MIE 1517 Project Report

Facial Expression Recognition

Group 5

Tongfei Li, Constance Lin, Jack Zhang, Welinlin Fu

December 8, 2023

Executive Summary

Facial expressions play a crucial role in conveying human emotions. This study uses approximately 30,000 images, divided into training, validation, and testing sets with an 8:1:1 ratio, for training a Convolutional Neural Network (CNN) model to classify emotions based on facial expressions. The original images are resized to 48x48 pixels for preprocessing before the model training.

The model is initially trained on mini-batches to identify optimal configurations for subsequent stages. Upon training the final model on the complete dataset, it achieved a training accuracy of 71.8%, a validation accuracy of 69.21%, and a test accuracy of 69.1%. These results indicate the model has a relatively good performance and ability to generalize well to unseen data.

Furthermore, when applied to new images captured in real-world situations, the model demonstrates satisfactory performance with an accuracy of 75%. This underscores the model's effectiveness in handling facial expression-based emotion classification in controlled datasets and practical scenarios.

Introduction

Facial expression serves as a pivotal element in social communication, acting as a "window" to convey human emotions and thoughts. In the field of computer science, it is worthwhile to explore methods for segmenting facial expressions with respect to the diversity of facial appearances. The project about real-time facial expression capture (Xu, H. & Dai, L, 2021) proves that machines can detect the facial expressions successfully. If machines can gain a better understanding of human motions, the applications of human-machine iterations could be extended to more areas. For example, they can be employed to monitor patients' conditions during medical treatment and enhance the service quality of virtual communication by adjusting the tone or response based on users' moods.

Motivated by these considerations, we implemented an emotion classification model to identify and categorize emotional expressions in images of people automatically. The model is designed to use 30,000 photos of human faces to predict five primary emotion types: anger, fear, happiness, sadness, and surprise. By training with approximately 27,000 images using convolutional neural networks, the model achieved an accuracy of 70% on a testing dataset comprising 3,000 images. Examples from the testing dataset are provided in the following five images. The outcomes indicate correct predictions for Samples 3, 4, and 5, but Samples 1 and 2 exhibit incorrect predictions.

```
Sample 1 - Predicted: fear, True: angry
Sample 2 - Predicted: angry, True: fear
Sample 3 - Predicted: happy, True: happy
Sample 4 - Predicted: sad, True: sad
Sample 5 - Predicted: surprise, True: surprise
```

Figure 1 - Outcomes of Testing Examples

While the testing accuracy is not significantly high, the model demonstrates success in classifying facial expressions across a diverse group of individuals. Therefore, it suggests that the model can effectively segment these five emotions across different people.

Data Overview

We used the Face Expression Recognition Dataset on Kaggle (J. Oheix, 2018) as the dataset for our model. All the images in the dataset are resized to 48x48 before training. Figure 2 is a set of 25 preprocessed images. Each row contains 5 images coming from the same emotion class, which is happy, surprise, angry, sad, and fear respectively. The examples are collected from different people, including female, male, adults, and children, the dataset has a diverse representation of individuals' facial expressions. Therefore, the dataset contains plenty of features in faces from each class for our CNN model to learn facial expressions.

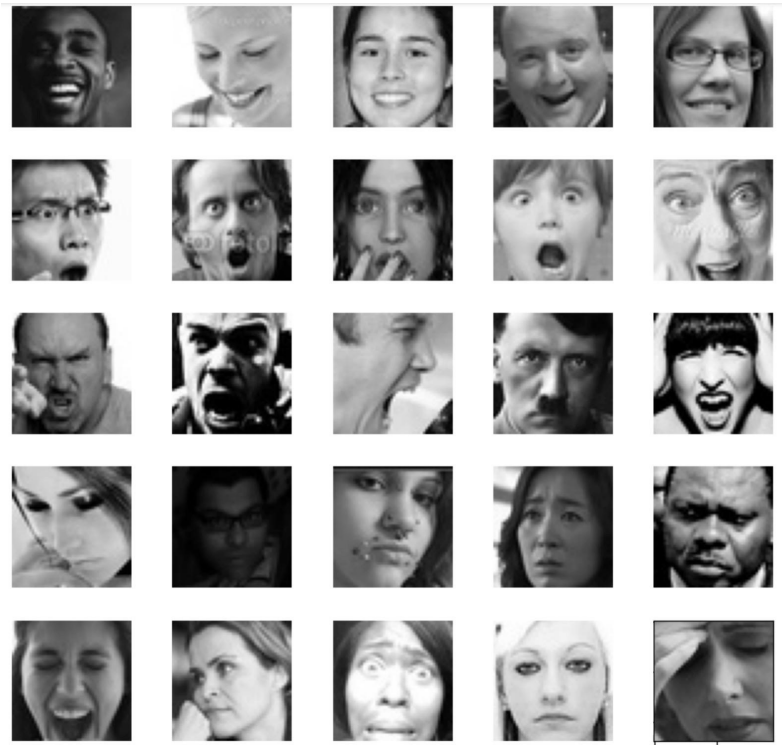

Figure 2 – Example of Training Dataset

The full dataset contains 29142 images of people's face expressions distributed across five distinct classes, which are happy, surprise, angry, sad and fear. The dataset is split into three subsets: training set, validation set and testing sets with an 8:1:1 ratio shown on Figure 3. The training set contains 23,403 images, while the validation and testing sets consist of 2,868 and 2,871 images, respectively.

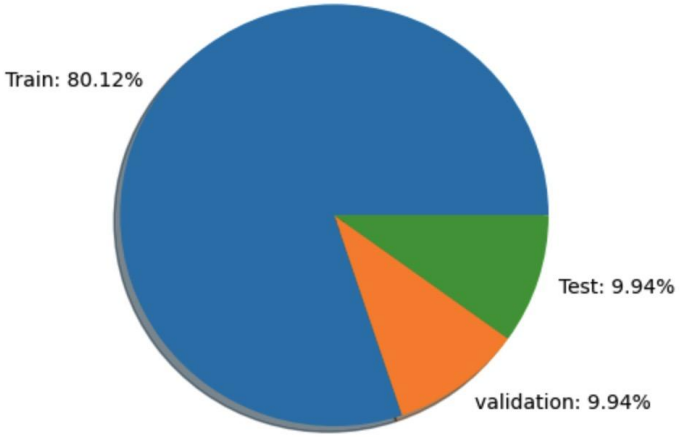

Figure 3 – Distribution of Dataset

Figures 4, 5 and 6 are the pie charts illustrate the components of the five classes on the training, validation, and testing datasets, respectively. All these three sets have a similar distribution of the five expression classes. Especially, 'happy' class is the primary component of all the sets, contributing to approximately 31%. The other four classes are much more balanced, with 20% for sad, 18% for fear, followed by angry and surprise, which are 17% and 14%. So, the model evaluation would be fair across three datasets, and the result would be consistent in the cross validation.

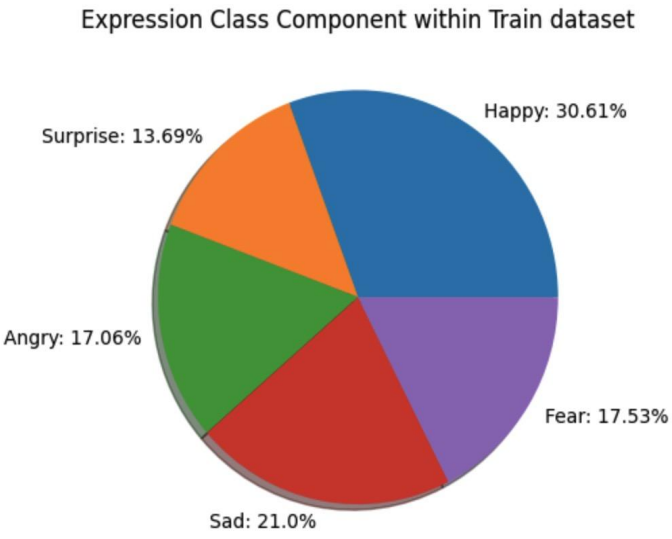

Figure 4 – Class Components of Training Dataset

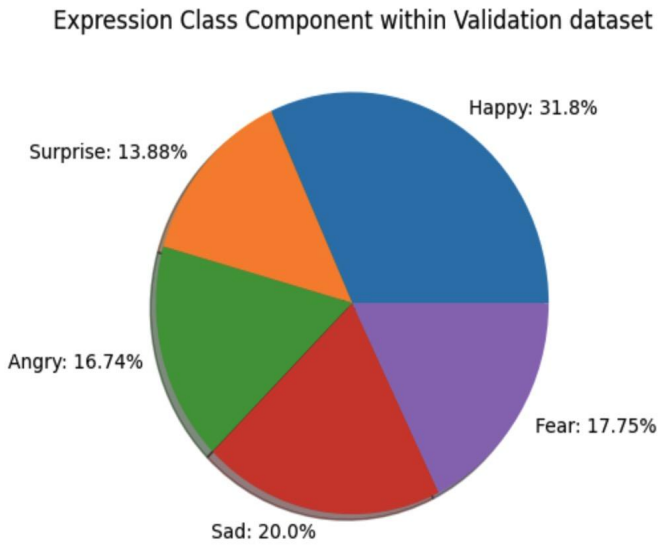

Figure 5 – Class Components of Validation Dataset

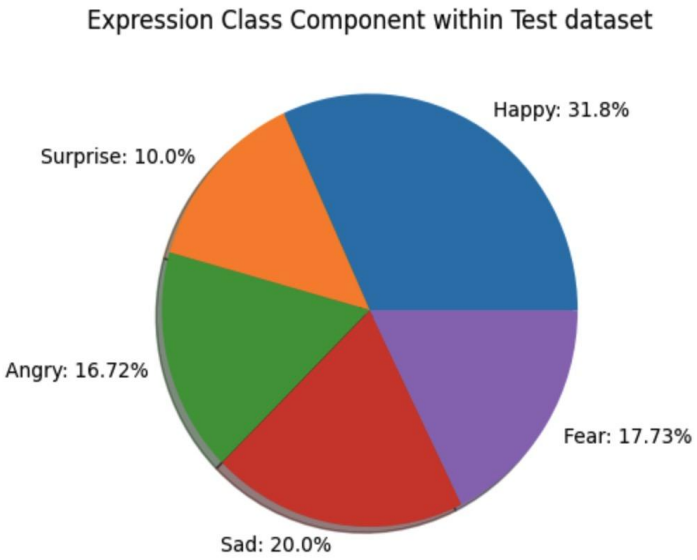

Figure 6 – Class Components of Testing Dataset

Data Loading and Processing

Our facial expression recognition model (J. Oheix, 2018) takes facial expression images as feature and their corresponding emotion categories as label. Initially, our dataset contains seven emotion categories: fear, sad, happy, surprise, neutral, disgust, and angry. However, we deliberately reduced the number of emotion classes to five by removing neutral and disgust to create a more balanced dataset and emphasize more precise distinctions between classes.

To facilitate the loading and preprocessing of our image data, we employed the ImageDataGenerator module, which helps load the dataset efficiently and incorporates data augmentation techniques. Additionally, we resized all images to a consistent 48x48 pixel resolution as part of our standardization process. This ensures uniformity in the input data and would help the model to learn and generalize across different facial expressions.

Finally, we split our dataset by the ratio 8:1:1 as the training, validation, and testing set. This systematic split enables us to evaluate the model's performance on unseen data and make necessary adjustments for better generalization.

Model Architectures

The model architecture is shown in the Figure 7 (P. Gavrikov, 2022), which starts with a Convolutional Layer (Conv2D) using 32 filters and a 3x3 kernel size. The activation function, Leaky Rectified Linear Unit (LeakyReLU), helps capture the detailed information. Additionally, Batch Normalization could make the training more efficient by standardizing and stabilizing the activations.

After the convolutional layer, a MaxPooling layer with a 2x2 pool size reduces spatial dimensions and captures essential features. A Dropout layer with a 0.25 dropout rate is included to avoid overfitting during training. This process is repeated with two more Conv2D layers, each doubling the number of filters (64 and 128) to capture more complex features. Following the convolutional layers, a flattened layer transforms the 2D output into a 1D array, preparing for the fully connected layers.

The last Dense layer acts as the output layer, with five units and a softmax activation function, reflecting the reduced number of emotional classes. This layer creates a probability distribution over the classes, helping the model make predictions.

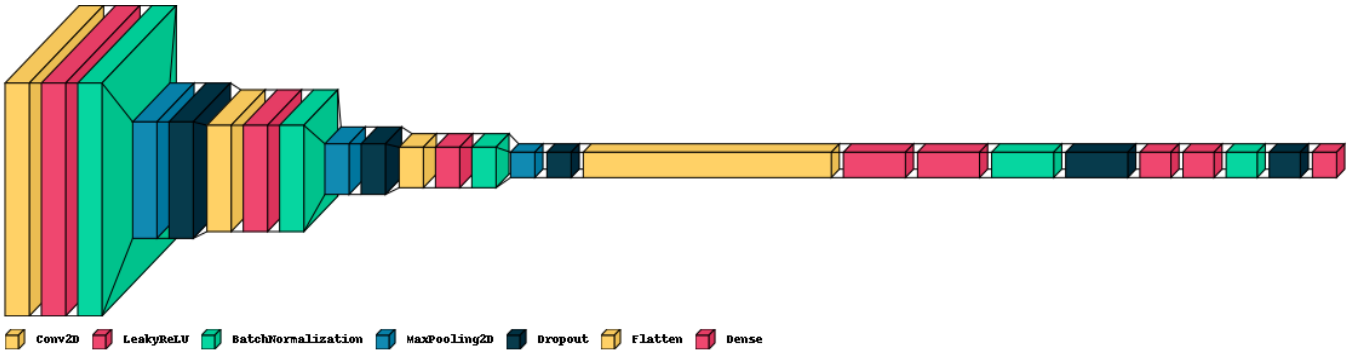

Figure 7 – Model Structure

Quantitative Results

In hyperparameter tuning, we tune the learning rate, the dropout rate of convolutional and dense layers, and the second fully connected layer size. By tuning the learning rate, we adjust the size of the model's steps during training to change its weights, control the convergence speed, and try to avoid unstable learning. Besides, we adapt the regularization strength by tuning the dropout rate to prevent the model from overfitting too quickly. The size of the dense layer is also tuned to adjust the model capacity to process information gained from the last convolutional layer. Consequently, our best CNN model has a learning rate of 0.0003, a dropout rate of 0.4 for convolutional layers, and 0.5 for dense layers.

In addition, accuracy is selected as the performance metric to evaluate our model performance because it is easy to understand and interpret. The best refined CNN model gets a train accuracy of 71.8%, which is a sign that the model successfully memorizes our train data. The validation accuracy is 69.21%, and the test accuracy is 69.1%, which means that our model captures underlying patterns across five expressions and generalizes on unseen data.

The Figures 8 and 9 below are the learning curves of the model, showing the training vs validation accuracy and loss of our best CNN model over 50 epochs.

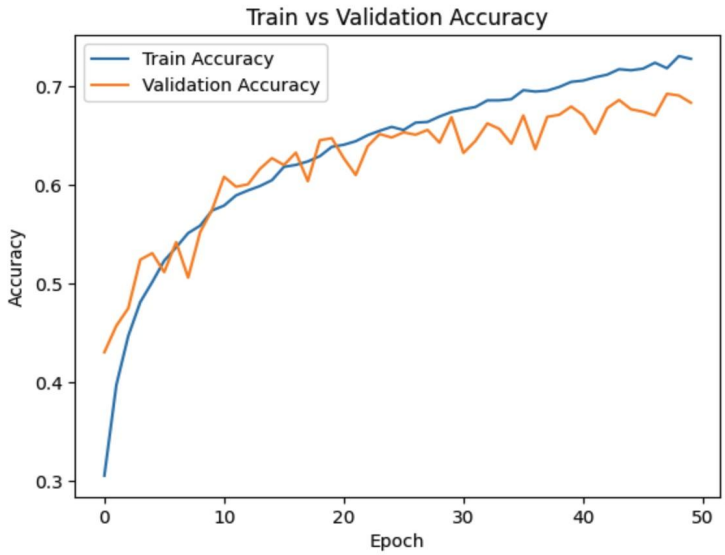

Figure 8 – Training vs Validation Accuracy

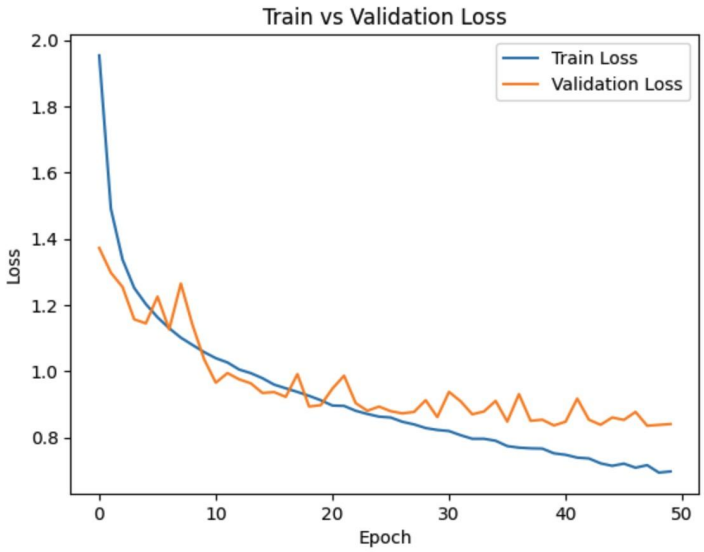

Figure 9 - Training vs Validation Loss

In the graph of training vs validation accuracy (Figures 8), the blue line represents the training accuracy, and the orange line represents the validation accuracy. Both accuracies increase over time as the model is learning more features from the input images. The training accuracy is lower than validation before epoch 5. After epoch 20, it is consistently higher than the validation accuracy. The training accuracy is stable, while validation accuracy fluctuates a lot. In addition, training, and validation accuracy level off in the last 20 epochs; the highest validation accuracy is around 69%, while training accuracy is around 72%. There's no sign of overfitting or underfitting; our model fits the data well.

In the graph of training vs validation loss (Figures 9), The blue line represents the training loss, and the orange line represents the validation loss. Both training and validation loss decrease over time as the model learns, which is what we expect to see. Initially, the training loss decreases rapidly. Then, it reaches its elbow point after epoch 30. The validation loss drops alongside the training loss, but there are more fluctuations.

Therefore, these two graphs show that our model is learning effectively from the data even though the validation losses are unstable.

Furthermore, class-wise performance is examined. It turns out that the model is best at recognizing happy faces, with 89% accuracy. But it struggles with fear faces, only getting 39% accuracy as shown on Figure 10.

Class	Class-wise Performance	Major Classification
Angry	60.42%	Sad
Fear	39.10%	Sad
Happy	88.83%	Sad
Sad	65.44%	Angry
Surprise	77.94%	Happy

Figure 10 - Class-wise Performance

Lastly, the confusion matrix of on testing dataset is also generated (Figure 11). The matrix's number at index (i, j) represents the number of samples in class i that are predicted as class j. The color scale indicates the frequency of predictions, with darker colors representing higher frequencies.

The image shows that a significant misclassification happens in class fear; 135 are misclassified as sad, and 88 are misclassified as fear. The model performs best in class happy. Almost all the samples are correctly classified. In terms of precision, happy scores the highest, with precision of 81%, whereas angry and sad scores the lowest, which is around 58%. In terms of recall, happy scores the highest as well, recall value is around 89%, fear has the lowest recall rate, the score is only 39%.

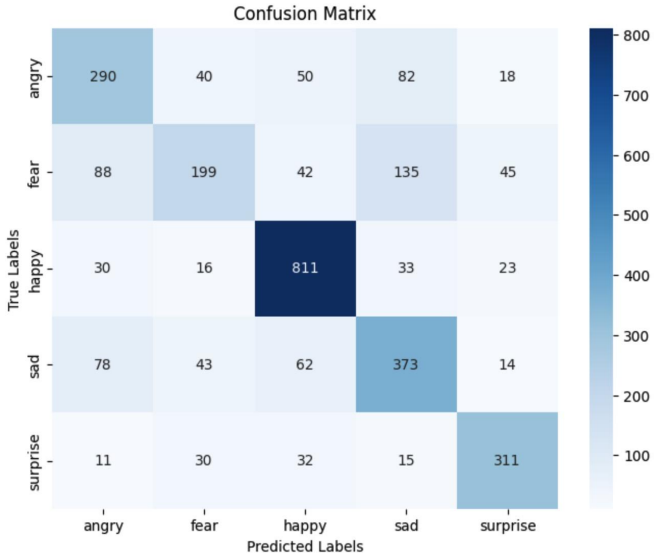

Figure 11 – Confusion Matrix of Testing Dataset

Overall, our model seems to be good at predicting positive emotions like happy and surprise, but struggles with negative emotions like angry, sad and fear. It suggests that the model may not have learned distinctive features among these expressions, or the dataset used for training might not have enough variance on these classes.

Qualitative Results

The interesting finding relies on the two parts: accuracy difference among five emotion categories, and accuracy difference between the mini-batch and the full dataset.

First, there is a significant difference in accuracy on different emotion types as shown on Figure 10. As we have seen previously, there is a discrepancy between the accuracy of the class happy and the class fear. This may be caused by the training sample size of these two classes. By checking the images, we also notice there are some similarities in the images among different classes. This might make the classification challenging and raise the problem of misclassification. This might also cause the classification result to deviate to the dominant class easily.

We also realize that there is difference on accuracy between the mini-batch and the full dataset. Our model is initially trained on the mini-batch to find the ideal model for the next stage efficiently. When conducting this process, we notice that the model trained on the mini-batch holds a different training and validation accuracy alongside the model trained on the complete set, as shown on Figure 12.

	Training Accuracy	Validation Accuracy
Best model trained on mini-batch	64.7%	51.9%
Best model trained on complete set	**71.8%**	**69.1%**

Figure 12 Training and Validation Accuracy on Mini-batch and Complete Set

Application on New Data

We collected 16 new images featuring various facial expressions, each corresponding to one of the targeted emotions: anger, fear, happiness, sadness, and surprise. The model successfully assigned emotion classes to these images with 75% accuracy, while four images were misclassified: sad from Individual 1, fear and sad from Individual 2, and angry from Individual 3. The following is the result:

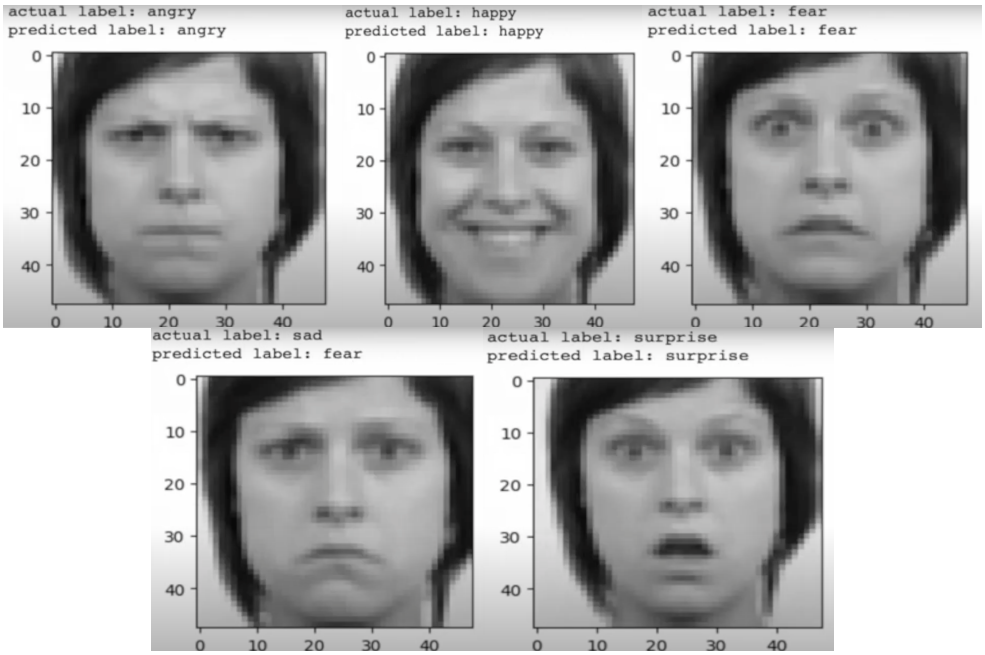

Figure 1- Individual 1

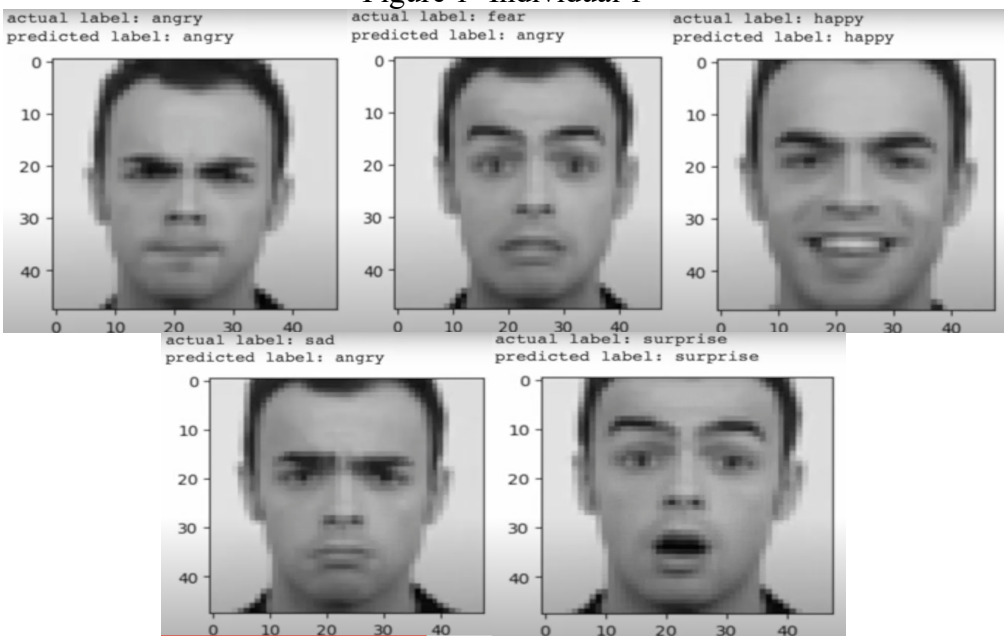

Figure 2 - Individual 2

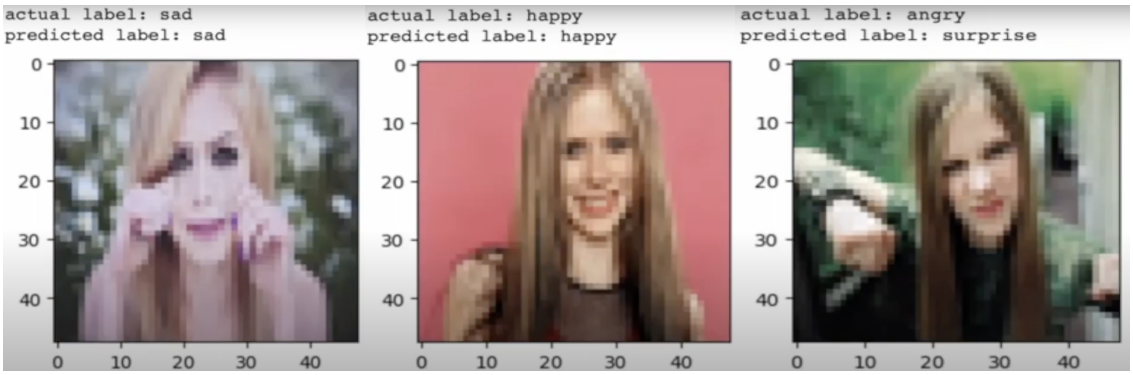

Figure 3 - Individual 3

Figure 4 - Individual 4

The above 16 images were obtained from a diverse group of four individuals, including women and men. This diversity highlights the model's proficiency in analyzing emotions across individuals. Furthermore, the model demonstrates the capability to discriminate different emotions expressed by the same person, as each individual displays at least three distinct facial expressions within this new dataset. Moreover, this new image set comprehensively covers all five targeted emotions, and each emotion class contains at least one accurately segmented example. Therefore, based on facial expressions, the model can predict the five targeted emotions from different people.

Discussion

From the result above, we notice that the final model achieves a training accuracy of 71.8%, a validation accuracy of 69.21%, and a test accuracy of 69.1%; from the curves, the model is well fit. However, we also notice that the accuracy among the five classes differs from that in the different categories. This might result from a biased prediction of specific classes due to an imbalanced dataset, where 30% of the images are happy faces.

Besides, the applicability of the model's application to real-world cases is restricted by its exclusive consideration of five classes. Given the model is trained only on Fear, Sad, Happy, Surprise, and Angry. The model would not be suitable for determining a random emotion in life beyond these five categories.

Additionally, training on the mini-batches is a good choice when determining which models are suitable to carry on to the next stage. When doing this project, we first trained the model to models on the mini-batches. Each of the batches contains 10% of the entire dataset. This method helps us find effective models faster. We could also observe which model is performing well in this task.

However, there is a difference in accuracy between batches and the full dataset under the same model, so the optimal model of the batch might not be the best for the full dataset. Therefore, we applied the best models of each batch to the full dataset and selected the one with the highest accuracy as the final optimal model. Even though it cannot guarantee this model is the best, it helps us compute a relatively reliable model in a shorter time.

Furthermore, the model's complexity needs to be improved as its accuracy is around 70%, which needs to be higher. We thought about increasing model complexity by adding more layers to improve accuracy, but we failed to finish it due to a significant increase in runtime.

References

Xu, H. & Dai, L (2021). High-Quality Real Time Facial Capture Based on Single Camera. Cornell University. Retrieved from https://arxiv.org/abs/2111.07556.

Oheix, J. (2018). Face expression recognition dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset.

Gavrikov, P. (2020). visualkeras. GitHub repository. Retrieved from https://github.com/paulgavrikov/visualkeras