A.I.ducation Analytics using Conventional Neural Network (CNN)

Zahra Pezeshki (40288066) - Data Specialist
Hema Reddy Muppidi (40236911) - Training Specialist
Oluwadamilola Okafor (40224938) - Evaluation Specialist

COMP 6721 Applied Artificial Intelligence Project Report
Fall 2023
Group: AK_ 1
GithubLink:https://github.com/Dami-Lola/A.I.ducationAnalytcs

Introduction

Facial expression recognition is one of the most popular applications of deep learning. Convolutional Neural Networks (CNNs) are a type of deep neural network used extensively for facial expression recognition [2].

This project aims to develop a Deep Learning Convolutional Neural Network (CNN) that analyses images of students in a classroom or online meeting setting and categorises them into distinct states or activities. The system analyses images in four different classes:

- **Neutral**: A student presenting neither active engagement nor disengagement, with relaxed facial features
- **Focused**: A student evidencing signs of active concentration with sharp and attentive eyes
- **Bored**: A student displaying signs of weariness or a lack of interest. This can be evidenced by droopy eyes or vacant stares.
- **Angry**: Signs of agitation or displeasure, which might manifest as tightened facial muscles, a tight-lipped frown, or narrowed eyes.

Dataset

There are a total of about 2800 images. The training dataset consists of about 2,200 images and the testing dataset consists of about 570 images. For each class, Neutral has a total of about 700 from Source 1 [2], Focused has a total of about 300 from Source 2 [4] and 250 from Source 3 [8], Bored has a total of about 860 from Source 1 [2], and Angry has a total of about 700 Source 1 [2]. The dataset for Neutral, Bored and Angry consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image [3]. Bored facial expressions are composed of multiple classes from different datasets mapped to substitute those in this class for training in the system. As for the dataset for Focused, it consists of colourful images with facials that express the individuals being attentive, observant, engaged and existing. The datasets were not readily available, so we had to search and find images that fit the class manually. The expressions and attributes in this dataset are similar to what you might encounter in real-world scenarios. The dataset includes some of the most common emotions people experience and express through facial expressions. The images are primarily frontal face shots, a common scenario for facial expression recognition applications.

The dataset was chosen because it is one of the most popular datasets for facial expression recognition and has been used extensively in research. It is also publicly available on Kaggle, making it easily accessible to researchers and developers. One of the challenges of using this dataset is that the images are relatively low resolution, making it difficult to classify some emotions accurately. Also, the dataset does not provide information on the individuals' ethnicities, ages, or genders in the images. Therefore, whether the dataset represents a diverse range of expressions, ethnicities, ages, and genders is unclear. However, it is one of the most popular datasets for facial expression recognition and has been used extensively in research [3]. These challenges can be overcome by using more advanced deep-learning techniques and architectures and improving facial expression recognition accuracy.

Update to Dataset

During the process of the third part of the project, the datasets were reduced to 550 each in total for each class. This still meets the minimum requirements in total for all classes which is 2000. This reduction was due to the manual segmentation and labelling of the dataset. There was a time constraint to labelling the initial datasets, as such an agreed amount was done. More images were eventually added to each class based on the bias analysis results.

The table below displays the total number of each class after the bias analysis

S/N	Classes	Source 1	Source 2	Source 3	Total
1	Neutral	570	0	0	570
2	Focused	0	210	340	550
3	Bored	580	0	0	580
4	Angry	560	0	0	560

Table 1: Distribution of training images

Data Cleaning

In the data cleaning, Keras was used for data augmentation. This class provides a variety of data augmentation options. We performed random rotations on the images. This was useful for introducing variation and making the model more robust to different orientations of objects in the images. Adjusted the brightness of images within a specified range. Resized the images to a specific size (48x48 pixels). Resizing is a common standardisation technique, ensuring that all images have the same dimensions. We set the colour to "grayscale," converting the images to grayscale. This standardises the colour format and reduces the dimensionality of the images. The class mode was set to 'categorical,' as we were working with a multi-class classification problem, and the labels were one-hot encoded. A directory path was specified where augmented images will be saved. This can be useful for inspecting the augmented data. Also, setting shuffle to true, the images during training, introducing randomness to the order in which images are presented to the model.

The perception of suitable brightness can be subjective and may vary from person to person. We faced a significant challenge when determining the proper brightness for the images. Images in a dataset may have been captured under different lighting conditions, leading to variations in brightness. It can be challenging to determine a one-size-fits-all brightness adjustment. Excessive brightness adjustments can result in information loss. An image must be sufficiently brightened to maintain essential details and become usable. In some cases, altering the brightness of an image may change its context or meaning, which could be problematic in specific applications. Careful consideration was given to the potential impact of brightness adjustments on the dataset's context. Data augmentation techniques were employed to simulate different lighting conditions. By introducing variations in brightness during augmentation, the model became more robust to different lighting scenarios.

Figure 1 Before-After Data Augmentation

Labelling

Roboflow Annotate is a self-serve image annotation tool that allows you to label training data quickly and export it to any format [5][6][7]. It was used to label and classify the images into training and testing folders. We had to upload each dataset class first and then specify the percentage for the training and for the test. The platform automatically distributes the dataset in their respective folders. Datasets for angry and neutral facial expressions were readily available and thus were uploaded once the agreed total number was determined.

Clear annotation guidelines and documentation are crucial to reducing ambiguities. The data specialist in the team provided explicit instructions on how to label different data instances. Data labelling is often an iterative process. There was an open line of communication between every team member for addressing ambiguities, that is seeking clarifications when anyone encountered ambiguous cases. The ambiguity we encountered was mapping different classes to suit the bored facial expression class. We had to decide what other class could fit this facial expression.

Rigorous quality control was done when merging datasets or mapping classes. This included reviewing a subset of labelled data to ensure that the mapping and merging processes do not introduce errors. Challenges arose when different datasets had distinct class definitions and when classes were imbalanced. The bored dataset had class mapping of sad and disgusted datasets as the facial expressions fit the description for that emotion. When merging multiple datasets,we had to manually scan through the images to determine if the images suited the class and add to the class folder. The focused dataset, however, posed a tedious challenge as we all had to search through the web to find facial expressions from different available sources. Mapping and merging had to be carefully done to avoid losing information and introducing bias.

After the number of dataset for each class was agreed upon, each class folder was uploaded into Roboflow. Gender and Age were the agreed attributes to analyse for bias. Annotation was created for all the groups under the chosen attributes. Each image was viewed and annotated based on the group they seemed best fit to fall under. Other/Non-Binary was added to the gender group for images that we could not identify the gender. After the long annotation session, a csv containing all the labels for the images was generated along with images for each class folder. The filename in the csv file is the same as the imagename, this helps for easy identification of the attributes for each image. The classes.csv was converted to updatedclasses.csv to have all the groups as text under each attribute. This same process was used for additional images after the bias analysis results.

Dataset Visualization

Class Distribution

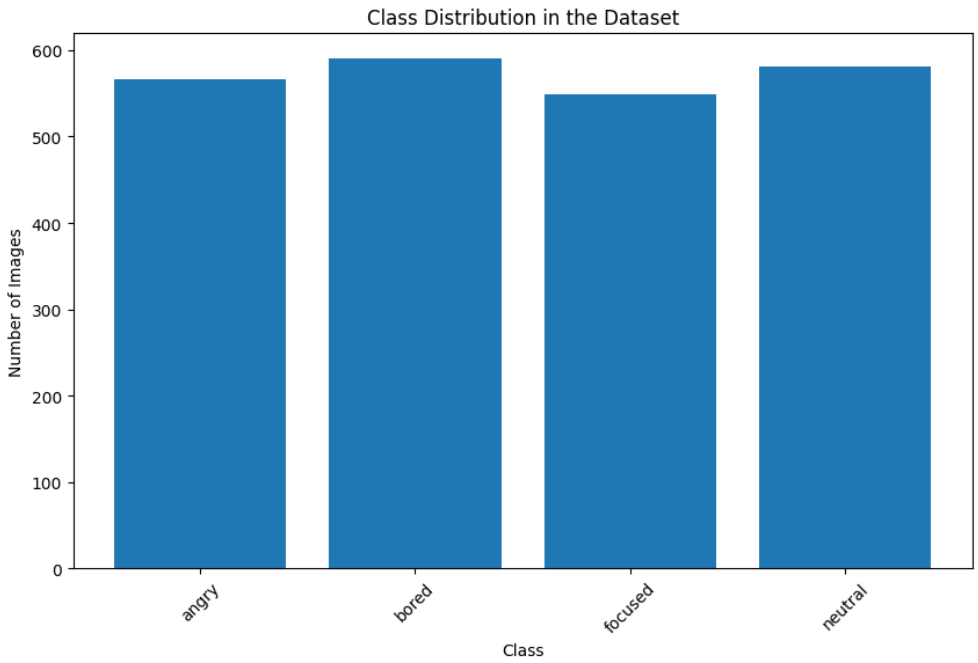


Figure 2 Class Distribution

The bar graphs show the number of images in each class, helping to understand data distribution across different emotional states (neutral, engaged, bored, angry). This shows a significant disparity in the number of images across the four classes. This indicates a class imbalance, where some classes have more data than others. Class imbalance can have a notable impact on model performance. Machine learning models tend to perform better when they have a balanced dataset, where each class has roughly an equal number of samples. This balance helps the model learn patterns and make predictions more effectively. In cases of class imbalance, the model may become biased toward the majority class, potentially leading to poor performance for minority classes. However, in our model and application, we will apply various strategies to address class imbalance. These strategies could include oversampling the minority classes ("focused" and "angry"), and undersampling the majority class ("bored").

Sample Images

Figure 3 Collection of 25 images in a 5 by 5 grid.

From the image collections, it was concluded that the datasets were labelled according to their intended classes. Very little or no mismatch in the images classifications.

Pixel Intensity Distribution

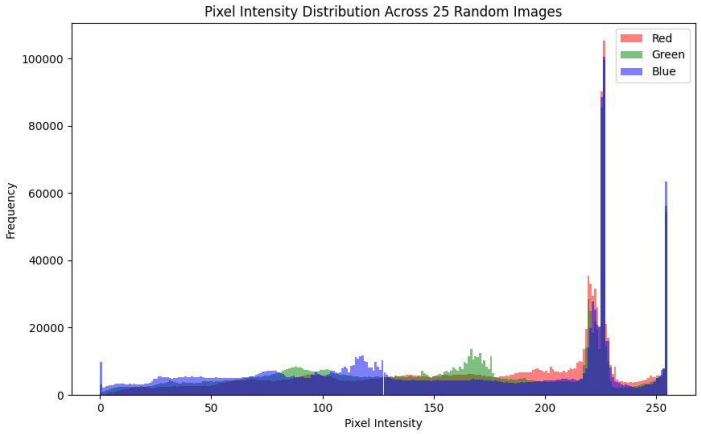


Figure 4 Pixel Intensity Distribution

Analysing the pixel intensity distribution provided insights into the nature of the images in the dataset. We observed significant variations in the pixel intensity histograms, especially for different colour channels (Red, Green, and Blue), which suggested that images in our dataset have varying lighting conditions. Some images might have been well-lit, while others might be underexposed or overexposed. This could affect the model's ability to generalise to different lighting conditions. Also, certain classes have a broader range of pixel intensities than others, which indicates that some classes contain images with more extreme lighting conditions. In this case, we considered preprocessing techniques like contrast adjustment to standardise lighting conditions across all images.

Additionally, analysing pixel intensity distributions helped us understand the lighting conditions in our images and make informed decisions about preprocessing techniques that could help enhance the model performance.

CNN Architecture

Model Overview and Architecture Details

The CNN architecture used in this project is a network made up of four convolutional layers followed by one fully connected linear layer. We arrived at this architecture choice by experimenting with the various number of layers, testing the accuracy of these network architectures, and selecting the best one.

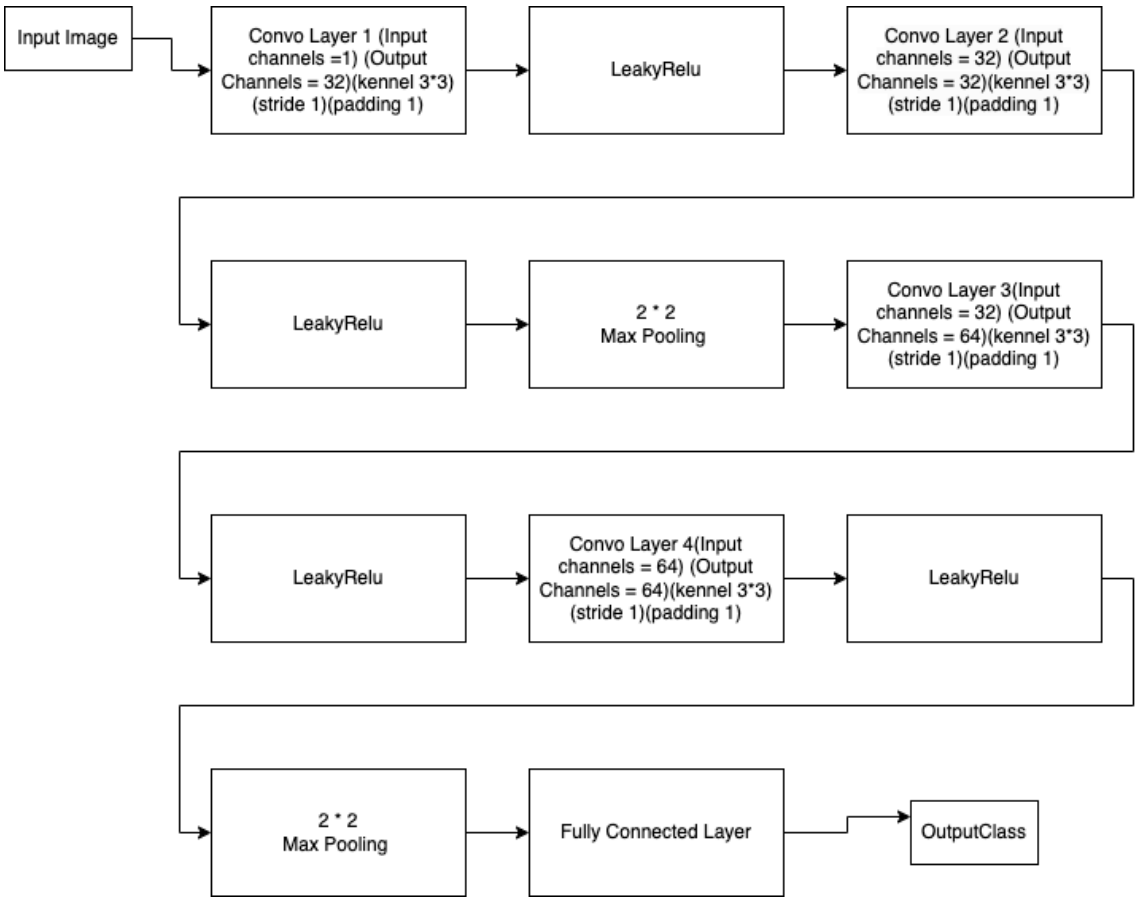


Figure 5: Diagram of CNN Architecture

Activation Function

Leaky ReLU is used as the activation function after each convolutional layer. This non-linearity allows a small, non-zero gradient when the input is negative, helping to avoid dead neurons during training.

Unique Features

The model uses a combination of convolutional layers and max pooling for feature extraction from the input images, followed by fully connected layers for classification. The use of batch normalization helps in stabilizing and accelerating the training process.

Training Process

There were several hyper-parameters that were experimented with, and tuned to produce our final CNN architectures, namely:

Number of Epochs: The model is trained for 15 epochs, iterating over the entire dataset 15 times. We initially started with with 10 epochs which were chosen randomly, recorded the test accuracy, and continued to increment the number of epochs till we finally settled on 15, after which we were not enhancing the accuracy.

Learning Rate: The learning rate is a hyperparameter that controls the step size during optimization. It controls how much the model changes in response to the estimated error each time the model weights are updated [9]. Too small a learning rate may mean our model takes too long to train or gets stuck, while too high a learning rate produces an unstable training where the weight updates swing around wildly. A value of 0.0001 was initially tested with, however we discovered that the loss was fluntualty as when we tested the model using 0.01. So we decided it was best to use a learning rate of 0.01 and the Adam optimizer adapts the learning rates of each parameter individually where we achieved better accuracy results.

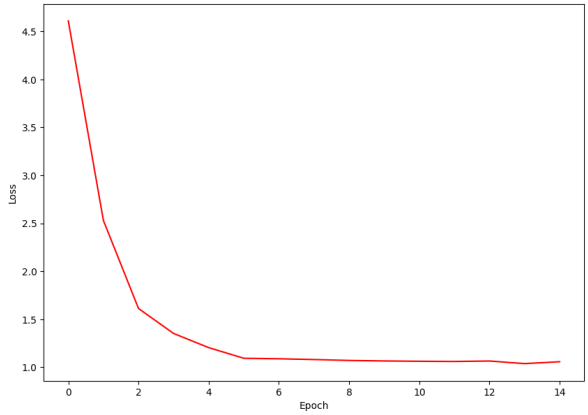
Loss Function: CrossEntropyLoss is a suitable choice for multi-class classification problems. It combines softmax activation and negative log-likelihood loss, making it well-suited for optimizing classification models.

Kernel Size: The kernel is simply the convolutional filter that is passed over our images during the convolution settled on 8, after which we were not enhancing the Accuracy.

Optimizer: The Adam optimizer is used to minimize the loss function. It is an adaptive optimization algorithm that adjusts the learning rates for each parameter individually based on their past gradients.

Mini-Batch Gradient Descent: The training is performed using mini-batch gradient descent. This involves updating the model's weights based on a small subset (mini-batch) of the training data in each iteration. This approach is computationally more efficient than processing the entire dataset in one go.

This training loop includes the optimization process, validation accuracy monitoring, and saving the model if the validation accuracy improves. The test set is then used to evaluate the final accuracy of the trained model.



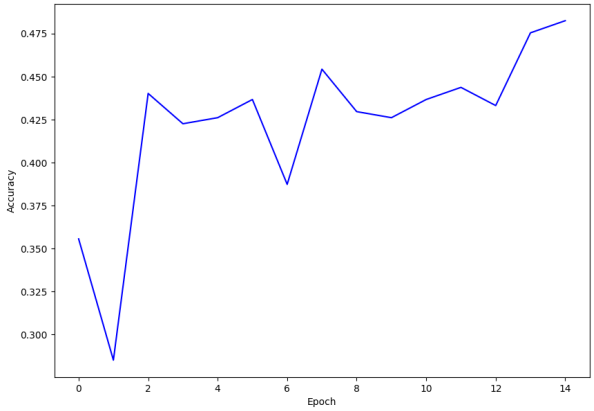


Figure 6: Running loss Main Model

Figure 7: Accuracy Projection of the Main Model

Comparison With Architecture Variants

Our main network, which we refer to as a “MainModel” consists of 4 convolutional layers and 1 fully connected layer. For the sake of comparison, we documented two variants of our base model, which should have:

- Adding and removing convolutional layers in one network
- Higher kernel sizes if 7 x 7 and lower kernel sizes of 2 x 2 in the second network.

We trained both new variant networks with the image data as before and saved the trained models for these networks. To select the right fit for the first variant, we tested with removing 1 convolutional layer from the existing main model and adding 1 convolutional layer from the existing main model. We compared and analysed both results to come to the conclusion that adding 1 convolutional layer produced better results than removing. Regarding the second variant, we tested with kernel sizes of 2*2 and 5*5 separately from the existing main model. When adjusting the kernel size to 5*5 performed better in terms of the resulting accuracy.

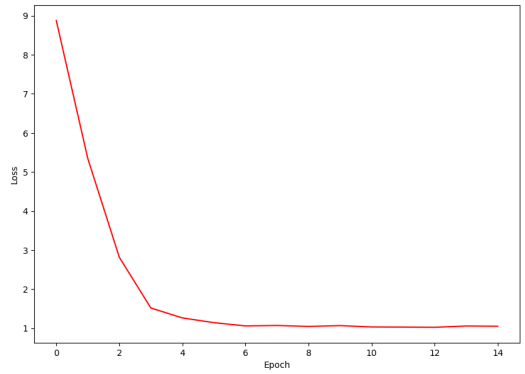


Figure 8: Running loss Variant 1

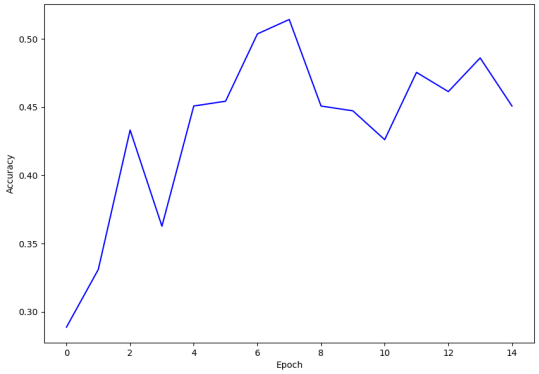


Figure 9: Accuracy Projection Variant 1

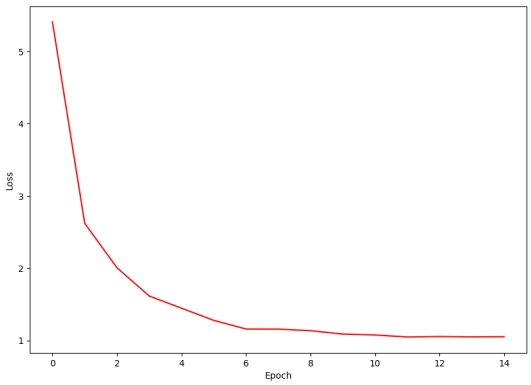


Figure 10: Running loss Variant 2

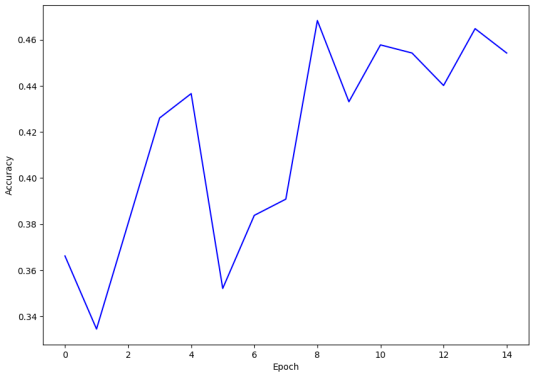


Figure 11: Accuracy Projection Variant 2

Changes to the Architecture

The main model was updated to have 6 convoclutional layers, maxpooling at the 5th and 6th layer and a dropout functions on the main moodel. The learning rate was reduced to 0.0001. The **early stopping** technique was employed evaluate the model on the validation set and monitor the loss. When the validation loss stops decreasing and begins to increase, this was a strong indicator of overfitting, as such, put a halt from further training. All these changes were made to get better accuracy.

Evaluation

Performance Metrics

Main Model: Has the highest precision, recall, and F1-measure among the three models, both in macro and micro averages. The relatively high precision indicates that when the model predicts positive, it is likely correct. The balanced precision and recall values suggest a model that is making well-rounded predictions without skewing heavily towards false positives or false negatives.

Variant 1: Has lower precision, recall, and F1-measure compared to Main Model. A lower F1-measure indicates that there might be an imbalance between precision and recall, or the model struggles with both false positives and false negatives. The micro-average values are lower than the macro-average values, indicating that the model may struggle more with individual instances than with overall trends.

Variant 2: Has the lowest macro F1-measure and precision among the three models. Despite having relatively high micro precision and recall, the micro F1-measure is not as high, suggesting a trade-off between precision and recall. The relatively low macro F1-measure indicates that the model might not generalize well to the entire dataset.

Implications for Facial Image Analysis

In facial image analysis, precision and recall are crucial metrics. Higher precision is desirable to minimize false positives (misclassifying non-faces as faces), while higher recall is crucial for capturing as many actual faces as possible. Main Model might be preferred due to its balanced precision and recall.

Model	Marco			Micro			Accuracy
	P	**R**	**F**	**P**	**R**	**F**	
Main Model	0.5	0.49	0.48	0.48	0.47	0.45	0.47
Varient 1	0.44	0.47	0.42	0.42	0.44	0.4	0.44
Varient 2	0.43	0.44	0.37	0.46	0.46	0.4	0.44

Table 2: Performance Matrix Table

Confusion Matrix Analysis

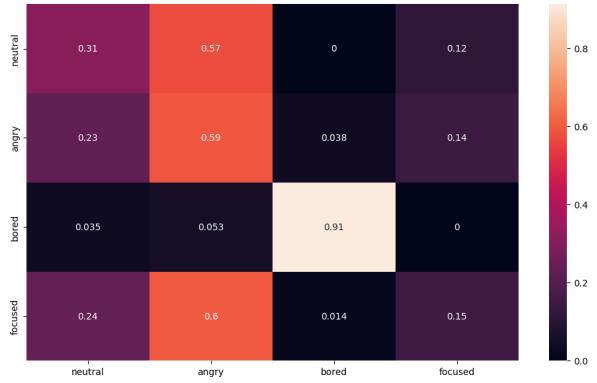


Figure 12: Confusion Matrix Main Model

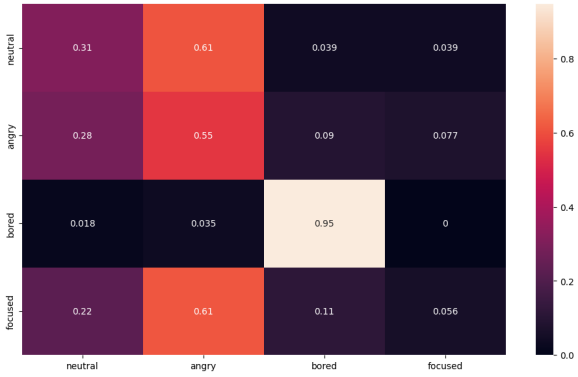


Figure 13: Confusion Matrix Variant 1

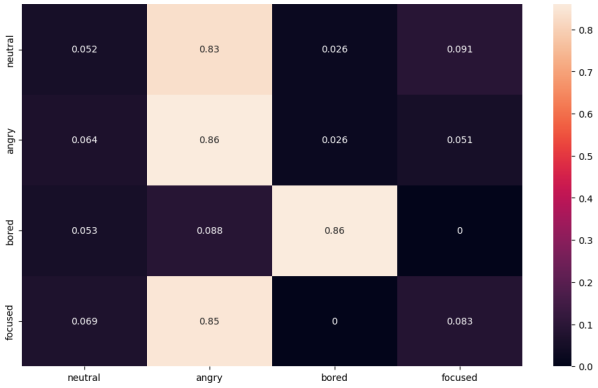


Figure 14: Confusion Matrix Variant 2

Most Frequently Confused Classes

Main Model: Neutral and Angry are often confused (0.57 misclassification rate). Focused is also frequently confused with Angry (0.6 misclassification rate).

Variant 1: Neutral and Angry are frequently confused (0.61 misclassification rate). Bored is confused with Angry (0.09 misclassification rate).

Variant 2: Neutral is often confused with Angry (0.83 misclassification rate). Bored is confused with Neutral (0.088 misclassification rate).

Speculations on Misclassifications

Neutral and Angry Confusions: Facial expressions for neutral and angry might share visual similarities, making it challenging for the models to distinguish between them.

Focused and Angry Confusions: The emotional expressions associated with focus and anger might share similar facial features, leading to misclassifications.

Dataset Quality: The dataset might have instances with ambiguous or mixed emotional expressions, making it challenging for the models to accurately classify certain samples.

Well-Recognized Classes:

Bored: **Main Model** and **Variant 1** show high precision for classifying boredom, with 0.91 and 0.95, respectively. It suggests that the models are relatively successful in identifying the bored class.

Focused: **Variant 2** shows a high precision of 0.86 for the focused class, indicating success in distinguishing focused expressions from other emotions.

The misclassifications might be due to inherent ambiguities in facial expressions, requiring more diverse training data or advanced techniques to handle subtle variations in emotions. The success in well-recognized classes suggests that the models can effectively capture certain emotional cues. Further analysis and fine-tuning may enhance overall performance, especially for frequently confused classes.

Impact of Architectural Variations

The depth of the model appears to contribute to its ability to capture features. **Variant 1**, with an additional convolutional layer, may have had an advantage in recognising more complex and abstract features, potentially improving its ability to recognise intricate patterns in facial expressions. **Variant 2** larger kernel sizes (5x5) captured broader patterns in facial expressions. This may have caused the model to struggle with recognizing finer details but excel in identifying overall emotional cues.

Conclusions and Forward Look

Best Performing Model

Variant 1 appears to be the most promising based on the observations. It benefits from an increased depth, allowing it to capture more complex and abstract features. Balances between capturing finer details and broader patterns, potentially providing a more comprehensive understanding of facial expressions.

Suggestions for Future Refinements:

- Continue experimenting with different architectures and hyperparameters to find the optimal balance between depth, kernel sizes, and model capacity.
- Implement data augmentation techniques to artificially increase the size of the training dataset. This can help improve generalization and reduce overfitting.
- Incorporate dropout layers, batch normalization, or other regularization techniques to mitigate overfitting, especially with deeper models.

Confusion Matrix of the Final Model

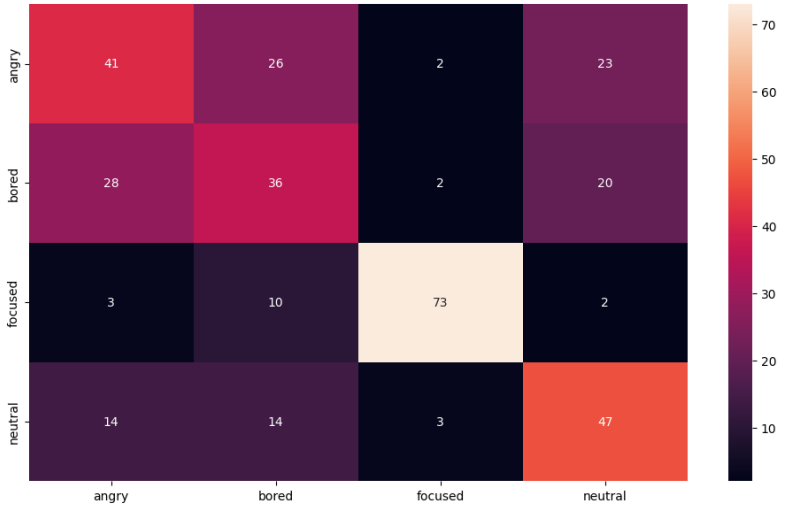


Figure 15: Confusion Matrix of final Model

K-fold cross-validation

Fold	Macro			Micro			Accuracy
	P	R	F	P	R	F	
1	0.89	0.83	0.82	0.8	0.8	0.8	63.76
2	0.625	0.625	0.53	0.6	0.6	0.6	62.01
3	1	1	1	1	1	1	62.88
4	0.33	0.5	0.375	0.4	0.4	0.4	60.7
5	0.375	0.5	0.42	0.6	0.6	0.6	62.01
6	0.33	0.33	0.33	0.4	0.4	0.4	65.07
7	0.5	0.5	0.44	0.5	0.5	0.5	63.16
8	1	1	1	1	1	1	62.72
9	0.83	0.83	0.78	0.75	0.75	0.75	60.09
10	0.33	0.5	0.39	0.5	0.5	0.5	55.7
Average	0.62	0.66	0.61	0.66	0.66	0.62	61.81

Table 2: 10-fold cross-validation Part 2 Saved Model

Fold	Macro			Micro			Accuracy
	P	R	F	P	R	F	
1	1	1	1	1	1	1	77.73
2	0.875	0.875	0.833	0.8	0.8	0.8	68.12
3	1	1	1	1	1	1	71.62

4	0.389	0.667	0.489	0.6	0.6	0.6	69
5	1	1	1	1	1	1	72.05
6	1	1	1	1	1	1	75.11
7	0.5	0.67	0.56	0.75	0.75	0.75	72.68
8	0.56	0.67	0.6	0.75	0.75	0.75	71.05
9	1	1	1	1	1	1	72.81
10	0.5	0.5	0.5	0.75	0.75	0.75	71.49
Average	0.781	0.834	0.797	0.865	0.865	0.865	72.4

Table 3: 10-fold cross-validation Part 3 Saved Model

Model in Part 2

Performance Variability: There's a notable fluctuation in performance metrics (Precision, Recall, F1-Score)across different folds. For example, in some folds, the Macro Precision reaches as high as 1.0, whereas in others, it drops to 0.33.

Overall Performance: The average performance metrics are moderate, with Macro Precision, Recall, and F1-score averaging around 0.62, 0.66, and 0.61, respectively. The average accuracy is 61.81%.

Inconsistency: The variability in performance across folds suggests inconsistency. This could be due to the model's overfitting to certain data subsets or not generalizing well across the dataset.

Model in Part 3

Higher and More Stable Performance: Compared to Model 2, Model 3 shows higher average performance metrics (Precision, Recall, F1-Score) with less variability across the folds.

Consistency in Performance: Most folds exhibit high performance metrics, with several folds achieving perfect scores (1.0) in Macro and Micro Precision, Recall, and F1-score.

Overall Performance: The average performance is better than Model 2, with average Precision, Recall, and F1-score all above 0.78 and an average accuracy of 72.4%.

General Observations

Model 3's Superior Performance: Model 3 outperforms Model 2 in terms of average performance metrics and consistency across folds. This suggests better generalization capabilities of Model 3 over Model 2.

Model 2's Inconsistency: The significant variability in Model 2's performance metrics across different folds indicates potential issues with overfitting or underfitting in certain data subsets.

Consistency vs. Average Performance: Model 3 not only has higher average performance metrics but also demonstrates more consistent results across different data subsets, indicating a more reliable model for varied data.

Train/Test Evaluation Compariso to 10-fold cross-validation Part 2

Analysis of Discrepancies

1. Data Diversity and Model Generalization

K-Fold Variation: The varied performance in k-fold cross-validation suggests that Model 2's performance is highly dependent on the specific data it's trained on. Some data segments may be easier or harder for the model to predict.

Fixed Split Limitation: The lower performance in the original train/test evaluation indicates that the test set in this split was particularly challenging for the model, or that the model had overfit to the training data.

2. Overfitting and Underfitting

Inconsistent K-Fold Results: The wide range of performance metrics in the k-fold cross-validation could be a sign of overfitting (model performing exceptionally well on some folds) or underfitting (performing poorly on others).

Stable but Lower Performance in Original Evaluation: The consistent but lower performance in the original evaluation suggests that when tested on a fixed set of data, the model may not generalize well, indicating potential issues with its ability to handle diverse data scenarios.

Bias Analysis.

Introduction

Age
Categories: Typically, age can be divided into several categories such as 'young', 'middle-aged', and 'senior'. These categories help in understanding how the model performs across different age groups.
Analysis Approach: The dataset was segmented based on these age categories. The performance of the AI model was then evaluated separately for each age group. This approach helped identify if the model's effectiveness varied significantly across age groups, which could indicate age-related bias.

Gender:
Categories: Gender categories typically include 'male', 'female', and 'other'. It's crucial to consider diverse gender representations to ensure the model's fairness and inclusivity.
Analysis Approach: Similar to age, the dataset was divided based on gender categories. The model's performance metrics (accuracy, precision, recall, and F1-score) were calculated for each gender group. This assessment helped determine if the model favored one gender over others or if it performed equally well across all genders.

Bias Detection Results:

Attribute	Group	Accuracy	Precision	Recall	F1-Score
Age	Young	47.76%	50.34%	47.01%	47.76%
	Middle-aged	56.37%	57.91%	57.61%	57.4%
	Senior	59.02%	58.65%	58.88%	58.74%
	Average	54.38%	55.63%	54.50%	54.63%
Gender	Male	55.94%	56.32%	55.62%	55.86%
	Female	44.72%	48.26%	47.96%	46.79%
	Other/Non-binary	53.03%	54.52%	51.72%	52.36%
	Average	51.23%	53.03%	51.77%	51.67%
Overall System Average		52.81%	54.33%	53.13%	53.15%

Table 4: Bias Analysis Table for Part 2 Model

Analysis for Age Attribute
Middle-Aged Group: Shows moderate performance across all metrics with accuracy at 56.37%. This group has balanced Precision and Recall, indicating a relatively equal rate of false positives and false negatives.
Senior Group: Exhibits the highest performance among the age groups, with accuracy at 59.02%. The Precision, Recall, and F1-Score are all slightly above the middle-aged group, suggesting better model reliability for this demographic.

Young Group: Has the lowest performance, with accuracy significantly lower at 47.76%.
Both Precision and Recall are lower, indicating the model struggles more with correctly identifying true cases and avoiding false positives in this group.

Analysis for Gender Attribute
Female Group: Shows the lowest performance among gender groups, with an accuracy of 44.72%.
Precision is higher than Recall, suggesting the model is more conservative in predicting positive cases for this group, leading to more false negatives.
Other/Non-Binary Group: Displays moderate performance with an accuracy of 53.03%.
Precision is slightly higher than Recall, indicating a slight tendency towards false negatives over false positives.
Male Group: Exhibits the highest performance among gender groups with an accuracy of 55.94%. The Precision and Recall are relatively balanced, suggesting a more equitable performance in terms of false positives and false negatives compared to other gender groups.

Overall Observations
Age Bias: The model performs best for the senior group and worst for the young group. This indicates a potential bias where the model is less effective for younger individuals.
Gender Bias: The model shows the lowest accuracy and F1-Score for the female group and the highest for the male group. This disparity suggests a gender bias where the model is more effective for males than females or other/non-binary individuals.

Bias Mitigation Steps:
Based on the first analysis, the dataset were increased for groups that were underrepresented. That had a significant impact on the bias analysis, the model was biases towards the group. Thereafter we tried several attempts to reduced/ increase the dataset to get to a common ground that at least represents all the groups appropriately.
Comparative Performance Analysis:

Attribute	Group	Accuracy	Precision	Recall	F1-Score
Age	Young	54.88%	55.11%	54.22%	54.52%
	Middle-aged	59.39%	60.24%	59.88%	59.83%
	Senior	52.31%	52.11%	48.88%	50.12%
	Average	55.53%	55.82%	54.33%	54.82%
Gender	Male	56.39%	56.03%	55.66%	55.63%
	Female	60.84%	62.73%	61.07%	61.13%
	Other/Non-binary	51.47%	54.23%	52.02%	52.96%
	Average	56.23%	57.66%	56.25%	56.57%
Overall System Average		55.88%	56.74%	55.29%	55.70%

Table 5: Bias Analysis Table for Part 3 Model

Age Attribute

After Mitigation Impact: Improved across all metrics, indicating a more balanced and effective model for this age group.

After Mitigation Impact: Decrease in all metrics, suggesting that the model's effectiveness for this group has reduced, possibly due to a decrease in representation in the dataset.

Young Group:

After Mitigation Impact: Significant improvement across all metrics, indicating successful mitigation efforts for this group.

Gender Attribute

Female Group:

After Mitigation Impact: Significant improvement, suggesting that the model now performs much better for this group.

Other/Non-Binary Group:

After Mitigation Impact: A slight decrease in accuracy and precision, but an improvement in recall and F1-Score. This indicates a mixed impact, possibly due to changes in the group's representation.

Male Group:

After Mitigation Impact: Slight improvement in accuracy, a minor decrease in precision and recall. The impact seems to be marginal for this group.

The bias mitigation efforts, primarily through adjusting the representation of certain groups in the dataset, have had a mixed impact.

Most Improved: The young and female groups have shown considerable improvements in performance metrics, indicating successful mitigation for these previously underperforming groups.

Trade-Offs: The senior and other/non-binary groups experienced some decrease in certain metrics, suggesting a trade-off effect. This might be due to a reduction in their representation or an indirect effect of balancing other groups.

REFERENCES

[1] D. V. Sang, N. Van Dat and D. P. Thuan, "Facial expression recognition using deep convolutional neural networks," 2017 9th International Conference on Knowledge and Systems Engineering (KSE), Hue, Vietnam, 2017, pp. 130-135, doi: 10.1109/KSE.2017.8119447.

[2] Face expression recognition dataset. https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset, 2020. Accessed: October 2022. CC0: Public Domain.

[3]Dumitru, Ian Goodfellow, Will Cukierski, Yoshua Bengio. (2013). Challenges in Representation Learning: Facial Expression Recognition Challenge. Kaggle. https://kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge

[4]https://www.freepik.com/search

[5]https://help.roboflow.com/get-started/the-labeling-interface

[6]Roboflow Annotate: Label Faster Than Ever

[7]Introducing Roboflow Annotate

[8]https://www.pinterest.ca/

[9] J. Brownlee. Understand the impact of learning rate on neural network performance, Sep 2020.

[10] Pavansanagapati. What is dropout regularization?, Jul 2019.