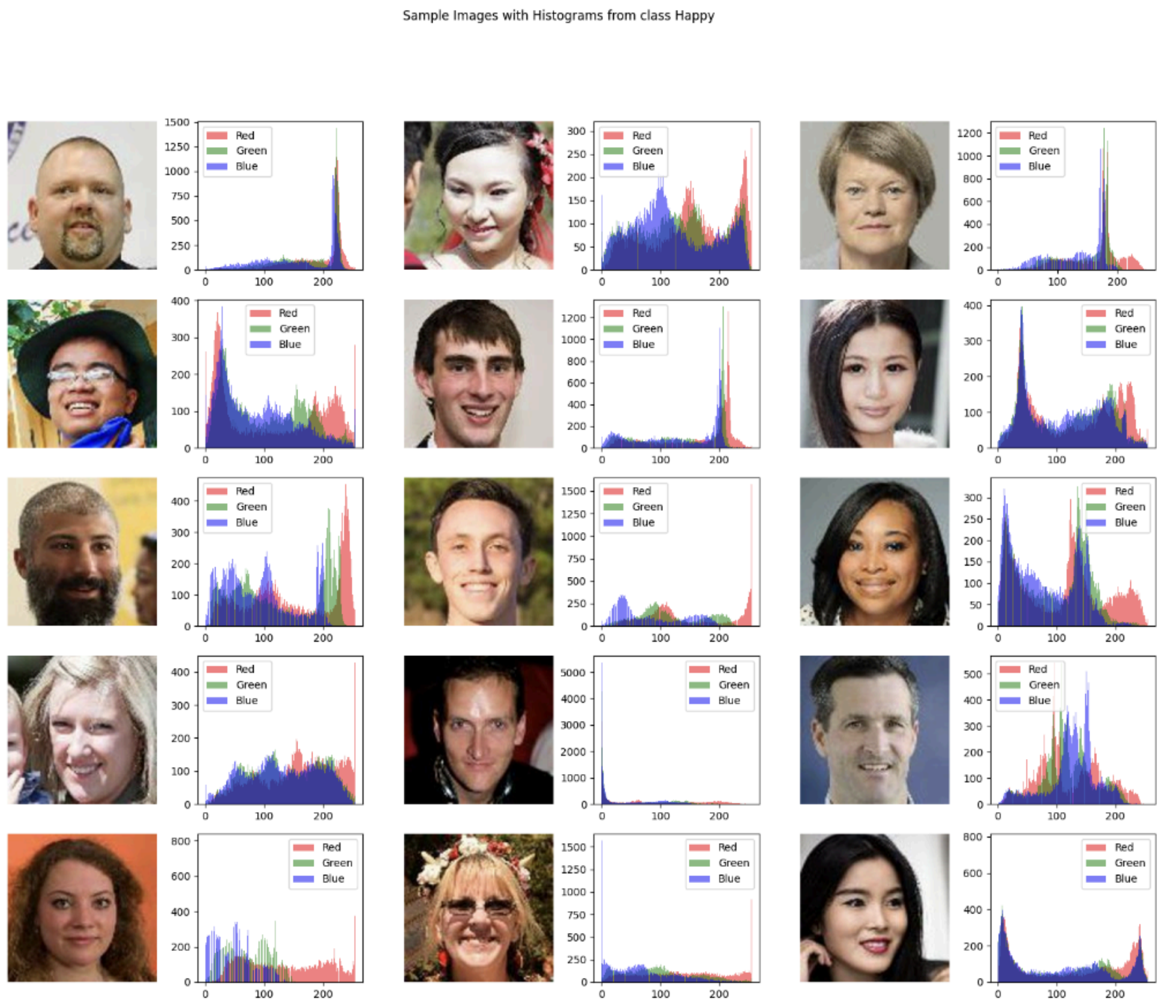
Class Focused:



Class Neutral:



Class Happy:


Sample Images with Histograms from class Happy
Red
Green
Blue

CNN Architecture

Model Overview and Architecture Details

The architecture of the main model consists of 3 convolutional layers, each having a kernel size of 3x3 . This can be seen as follow:

- **Number of Convolutional Layers:** 3
- **Convolutional Layers:**
- Conv1: 3x3 kernel, 32 filters, stride 1, padding 1
- Conv2: 3x3 kernel, 64 filters, stride 1, padding 1
- Conv3: 3x3 kernel, 128 filters, stride 1, padding 1
- **Activation Functions:** ReLU applied after each convolutional layer and in the first fully connected layer
- **Pooling:** Max pooling with a 2x2 kernel and stride 2 after each convolutional layer
- **Fully Connected Layers:** FC1: 128 neurons, FC2: 4 neurons corresponding to the 4 classes
- **Output Layer:** 4 neurons corresponding to the 4 classes. Includes a softmax activation function to produce the class probabilities.
- **Regularization Techniques:** None

The changes made for Variant 1 that are different from the main model is that Variant 1 only has two convolutional layers, with a kernel size of 3x3. More details can be seen as follows:

- **Number of Convolutional Layers:** 2
- **Convolutional Layers:**
- Conv1: 3x3 kernel, 32 filters, stride 1, padding 1
- Conv2: 3x3 kernel, 64 filters, stride 1, padding 1
- **Activation Functions:** ReLU applied after each convolutional layer and in the output layer
- **Pooling:** Max pooling with a 2x2 kernel and stride 2 after each convolutional layer
- **Fully Connected Layers:** FC1: 128 neurons, FC2: 4 neurons corresponding to the 4 classes
- **Output Layer:** 4 neurons corresponding to the 4 classes. Includes a softmax activation function to produce the class probabilities.
- **Regularization Techniques:** None

Lastly, the changes that were applied to Variant 2 was that we kept the same amount of convolutional layers as the main model (3), but we modified the kernel size to 5x5. More details:

- **Number of Convolutional Layers:** 3
- **Convolutional Layers:**
- Conv1: 5x5 kernel, 32 filters, stride 1, padding 1
- Conv2: 5x5 kernel, 64 filters, stride 1, padding 1
- Conv3: 5x5 kernel, 128 filters, stride 1, padding 1
- **Activation Functions:** ReLU applied after each convolutional layer and in the output layer
- **Pooling:** Max pooling with a 2x2 kernel and stride 2 after each convolutional layer
- **Fully Connected Layers:** FC1: 256 neurons, FC2: 4 neurons corresponding to the 4 classes
- **Output Layer:** 4 neurons corresponding to the 4 classes. Includes a softmax activation function to produce the class probabilities.
- **Regularization Techniques:** None

Training Process

The training process for the models (Main Model, Variant 1, Variant 2) involves several key components to optimize each model's performance. Below are the details for the methodology used for training, including the number of epochs, learning rate, loss function, and optimization algorithms.

- Number of epochs: 10
- Learning rate: 0.001
- Loss function: CrossEntropyLoss
- Optimization algorithm: Adam optimizer
- Patience set to 3 for early stopping. We also tested with patience=4 and patience=2 but we found that we got the best results with patience=3.

The models (MainModel, Variant1, Variant2) are trained for 10 epochs each using the Adam optimizer with a learning rate of 0.001. The CrossEntropyLoss function is used as the loss function. This function was used for calculating the loss between predicted class probabilities and actual class labels.

The training process includes mini-batch gradient descent, where the dataset is split into batches of size 32 for training. The mini-batch gradient descent helps with the computation efficiency and optimizes memory usage compared to processing the entire dataset at once.

Early stopping is also implemented to prevent overfitting and to monitor validation loss. It stops training if it doesn't improve for a certain number of epochs (patience=3). This helps prevent overfitting and ensures that the model generalizes well to unseen data.

Evaluation

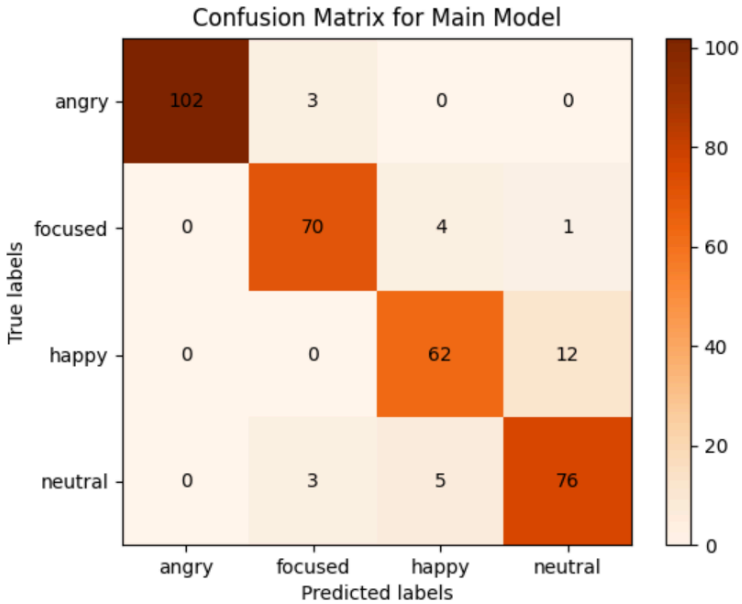
Performance Metrics

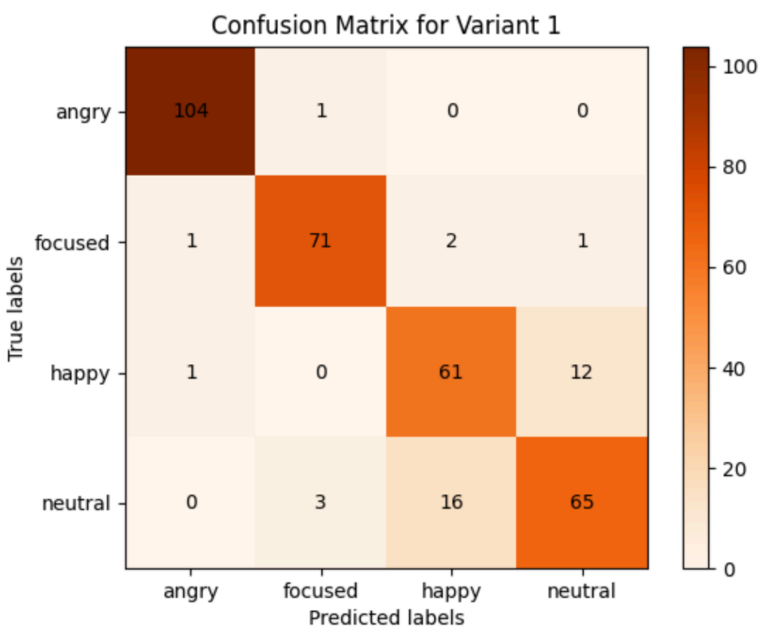
Model	Macro P	Macro R	Macro F	Micro P	Micro R	Micro F	Accuracy
Main Model	0.9121	0.9118	0.9116	0.9172	0.9172	0.9172	0.9172
Variation 1	0.8833	0.8838	0.8831	0.8905	0.8905	0.8905	0.8905
Variation 2	0.9065	0.8996	0.9005	0.9083	0.9083	0.9083	0.9083

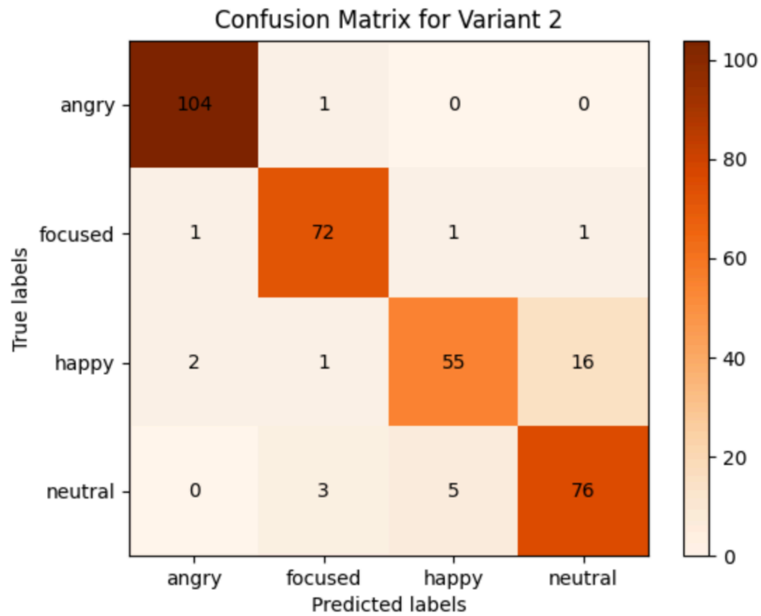
The Main Model has three convolutional layers and a 3x3 kernel size. The Variant 1 model has two convolutional layers and a 3x3 kernel. The Variant 2 model has three convolutional layers and a 5x5 kernel. After running a series of experiments with varying kernel sizes and convolutional layers (see experiments below), these three models were determined to have the highest performance.

As seen in the table above, the Main Model has the best performance out of the three with higher precision, recall and F1-measure for both macro and micro as well as overall accuracy. In second place would be the Variation 2 model, and in third the Variation 1 model. The performance for the two variants change a little each time one trains the models. However, the Main Model consistently yields the best performance which implies that a smaller kernel size with more convolutional layers yields the best results in the context of facial recognition analysis.

Confusion Matrix Analysis


Confusion Matrix for Main Model
True labels
angry
focused
happy
neutral
102
3
0
0
0
70
4
1
0
0
62
12
0
3
5
76
Predicted labels
100
80
60
40
20
0


Confusion Matrix for Variant 1
True labels
angry
focused
happy
neutral
104
1
0
0
1
71
2
1
1
0
61
12
0
3
16
65
Predicted labels
100
80
60
40
20
0



In general, our models were able to correctly classify images. This can be attributed to the fact that team members thoroughly cleaned the data during phase one of the project. To ensure quality data, the team manually inspected each image used in the dataset to ensure they were correctly labeled (placed in the appropriate class: angry, happy, focused or neutral) and had a good view of the person's face. The team also converted all images to JPEG, resized and renamed them all to have standardized data during the data cleaning part of the project.

Our models did exceptionally well at correctly classifying images that were angry and images that were focused. The images used to train the model to recognize an angry facial expression contained little room for misinterpretation. In other words, the training images for anger brazenly depicted an angry face with almost no nuance. Similarly, the images used to train the model to recognize a focused facial expression very obviously showed a person looking focused with little to no nuance. The clear representation of angry and focused facial expressions in our training data is a possible reason for angry and focused to be exceptionally well-recognized classes.

Our models had some difficulty differentiating between a neutral facial expression and a happy facial expression. The images used to train the model to look happy contained some nuance. Some people in our happy training data were only slightly smiling which could be misinterpreted as having a neutral expression. Likewise, some people in the neutral training data appeared relaxed and showed slight signs of contentment which could be misinterpreted as happy. The nuance in the micro-expressions of our training data for happy and neutral could have contributed to the models' difficulty in differentiating between those two classes.

Impact of Architectural Variations

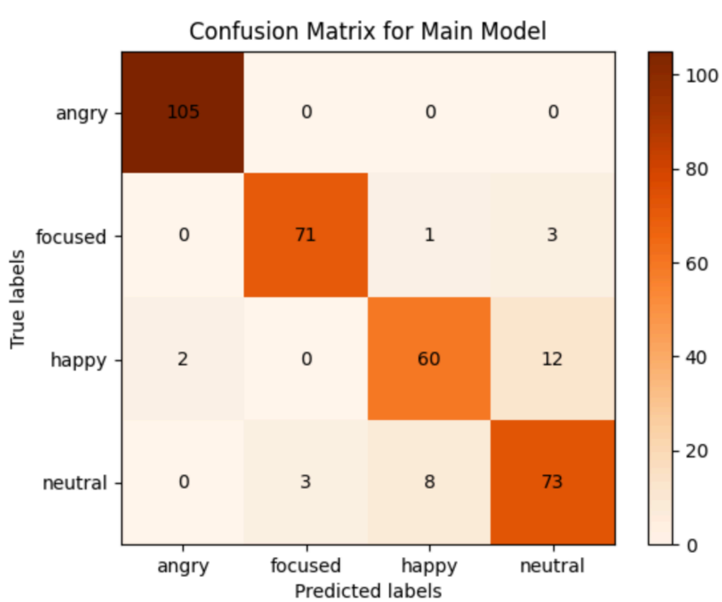
To understand how kernel size and number of convolutional layers affects performance, the team ran a series of experiments. A detailed analysis of each experiment and its implications can be found below.

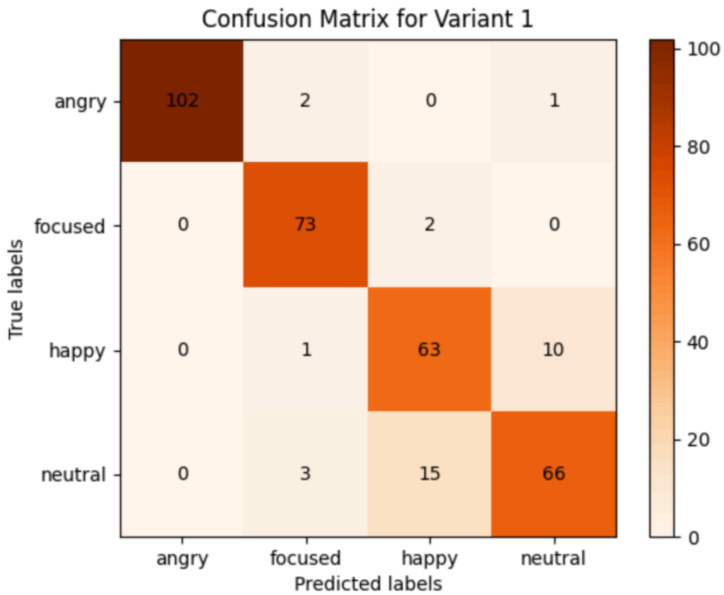
First Experiment:

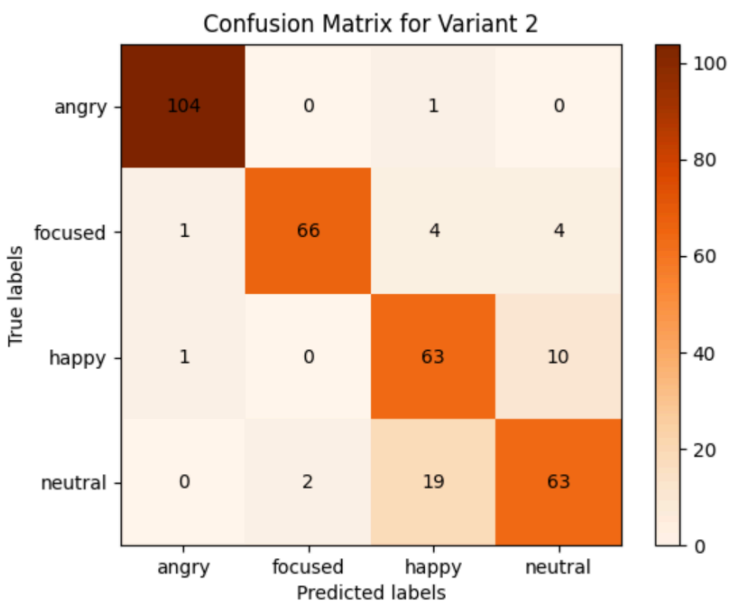
Description of Model

Model	Number of Convolutional Layers	Convolution 1	Max Pooling	Convolution 2	Convolution 3
Main Model	2	3x3	2x2	3x3	N/A
Variant 1	2	5x5	2x2	5x5	N/A
Variant 2	2	7x7	3x3	7x7	N/A

Confusion Matrices:







<u>Summary of Metrics:</u>

Model	Macro P	Macro R	Macro F	Micro P	Micro R	Micro F	Accuracy
Main Model	0.91	0.9066	0.9079	0.9142	0.9142	0.9142	0.9142
Variation 1	0.8922	0.8955	0.8929	0.8994	0.8994	0.8994	0.8994
Variation 2	0.8735	0.868	0.8685	0.8757	0.8757	0.8757	0.8757

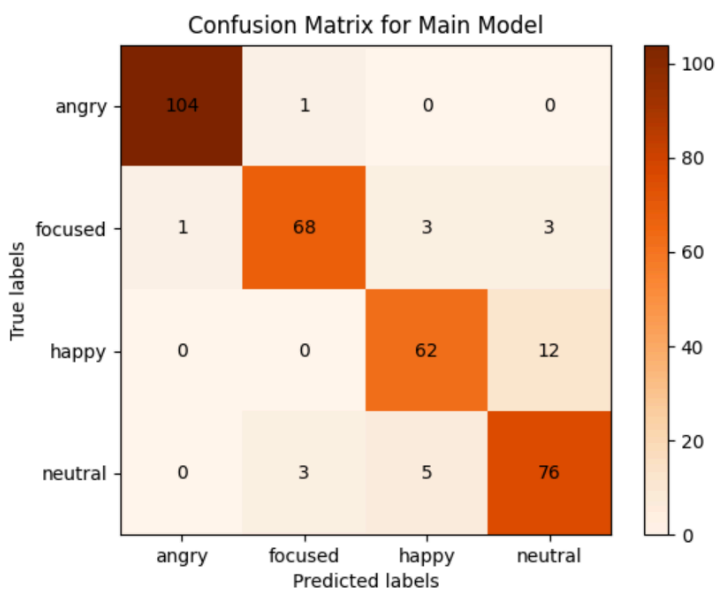
In the first experiment, all models used the same number of convolutional layers (which was two) and varying kernel sizes. The Main Model used a kernel size of 3x3, Variant 1 used a kernel size of 5x5 and Variant 2 used a kernel size of 7x7. The confusion matrices and the table summarizing the metrics for each model clearly indicate that the Main Model performed the best in experiment one.

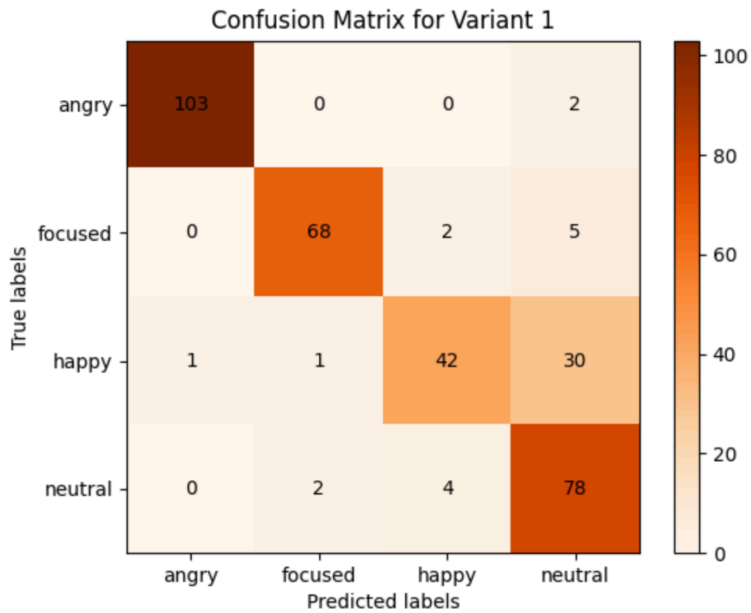
Second Experiment:

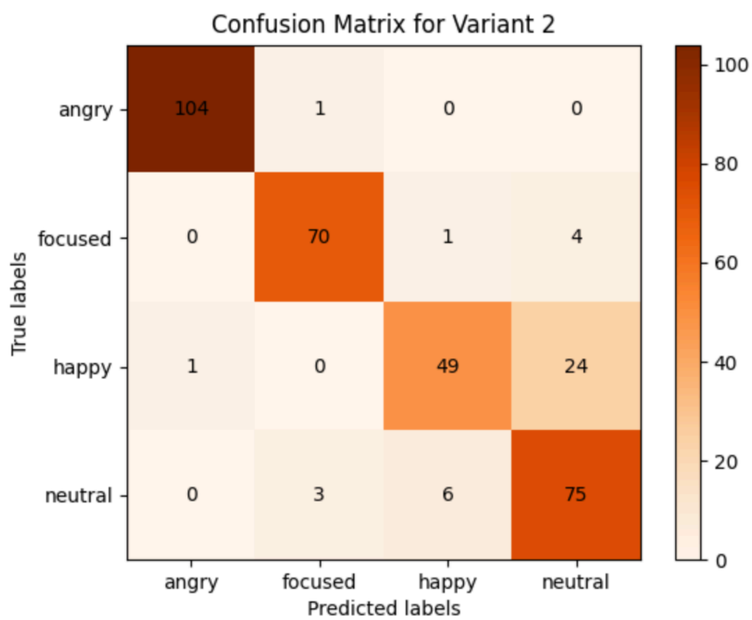
<u>Description of Model</u>

Model	Number of Convolutional Layers	Convolution 1	Max Pooling	Convolution 2	Convolution 3
Main Model	3	3x3	2x2	3x3	3x3
Variant 1	3	5x5	2x2	5x5	5x5
Variant 2	3	7x7	3x3	7x7	7x7

<u>Confusion Matrices:</u>




Confusion Matrix for Variant 1
True labels
Predicted labels
angry
focused
happy
neutral
103
0
0
2
0
68
2
5
1
1
42
30
0
2
4
78
100
80
60
40
20
0


Confusion Matrix for Variant 2
True labels
Predicted labels
angry
focused
happy
neutral
104
1
0
0
0
70
1
4
1
0
49
24
0
3
6
75
100
80
60
40
20
0

Summary of Metrics:

Model	Macro P	Macro R	Macro F	Micro P	Micro R	Micro F	Accuracy
Main Model	0.9139	0.9099	0.9113	0.9172	0.9172	0.9172	0.9172
Variation 1	0.8753	0.8459	0.8474	0.8609	0.8609	0.8609	0.8609
Variation 2	0.8849	0.8697	0.8715	0.8817	0.8817	0.8817	0.8817

In the second experiment, all models used the same number of convolutional layers (which was three) and varying kernel sizes. The Main Model used a kernel size of 3x3, Variant 1 used a kernel size of 5x5 and Variant 2 used a kernel size of 7x7. The confusion matrices and the table summarizing the metrics for each model clearly indicate that the Main Model performed the best in experiment two.

Impact of Architectural Variations:

Facial image analysis relies heavily on the depth of convolutional layers. Models with deeper architectures, such as Variant 1 in the first experiment, which performed well in the recall, capture deeper facial features due to increased layering. Deeper models, on the other hand, run a risk of overfitting if they are not appropriately regularized, particularly with small datasets. Variant 1's depth in our models most certainly contributed to its strong recall, but careful regularization was required to avoid overfitting.

Kernel size variations affect a model's ability to capture face features at various scales. Smaller kernel sizes (e.g. 3x3) are useful for finer details like facial textures and lines, whereas larger kernels (e.g. 5x5 or 7x7) are better suited for wider aspects like overall face shape and spatial relationships between facial elements. Each version in our models most likely used varied kernel sizes to find a balance between collecting precise facial traits and overall facial attributes.

Primary Findings:

The Main Model from Experiment 2 had the best performance, with excellent recall and competitive precision, so it is useful in complete face detection environments. The Main Model provided balanced precision and recall metrics, making it ideal for broad facial recognition apps. In general, it seems that a smaller kernel size with more convolutional layers yields the best performance for facial recognition according to our findings.

Suggestions for future refinements:

1) Refinements to the Model Architecture:

   We should try several depth configurations to make sure the models accurately reflect the required facial characteristics without overfitting. Also, we should adjust kernel size selections to get a better balance between capturing facial features and fine details.

2) Training Strategies:
   Use strong regularization techniques to avoid overfitting in deeper models such as Variant 1. Explore creative data augmentation techniques to improve model reliability and generalization.

3) Dataset Considerations:

   To increase model performance across different demographics and situations, ensure that datasets are well-balanced and include a variety of facial features.

References

[1] N. Segal, “Facial expressions training data,” Kaggle, https://www.kaggle.com/datasets/noamsegal/affectnet-training-data?select=anger (accessed May 24, 2024).

[2] Focused Photos, Download Free Focused Stock Photos & HD Images, https://www.pexels.com/search/focused/ (accessed May 25, 2024).

[3] Freepik, https://www.freepik.com/search?ai=excluded&format=search&last_filter=query&last_value=&query=&type=photo (accessed May 25, 2024).

[4] Unsplash, “Beautiful free Images & Pictures | Unsplash,” *Unsplash*. https://unsplash.com/ (accessed May 25, 2024).

[5] GeeksforGeeks, “Convert PNG to JPG using python,” GeeksforGeeks, https://www.geeksforgeeks.org/convert-png-to-jpg-using-python/ (accessed May 25, 2024).

[6] “How to resize all images in a folder using Python,” Cloudinary, https://cloudinary.com/guides/bulk-image-resize/how-to-resize-all-images-in-a-folder-using-python (accessed May 26, 2024).

[7] “Rename multiple files using Python,” GeeksforGeeks, https://www.geeksforgeeks.org/rename-multiple-files-using-python/ (accessed May 27, 2024).

[8] “Visualization with python,” Matplotlib, https://matplotlib.org/ (accessed May 27, 2024).

[9] S. Khant, “Why (how) do we split train, valid, and test?,” *Artificialis*, Jul. 20, 2023. https://medium.com/artificialis/why-how-we-split-train-valid-and-test-fb4d6746ede (accessed June 7, 2024).

[10] N. Vishwakarma, “What is Adam Optimizer?,” *Analytics Vidhya*, Sep. 29, 2023. https://www.analyticsvidhya.com/blog/2023/09/what-is-adam-optimizer/#:~:text=Practical%20Tips%20for%20Using%20Adam%20Optimizer (accessed June 8, 2024).

[11] “Building a Convolutional Neural Network using PyTorch,” *GeeksforGeeks*, May 10, 2024. https://www.geeksforgeeks.org/building-a-convolutional-neural-network-using-pytorch/. (accessed Jun. 8, 2024)

[12] Geeksforgeeks, “Confusion Matrix in Machine Learning - GeeksforGeeks,” *GeeksforGeeks*, Feb. 07, 2018. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/ . (accessed Jun. 6, 2024)

[13] "Saving and Loading Models — PyTorch Tutorials 1.4.0 documentation," *Pytorch.org*, 2017. https://pytorch.org/tutorials/beginner/saving_loading_models.html. (accessed: Jun. 10, 2024)

[14] "Build, train, and run your PyTorch model," *Red Hat Developer*, Sep. 28, 2022. https://developers.redhat.com/learning/learn:openshift-ai:how-create-pytorch-model/resource/resources:build-train-and-run-your-pytorch-model. (accessed: Jun. 10, 2024)