

COMP 472

Final Project Report

April 21, 2024
Sherief Soliman – 29248323
Group: AZ_10

Contents

| | |
|---|----|
| 1- Overview of the Existing Datasets Used | 4 |
| Justification for Dataset Choices | 4 |
| Challenges and Considerations | 5 |
| Provenance Information | 5 |
| 2- Data Cleaning | 5 |
| Resizing Images: | 5 |
| Brightness Adjustment: | 6 |
| Challenges and Solutions: | 6 |
| 3- Labeling Overview: | 7 |
| Methods and Tools: | 7 |
| Creation of Focused/Engaged Class: | 7 |
| Challenges and Solutions: | 7 |
| 4- Dataset Visualization | 8 |
| Class Distribution: | 8 |
| Sample Images: | 9 |
| Pixel Intensity Distribution: | 9 |
| 5- Model Overview and Architecture Details | 11 |
| Main Model Overview | 11 |
| Convolutional Layers: | 11 |
| Activation and Pooling: | 11 |
| Dropout: | 11 |
| Fully Connected Layers: | 11 |
| Design Nuisances: | 11 |
| Variants: | 12 |
| 6- Training the Model | 12 |
| Number of Epochs and Early Stopping | 12 |
| Learning Rate | 12 |
| Loss Function | 12 |
| Optimization Algorithm | 12 |
| 7- Evaluation of All Variants | 13 |
| Performance Metrics: | 13 |
| Confusion Matrix Analysis | 13 |
| Impact of Architectural Variations | 15 |
| Changes Made for Part III | 15 |
| 8- Main Model Evaluation using K-Fold | 16 |
| Updated Performance of Main Model | 16 |
| K-Fold evaluation of Main Model from Part II | 16 |

| | |
|---|----|
| 9- Bias Detection and Analysis | 18 |
| Introduction..... | 18 |
| Bias Detection Results | 18 |
| Bias Robustness Check | 20 |
| 10 - Conclusion: | 22 |
| Appendix..... | 23 |
| Expectations of Originality | 23 |
| GitHub Repository Link..... | 23 |

1- Overview of the Existing Datasets Used

For our facial emotion recognition project, we utilized two primary datasets sourced from Kaggle: the FER Dataset <https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset/data> and the other Emotion Detection

Dataset <https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer> . These datasets provided a comprehensive range of facial expressions relevant to our project objectives. The datasets utilized in this project consist of grayscale images with dimensions of 48x48 pixels. All images within both datasets share this uniform size, eliminating the need for data reshaping during preprocessing.

These images all consist of close face shots of the subjects. The subjects are varied across age and gender demographics to reduce bias. As the goal of the images is to primarily focus on capturing facial expressions, there is minimal background information. The faces take up most of the pixel space. The images are standardized to the 48x48 pixel resolution and grayscale color palette to ensure consistency in size and quality across the dataset.

The first dataset utilized is the Facial Expression Recognition (FER) dataset, which originally contains a total of 28,821 grayscale images categorized into seven emotion classes. However, for this project, a subset of three classes—Happy, Neutral, and Surprised—was selected. The distribution of samples in this subset is as follows:

Dataset 1: FER Dataset

- **Total Samples:** 28,821
- **Classes:** 7 (Anger, Disgust, Fear, Happy, Sadness, Surprise, Neutral)
- **Distribution (Subset Used):**
 - Happy: 7,164
 - Neutral: 4,982
 - Surprised: 3,205

Dataset 2: Ananthu017 Emotion Detection FER Dataset

- **Total Samples:** 35,685
- **Classes:** 7 (Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise)

Combined Dataset for Project

- **Total Samples:** 3,150
- **Classes:** 4 (Happy, Neutral, Surprised, Focused/Engaged)
- **Distribution (Subset Used):**
 - Happy: 800
 - Neutral: 800
 - Surprised: 800
 - Focused/Engaged: 750

Justification for Dataset Choices

The selection of datasets was driven by the need to create a robust facial emotion recognition model specifically tailored to classroom or online meeting settings. The FER dataset provided a substantial

amount of labeled data covering various facial expressions, including happiness, neutrality, and surprise. However, the lack of a focused/engaged class necessitated the exploration of additional datasets.

The Ananthu017 Emotion Detection FER Dataset complemented the FER dataset by providing additional images for the focused/engaged class. As both datasets contain images of the same size and grayscale format, meeting our consistency requirements, combining them allowed for the creation of a more comprehensive and balanced dataset, essential for training a reliable deep learning model.

Relevance to Project Objectives:

- The FER Dataset offered a wide range of emotions, including three of the required classes: happiness, neutrality, and surprise. It provided a substantial number of images for these classes, making it a suitable primary dataset.
- The Second Emotion Detection supplemented our dataset by offering additional images within the same classes. While it didn't explicitly include the focused/engaged class, it enriched the dataset's diversity, contributing to the model's overall performance.

Challenges and Considerations

1. **Dataset Imbalance:** While the FER dataset initially contained a sizable number of samples, the distribution across classes was imbalanced, requiring careful selection and sampling to ensure equal representation of the desired classes.
2. **Manual Labeling:** The creation of the focused/engaged class involved manual labeling of images sourced from both datasets. This process required meticulous selection and verification to maintain class coherence and balance within the dataset.
3. **Limited Class Availability:** The absence of the focused/engaged class in existing datasets necessitated the sourcing and integration of additional data from multiple sources, adding complexity to the dataset creation process.

Provenance Information

| Dataset | Number of Images (3 classes) | Licensing Type | Source |
|--------------------|-------------------------------|----------------|--------|
| FER Dataset | 16,351 | CC BY 4.0 | Kaggle |
| Ananthu017 Dataset | 19,000 | CC0 | Kaggle |

2- Data Cleaning

In the process of preparing our dataset for training, several data cleaning techniques were employed to standardize the dataset and enhance its quality. These techniques included resizing images to a consistent size, adjusting brightness, and ensuring uniformity in the dataset.

Resizing Images:

One of the initial steps in data cleaning involved resizing the images to a standard size of 48x48 pixels. This standardization ensured consistency in the dimensions of all images in the dataset. By resizing the images, we eliminated variations in size that could potentially introduce biases during

model training. Additionally, resizing allowed us to optimize computational resources by working with images of uniform dimensions.

Brightness Adjustment:

To address variations in lighting conditions across images, brightness adjustment techniques were applied. These techniques involved scaling the pixel values of the images to enhance or reduce their brightness while preserving their overall appearance. By adjusting brightness, we aimed to minimize the impact of lighting variations on the model's performance and ensure that it learned to recognize facial expressions accurately under different lighting conditions.

Challenges and Solutions:

One of the challenges encountered during the data cleaning process was the presence of images with inconsistent sizes and lighting conditions. To address this challenge, a combination of resizing and brightness adjustment techniques was applied uniformly across the dataset. By standardizing the size and brightness of all images, we mitigated the effects of these variations and ensured that the dataset was suitable for training our model.

Example Illustration:

To illustrate the impact of data cleaning techniques, consider the following example images:

Original Image



Adjusted Image



Original Image



Adjusted Image



Original Image



Adjusted Image



Original Image



Adjusted Image



Original Image



Adjusted Image



•

The before-and-after comparison clearly demonstrates how slight rotations or brightness adjustments improved the quality and consistency of the images, making them more suitable for training a facial expression recognition model.

In conclusion, through the application of standardization techniques such as resizing and brightness adjustment, we successfully cleaned our dataset to ensure uniformity and enhance the quality of the images. These efforts were essential for preparing a high-quality dataset capable of training a robust deep learning model for facial expression recognition.

3- Labeling Overview:

Labeling the dataset involved assigning appropriate emotion labels to each image, ensuring that the dataset accurately represented the range of facial expressions relevant to the project. This process included merging multiple datasets, mapping existing classes, and manually creating the new class, "Focused/Engaged."

Methods and Tools:

The labeling process primarily involved manual annotation using image labeling software and platforms such as LabelImg, LabelMe, or custom scripts by authors of datasets. Each image was examined, and the corresponding emotion label was assigned based on the facial expression depicted in the image. For datasets with pre-existing labels, this step was relatively straightforward. For the manually created "Focused/Engaged" class, a more nuanced approach was required.

Creation of Focused/Engaged Class:

The "Focused/Engaged" class was manually created by merging overlapping images from other classes, including Angry, Sad, Surprise, and Contempt. This process involved carefully reviewing images from these classes and selecting those that exhibited signs of focus or engagement, such as intense concentration or active participation. By tagging these images, a new class representing the focused or engaged state was formed, providing a comprehensive representation of student engagement in classroom or online settings.

Challenges and Solutions:

During the labeling process, ambiguities were encountered in images where facial expressions were subtle or ambiguous. In such cases, decisions were made based on the predominant emotion conveyed by the facial features. Additionally, consensus among annotators or consultation with domain experts was sought to resolve ambiguous cases and ensure accurate labeling.

In addition, merging multiple datasets and mapping classes presented challenges related to class imbalance and overlapping categories. To address these challenges, careful consideration was given to the selection of images to create a balanced and representative dataset. Additionally, clear documentation of the mapping process and the rationale behind it helped maintain transparency and reproducibility.

Overall, the labeling process involved a combination of manual annotation, merging datasets, and creating new classes to ensure the dataset's completeness and relevance to the project objectives. By carefully navigating ambiguities and challenges, we arrived at a well-labeled dataset suitable for training a facial expression recognition model.

4- Dataset Visualization

Class Distribution:

The class distribution analysis provides insights into the distribution of images across different emotion classes in the dataset. A bar graph representing the number of images in each class is presented below:

The bar graph indicates that the dataset contains a relatively balanced distribution of images across the different emotion classes. This balance is essential for training a machine learning model that can effectively recognize and classify facial expressions.

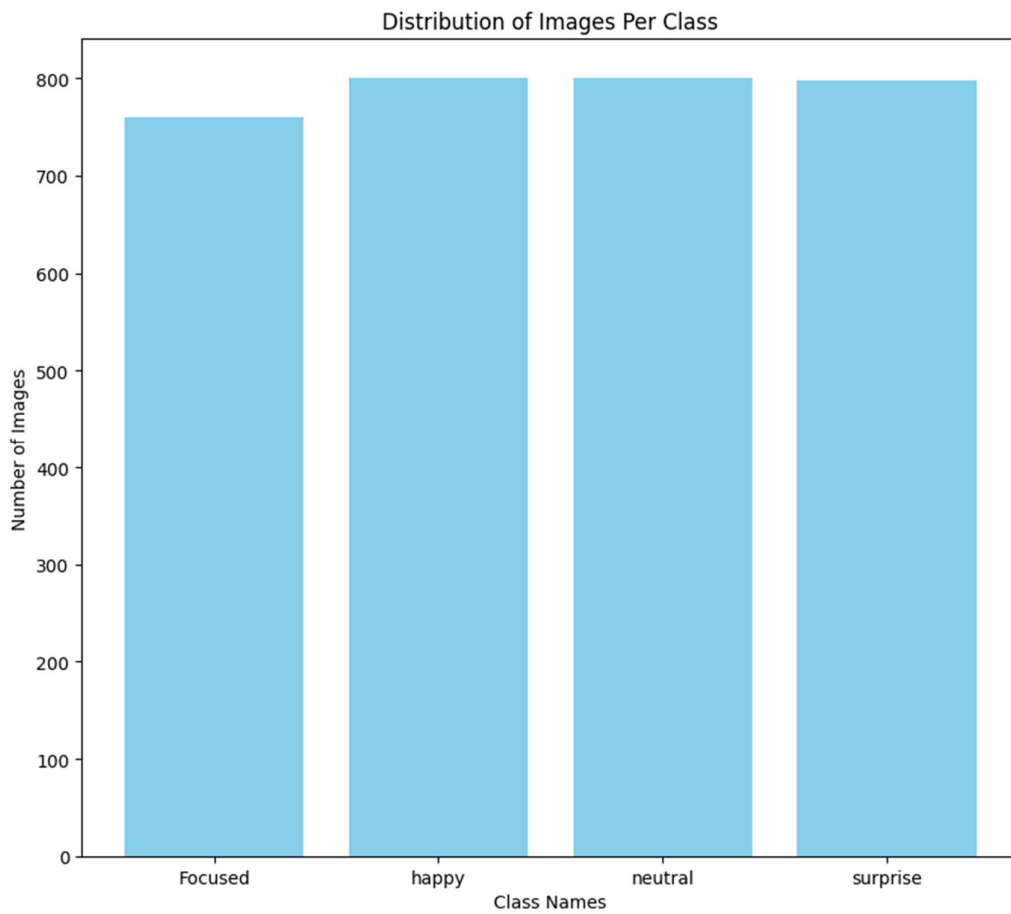


Figure 1: Class Distribution of Images

Sample Images:

A collection of 25 randomly chosen images from each class is displayed below in a 5x5 grid format. The sample images showcase the variety of facial expressions present in the dataset. By visually inspecting these images, potential anomalies or mislabeling can be identified. Additionally, the random selection of images ensures that the dataset's content is well-represented in the visualization.

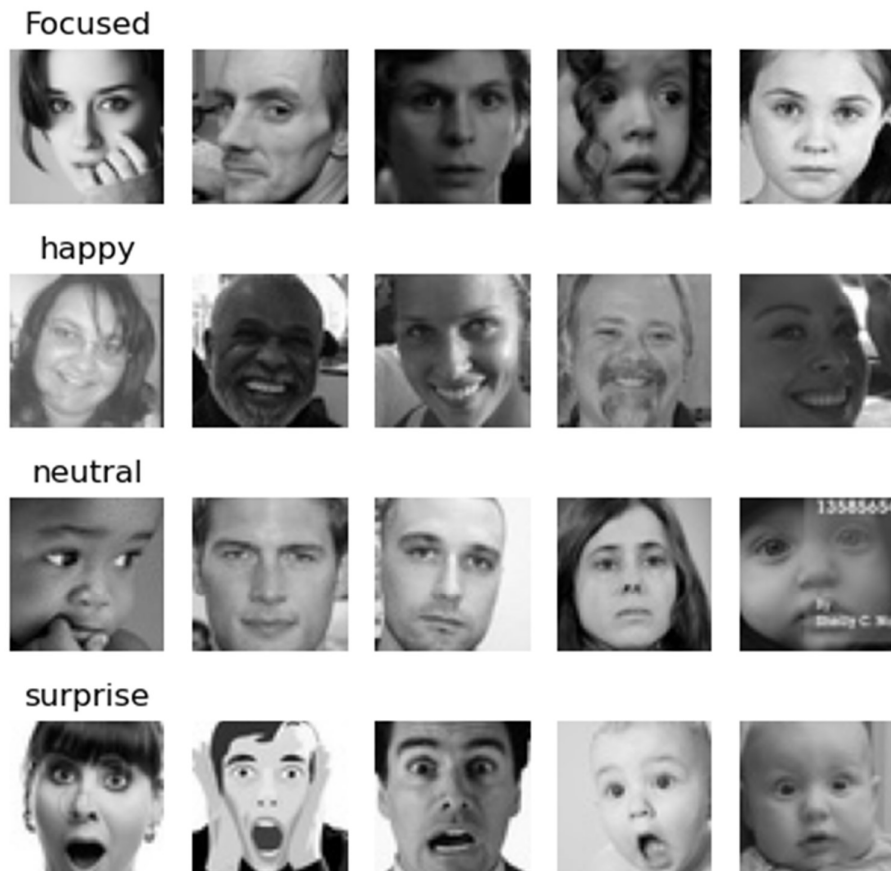


Figure 2: 4x5 grid of randomly chose images from each class

Pixel

Intensity Distribution:

A histogram illustrating the distribution of pixel intensities for the randomly selected images is presented below:

The histogram provides insights into the variations in lighting conditions among the images. In grayscale images, the pixel intensity ranges from 0 (black) to 255 (white). The distribution of pixel intensities reveals the prevalence of certain intensity levels, which can indicate common lighting conditions or image characteristics within the dataset.

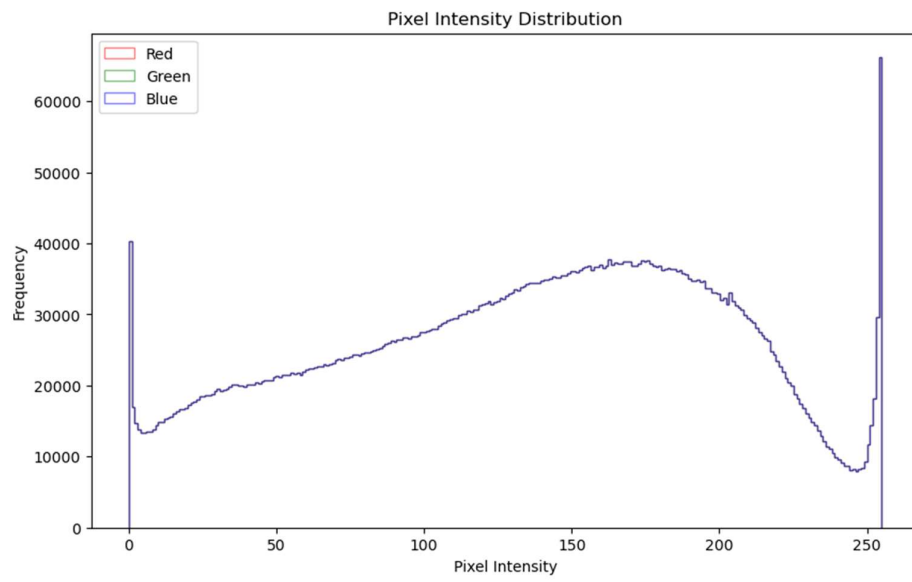


Figure 3: histogram illustrating the distribution of pixel intensities

5- Model Overview and Architecture Details

Main Model Overview

The main model is a convolutional neural network that is designed for prediction and classification of the data prepared previously. This model was decided upon after experimentation and iteration with different setups to see what gave the best results. It incorporates a series of convolutional layers, each followed by batch normalization and ReLU activation functions, and utilizes max pooling for spatial dimensionality reduction. The architecture emphasizes feature extraction through its convolutional layers and employs dropout for regularization before proceeding to a sequence of fully connected layers for classification.

Convolutional Layers:

There are a total of three convolutional layers in the main model. The model begins with a convolutional layer with 32 filters with a kernel size 5x5. This is then followed by a second convolutional layer with 64 and a kernel size of 3x3. The final layer is 128 filters with a kernel size of 3x3, respectively.

Activation and Pooling:

After batch normalization is applied to each layer, the Rectified Linear Unit or ReLU activation function is then applied. This activation function is applied to assist the model in learning more complex patterns in the data by introducing non-linearity. A max pooling operation is then applied after the activation function. The pooling is done with a kernel of size 3x3, and a stride of 2. This is meant to reduce the computation required in the network.

Dropout:

Before the fully connected layers, a dropout layer is applied with a rate of 0.5. The reason this layer is used is to help prevent possible overfitting during training. It will randomly drop units and connections throughout the training process to equally distribute the effect of each neuron on the next layer. The rate of 0.5 was chosen as it was the standard recommended rate. No experimentation was done with different values of the dropout rate.

Fully Connected Layers:

Finally, the model concludes with three fully connected layers. The first layer was set to 512 to have enough capacity to learn from the features extracted at the convolutional layers without too much risk of overfitting. The second layer was introduced to funnel the flow of information towards the output as the jump from 512 neurons to 4 outputs seemed drastic. This can compress the feature representation, hopefully allowing the model to focus on more relevant information for classification. The final layer takes the 128 and produces the final 4 classifications.

Design Nuisances:

For the main model, the activation function of ReLU was used over leaky ReLU. While in theory leaky ReLU would be the better choice as it attempts to solve the vanishing gradient problem, the performance appeared to be slightly better with the ReLU activation function applied. The dropout rate was discussed above, employing a standard rate of 0.5. Finally, it was decided that three fully

connected layers would be used to gradually step down from the number of neurons towards the 4 classification outputs.

Variants:

Both variants use a kernel of 3x3 for the first convolution layer as opposed to the 5x5 size chosen for the main model. They also both employ the leaky ReLU activation function over the ReLU chosen. This was because through experimentation, the 5x5 kernel size at this layer yielded the best result. Variant 1 uses the same implementation as the main model except for the kernel size. Variant 2 was experimentation with additional layers to see their effect on the performance of the models.

6- Training the Model

Number of Epochs and Early Stopping

As we wanted to ensure we are running a minimum of 10 epochs for each model, the maximum was set to 20 to observe any early stopping that may occur. To implement early stopping, a patience value was set to 3. This value was used to determine if there was no improvement in the loss rate of the validation data. Should there be three consecutive epochs that did not have improvement at the validation loss rate, the training would stop. To ensure that the model had the best loss rate, only those epochs which provided a lower loss rate were saved. This ensures that the best and not the last model was the one ultimately used for the performance evaluation.

Learning Rate

The learning rate that was decided on was 0.001. This was chosen through experimentation as it led to the most epochs being run when training the data and a steady decrease in loss across the epochs until reaching the stopping point. Learning rates higher, such as 0.01, would not reach the minimum of 10 epochs before stopping would occur. This finer convergence with more epochs displayed better performance, which is why this learning rate was used.

Loss Function

The Cross Entropy Loss function was utilized. As the model is intended for a multi-class classification task, this is the most suitable loss function since each instance will belong to exactly one class. The loss is computed across all classes and then averaged over all observations to give the final loss result. Logit handling is automated in pytorch as the `nn.CrossEntropyLoss` function as it already computes the log softmax which is passed from the output of the model.

Optimization Algorithm

The Adam Optimizer was chosen as it is suitable for a wide range of tasks. It is efficient and has an adaptive learning rate.

7- Evaluation of All Variants

Performance Metrics:

The following performance metrics were observed when running the three models through the evaluation script developed. Please note that these metrics were measured using build in sklearn.metrics libraries. The following table highlights the performance of the three models: Main Model, Variant1, and Variant2.

Table 1: Performance measurements of the three models

| Model | Macro | | | Micro | | | Accuracy |
|------------|-------|------|------|-------|------|------|----------|
| | P | R | F | P | R | F | |
| Main Model | 0.56 | 0.56 | 0.55 | 0.59 | 0.59 | 0.59 | 0.59 |
| Variant 1 | 0.44 | 0.47 | 0.44 | 0.51 | 0.51 | 0.51 | 0.51 |
| Variant 2 | 0.51 | 0.52 | 0.49 | 0.56 | 0.56 | 0.56 | 0.56 |

As can be seen in the table, the main model provided the best overall performance at both the micro and macro level. Variant 1, which was the first base model that was tested, clearly underperformed. No macro metric was found to be over 50%. Variant 2 had an additional layer added to Variant 1, which showed improvement overall to the performance of the model. Both precision and recall are over 50%, with the F1-score almost at 50% at the macro level.

Confusion Matrix Analysis

The Confusion Matrices for the Main Model, Variant1, and Variant 2 can be seen in figures 4, 5 and 6 respectively. Looking at the four classes, the most confusion occurred between the Focused and neutral classes across all three models. This was also observed during experimentation of the different kernels and layers as well. There is also a decent amount of confusion between happy, neutral, and focused. This does make sense as a neutral expression would have the least amount of distinctive facial features. The other classes training data images would need to distinguish themselves from the neutral class. This is consistent with the better prediction on the surprise class as these images often have very clear signs on the face, such as wide eyes or a wide-open mouth.

Another key observation is the poor prediction of the Focused class across all three models. The data for neutral, happy, and surprised was all able to be procured through existing datasets, and as such there was an abundance of images. This meant that we were able to hit our max samples of 800 across training, testing and validation sets. However, the Focused class was limited to 500 total images of the three sets. This potentially introduced a bias against this class. Also, the nature of the focused tells are the face are much more subtle. In choosing the images, an attempt was made to choose faces with a furrowed brow, squinting eyes, hand touching around the temple, or hand under the chin. A complete overhaul of this class's dataset is under consideration.

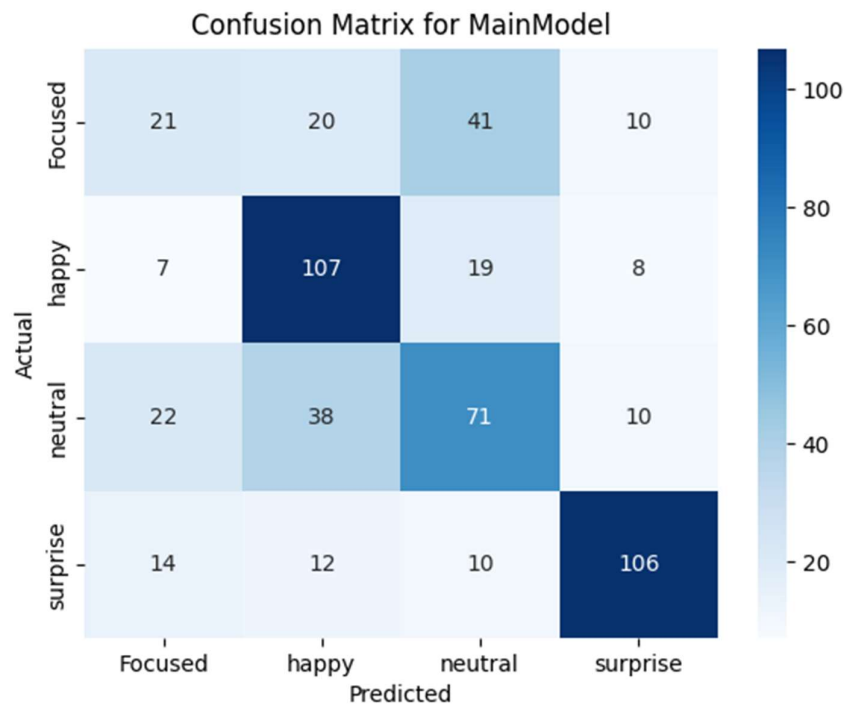


Figure 4: Confusion Matrix for Main Model

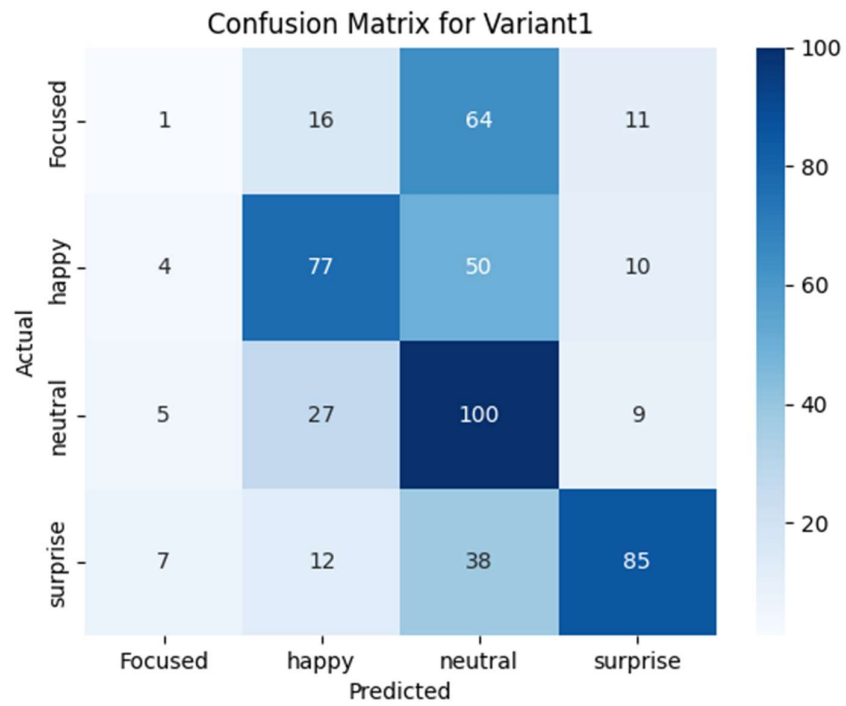


Figure 5: Confusion Matrix for Variant1

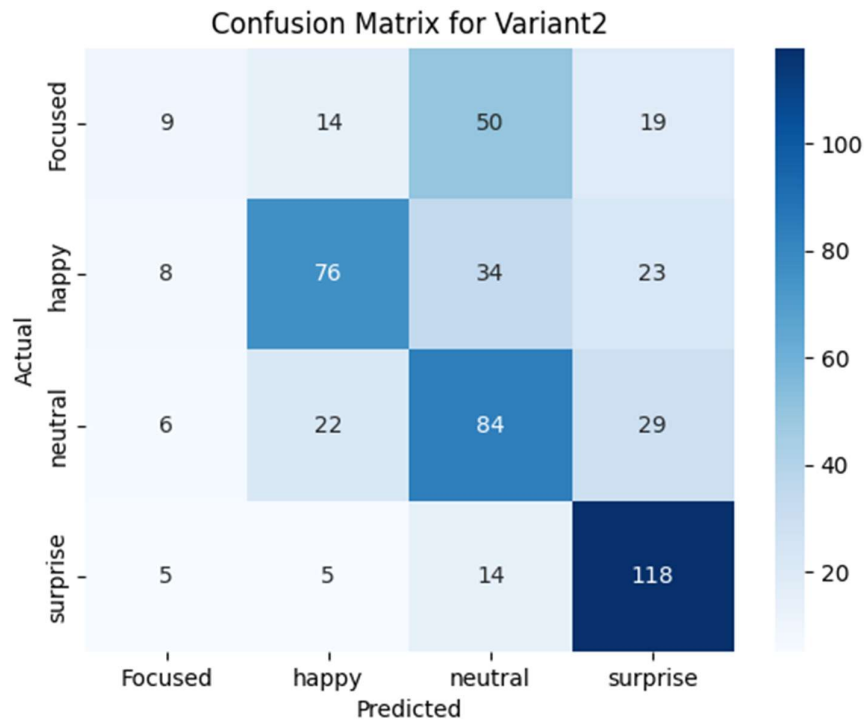


Figure 6: Confusion Matrix for Variant2

Impact of Architectural Variations

During experimentation with different layers, the main comparison will be done between Variant1 and Variant 2 as they have identical kernels otherwise. Variant 2 had better performance metrics overall, indicating that the additional layers did assist with better prediction. This meant that there were more detailed features potentially recognized by the model with more layers. However, an argument could be made that there was some overfitting as compared to variant1. Looking at the surprise confusion matrix of Variant 2, we can see that there are a lot more images predicted to be in this class, both correctly and incorrectly.

The impact of the kernel size was a lot harder to observe. A lot of experimentation was done with different kernel sizes on the first layer with minimal change in performance. However, when the kernel of the max pool was changed from 2x2 to 3x3, this appeared to have the best performance. Combining this with a first layer kernel of 5x5 appears to have allowed the Main Model to capture more broad features, which helped particularly with the Focused class.

Changes Made for Part III

Before any testing for bias, changes were made to the dataset to add additional images to the Focused class for training as this was one of the identified issues in the previous. This did impact some of the performance metrics, and as such the performance of the main model. These will be presented in the next section before the evaluation of using k-fold training methodology.

8- Main Model Evaluation using K-Fold

Updated Performance of Main Model

Below is the performance of the main model after an alteration of the dataset. There was a 50% increase in the number of images in the Focused class. The dataset was also altered across the board to be more representative of each class. Previously, the first 800 images of the entire downloaded dataset were being chosen in the happy, neutral, and surprise. More representative images were chosen for each class instead. AS can be seen, a more consistent performance across all metrics was measured

Table 2: Updated metrics for Main Model with new dataset

| Main Model | Macro | | | Micro | | | Accuracy |
|------------|-------|------|-----|-------|------|------|----------|
| | P | R | F | P | R | F | |
| | 0.61 | 0.59 | 0.6 | 0.59 | 0.59 | 0.59 | |

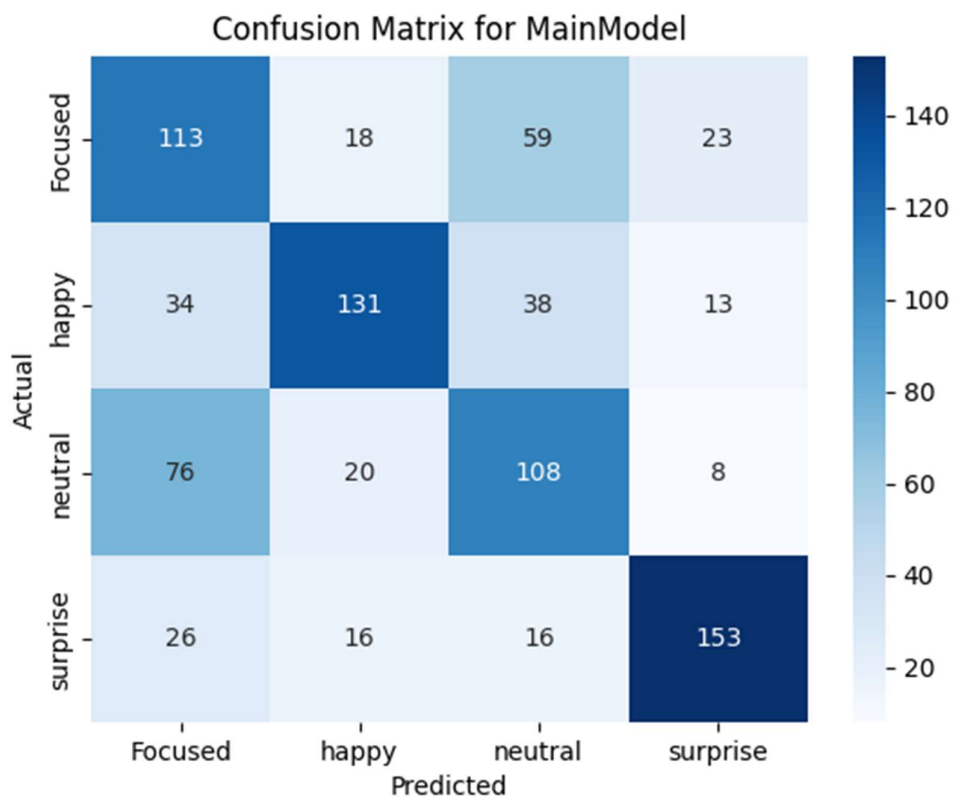


Figure 7: Updated Main Model Confusion Matrix

K-Fold evaluation of Main Model from Part II

Table 3 shows the evaluation using 10-fold Cross evaluation of the final model.

Table 3: K-Fold Cross Validation based on training of the same CNN of Main Model (final) with same dataset

| Fold | Macro | | | Micro | | | Accuracy |
|----------------|-------|------|------|-------|------|------|----------|
| | P | R | F | P | R | F | |
| 1 | 0.62 | 0.58 | 0.57 | 0.58 | 0.58 | 0.58 | 0.58 |
| 2 | 0.69 | 0.57 | 0.50 | 0.55 | 0.55 | 0.55 | 0.55 |
| 3 | 0.69 | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 |
| 4 | 0.68 | 0.67 | 0.67 | 0.68 | 0.68 | 0.68 | 0.68 |
| 5 | 0.66 | 0.64 | 0.62 | 0.66 | 0.66 | 0.66 | 0.66 |
| 6 | 0.66 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 |
| 7 | 0.64 | 0.69 | 0.61 | 0.64 | 0.64 | 0.64 | 0.64 |
| 8 | 0.71 | 0.69 | 0.70 | 0.69 | 0.69 | 0.69 | 0.69 |
| 9 | 0.60 | 0.61 | 0.60 | 0.61 | 0.61 | 0.61 | 0.61 |
| 10 | 0.63 | 0.63 | 0.62 | 0.63 | 0.63 | 0.63 | 0.63 |
| Average | 0.66 | 0.63 | 0.62 | 0.63 | 0.63 | 0.63 | 0.63 |

Looking at the cross-validation results of the table, this is some variance in performance depending on the fold. This does indicate that depending on the segment of data that is involved in the training of the model, the performance metric can be affected by up to 11%. While an attempt for consistent results was made by saving the testing image indices used during training, and only evaluating on loaded indices and model folds, for an undetermined reason subsequent runs of the k-fold evaluation script would result in different performance metrics. This does further support the conclusion that different segments during training lead to different performance outcomes.

No second table is presented as the model that was carried over from Part II is the same CNN structure. Only the data used to train the model was changed.

9- Bias Detection and Analysis

Introduction

As the team size was less than three, only one bias attribute was analyzed as per the instructions for such team sizes. The attribute chose was the gender attribute which was split into three categories: male, female and non-binary. To perform the bias detection, the first step was to split the data in each emotional class to its appropriate gender. This was done manually in the dataset folders. To evaluate the data with the existing model, the gender was put as the outer folder, with the classes containing images of only that gender within.

A split method identical to that used to measure the performance of the overall model was used. This means that regardless of the dataset size within a given gender, 15% of the images were used for validation. No training was redone initially on the model.

Bias Detection Results

Running the evaluation of the data that was split according to gender, the following performance metrics were measured.

Table 4: Bias Analysis results on Gender split data

| Attribute | Group | Accuracy | Precision | Recall | F1-Score |
|-----------|------------|----------|-----------|--------|----------|
| Gender | Male | 0.67 | 0.67 | 0.67 | 0.67 |
| | Female | 0.72 | 0.71 | 0.71 | 0.7 |
| | Non-Binary | 0.66 | 0.67 | 0.65 | 0.65 |
| | Average | 0.68 | 0.68 | 0.68 | 0.67 |

Based on the results, it does not appear that the model is detecting any bias. This was odd as the images categorized as non-binary were severely underrepresented. This could be a result of overfitting, or that the images that are used in the non-binary categorization represented the emotion well, producing the results seen.

Something that was noticed during the manual split of the dataset was that in the images, there was no clear indicator that some one may be non-binary. Most images had clear identifiers of either a male or female. This may be an inherent bias that existed during the dataset distribution, even to humans, there are no clear physical signs that can make a non-binary identification easy from a still image of the face only. Non-binary individuals can be easily mistaken for either of the other genders, which is why they typically identify their pronouns. While that is beyond the scope of this class, the reason this was mentioned is because we are meant to take an ethical approach. For there to be accurate data for non-binary individuals or genders overall, I believe there should be pre-labelling of the image.

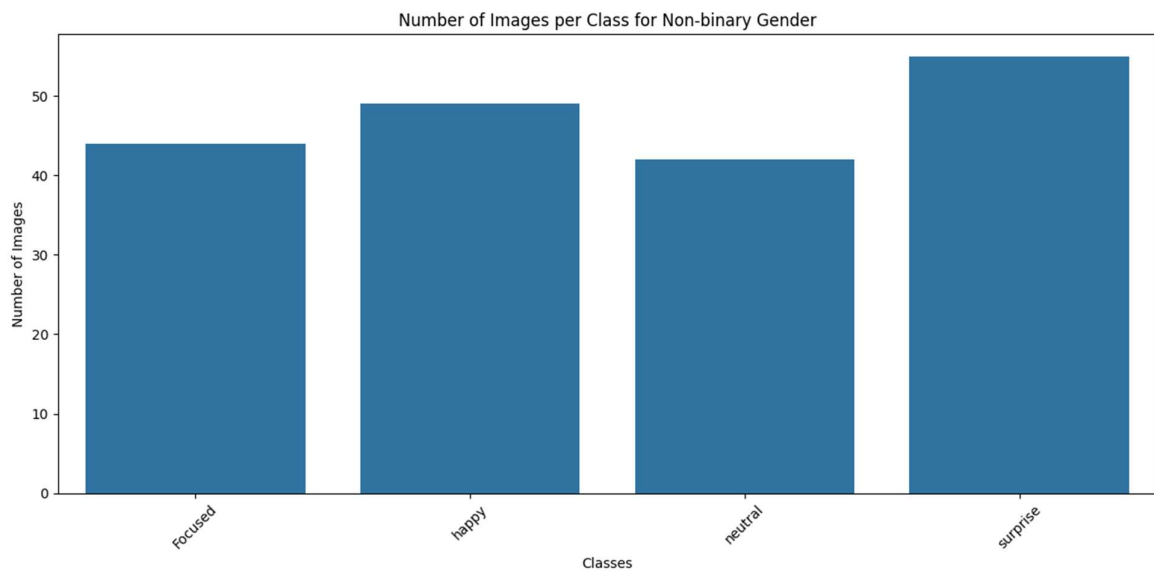
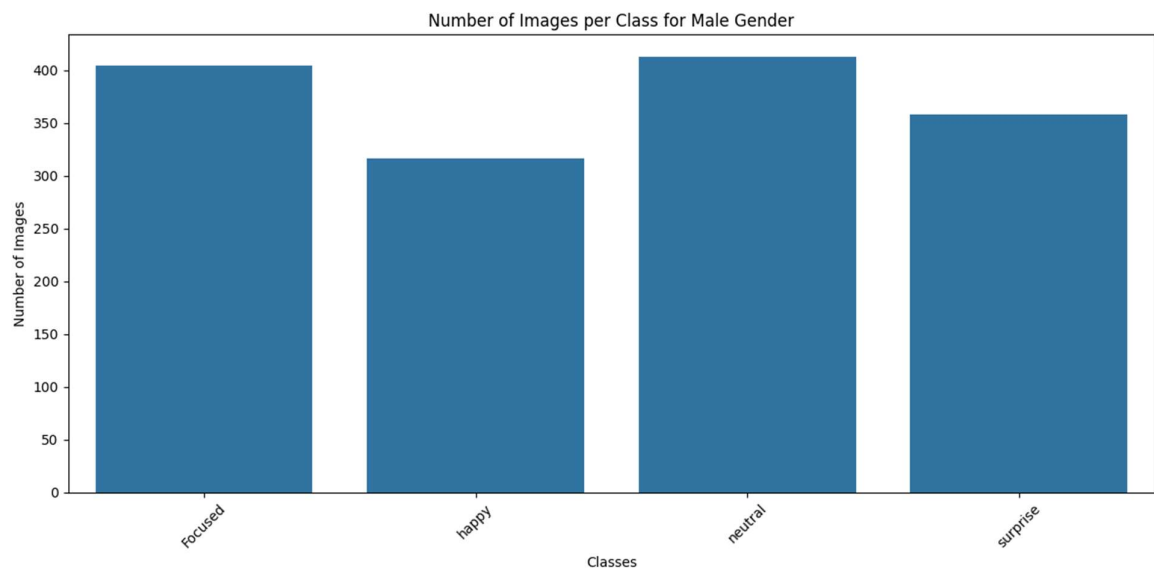
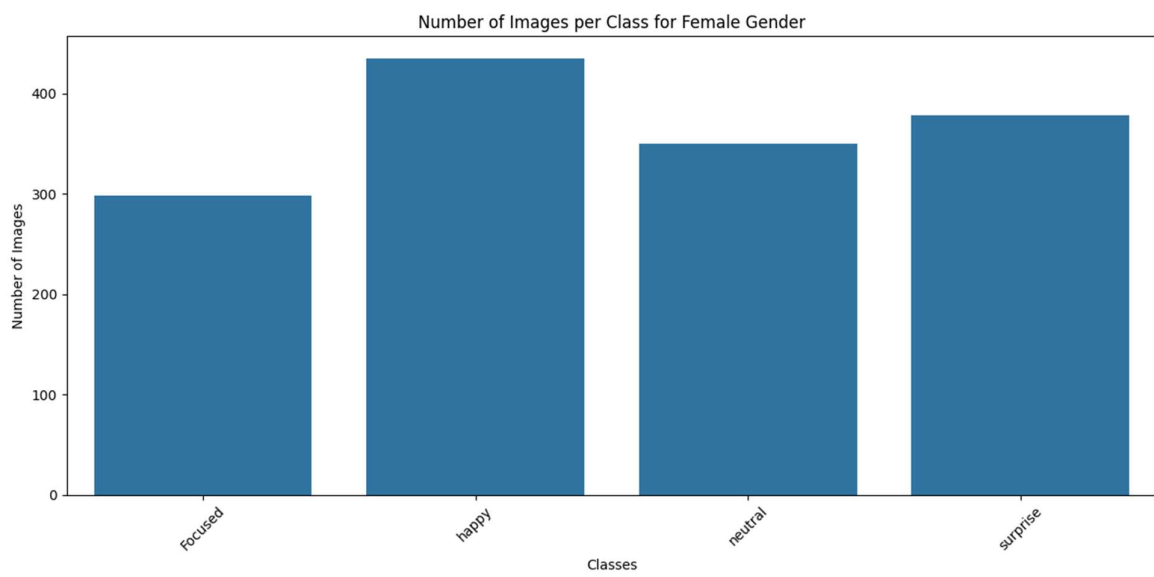


Figure 8: Number of images per gender of each Class per gender of initial split.

Bias Robustness Check

Despite the under-representation of images categorized as non-binary, it is concluded that the model is running without bias with regards to gender. To verify the model's response to bias, a bias robustness check was performed with three different levels of bias introduced. For each level, the male gender was used to underrepresent it relative to the female gender. For level 1, 15% of the original images within the male class were removed. The new dataset was used to retrain the Main Model. Using the same data used to make the original evaluation, the new evaluation metrics were recorded. A similar process was repeated for levels 2 and 3, where 30% and 50% of the male images were removed respectively. The tables below represent the findings:

Table 5: Male metrics with various biases introduced in the dataset.

| Male | | | | |
|------------------|----------|-----------|--------|----------|
| Bias Level | Accuracy | Precision | Recall | F1-Score |
| Level 1 | 0.59 | 0.6 | 0.6 | 0.58 |
| Level 2 | 0.62 | 0.67 | 0.61 | 0.62 |
| Level 3 | 0.53 | 0.55 | 0.54 | 0.54 |
| Original Dataset | 0.67 | 0.67 | 0.67 | 0.67 |

Table 6: Female metrics with various biases introduced in the dataset.

| Female | | | | |
|------------------|----------|-----------|--------|----------|
| Bias Level | Accuracy | Precision | Recall | F1-Score |
| Level 1 | 0.69 | 0.66 | 0.66 | 0.65 |
| Level 2 | 0.66 | 0.67 | 0.61 | 0.62 |
| Level 3 | 0.54 | 0.54 | 0.54 | 0.53 |
| Original Dataset | 0.72 | 0.71 | 0.71 | 0.7 |

Table 7: Non-Binary metrics with various biases introduced in the dataset.

| Non-Binary | | | | |
|------------------|----------|-----------|--------|----------|
| Bias Level | Accuracy | Precision | Recall | F1-Score |
| Level 1 | 0.69 | 0.69 | 0.68 | 0.66 |
| Level 2 | 0.59 | 0.68 | 0.58 | 0.6 |
| Level 3 | 0.66 | 0.67 | 0.66 | 0.64 |
| Original Dataset | 0.66 | 0.67 | 0.65 | 0.65 |

Upon review of these tables, it appears that despite the performance changes across all genders, there is no clear bias identified. The drop in performance for level 3 can be attributed to the removal of a significant portion of the data. It seems clear that the display of emotion is what determines the output regardless of gender. Using greyscale images with the preprocessing that is being done seems to keep the performance steady across genders.

Should bias have been introduced, the mitigation strategy was to generate synthetic images. This was to be done using the Keras library, which is part of the Tensorflow plugin. A script, `synthetic_data.py`, was written which balanced all the gender to have a similar number of images

per class (approximately 400 per emotion class). The images are generated by taking the existing images and distorting them in some manner, such as rotation or stretching a certain part of the image. The effects of using such data were not thoroughly analyzed, but there is a possibility that it may cause overfitting.

10 - Conclusion:

The dataset visualization analysis provides valuable insights into the dataset's content and characteristics. The balanced class distribution ensures that the dataset is suitable for training a facial expression recognition model. Additionally, the sample images highlight the diversity of facial expressions present in the dataset, while the pixel intensity distribution reveals variations in lighting conditions among images. Overall, these visualizations aid in understanding the dataset's content, identifying potential anomalies, and informing preprocessing steps for model training.

With regards to the models, the Main Model was chosen to be the best based on its performance metrics and confusion matrix analysis. This was attributed to the use of the 3x3 kernel for the max pooling, and well as a starting 5x5 kernel for convolution layer 1. While the training strategies do seem sound, seeing various models work with the dataset put together in part 1 has shown some potential flaws in the training data. If this data is cleaned up, perhaps additional layers on the Main Model chosen can help identify fine facial features to further increase the prediction rate.

From part two to part three, the dataset was cleaned up. Images more representative of the emotion were chosen for each class as opposed to using the images that happened to be listed first in the dataset. More images were added to the Focused class to bring it closer to the number of images in the other classes. This was to help alleviate any potential issues that may have been caused by the under representation of this class.

Evaluation of the main model using k-fold cross validation showed that certain segments of the data set may not be distinguished enough in terms of the display of emotion. On average, the model did perform at an acceptable level for each, on par or above the measured performance using the scikit learn functions.

The model used also appears to be fair based on the bias robustness check that was done. There was no significant deviation in the performance of one gender over another. Time permitting, it would have been interesting to see if another attribute such as age or race may have had bias introduced. It is safe to conclude that with respect to gender, the model is performing fairly and reliably.

Appendix

Expectations of Originality

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 citation¹. 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
Name: Sherief Soliman
Signature: 

Instructor: Dr. René Witte
I.D. #: 29248323
Date: March 11, 2024

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

GitHub Repository Link

https://github.com/SheriefS/COMP472_Project/

References

- [1] Jonathan Oheix. "Face Expression Recognition Dataset". Kaggle, 2021. [Online]. Available: <https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset/data>. [Accessed: March 9, 2024].
- [2] Ananthu. "Emotion Detection FER". Kaggle, 2021. [Online]. Available: <https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer>. [Accessed: March 9, 2024].
- [3] "LabelImg". Github, 2024. [Online]. Available: <https://github.com/tzutalin/labelImg>. [Accessed: March 9, 2024].
- [4] "OpenCV Documentation". OpenCV, 2024. [Online]. Available: <https://docs.opencv.org/4.x/index.html>. [Accessed: March 9, 2024].
- [5] "Matplotlib Documentation". Matplotlib, 2024. [Online]. Available: <https://matplotlib.org/stable/contents.html>. [Accessed: March 9, 2024].
- [6] "scikit-learn Documentation". scikit-learn, 2024. [Online]. Available: <https://scikit-learn.org/stable/documentation.html>. [Accessed: March 9, 2024].
- [7] J. Brown. "Understanding CNNs: From Feedforward to Deep Learning". Springer, 2019.
- [8] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". Nature, 2015. [Online]. Available: <https://doi.org/10.1038/nature14539>. [Accessed: March 9, 2024].
- [9] "Pytorch Documentation". Pytorch, 2023. [Online]. Available: <https://pytorch.org/docs/stable/index.html>. [Accessed: March 30, 2024]