1. **Problem Statement**
2. **Solutions Work Flow**
3. **Challenges Faced**
4. **Work Flow Diagram**
5. **Medium Blog (in details)**

**About the Dateset:**

Kaggle - https://www.kaggle.com/datasets/msambare/fer2013

The data 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.

The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)

In this dataset, we have two directory, train and test directory. Each directory contain 6 other directories, which indicate 6 different class name. (i.e 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)

**Challenges:**

The capacity to recognize non-verbal cues, such as facial expressions, is essential for understanding others' feelings and intentions

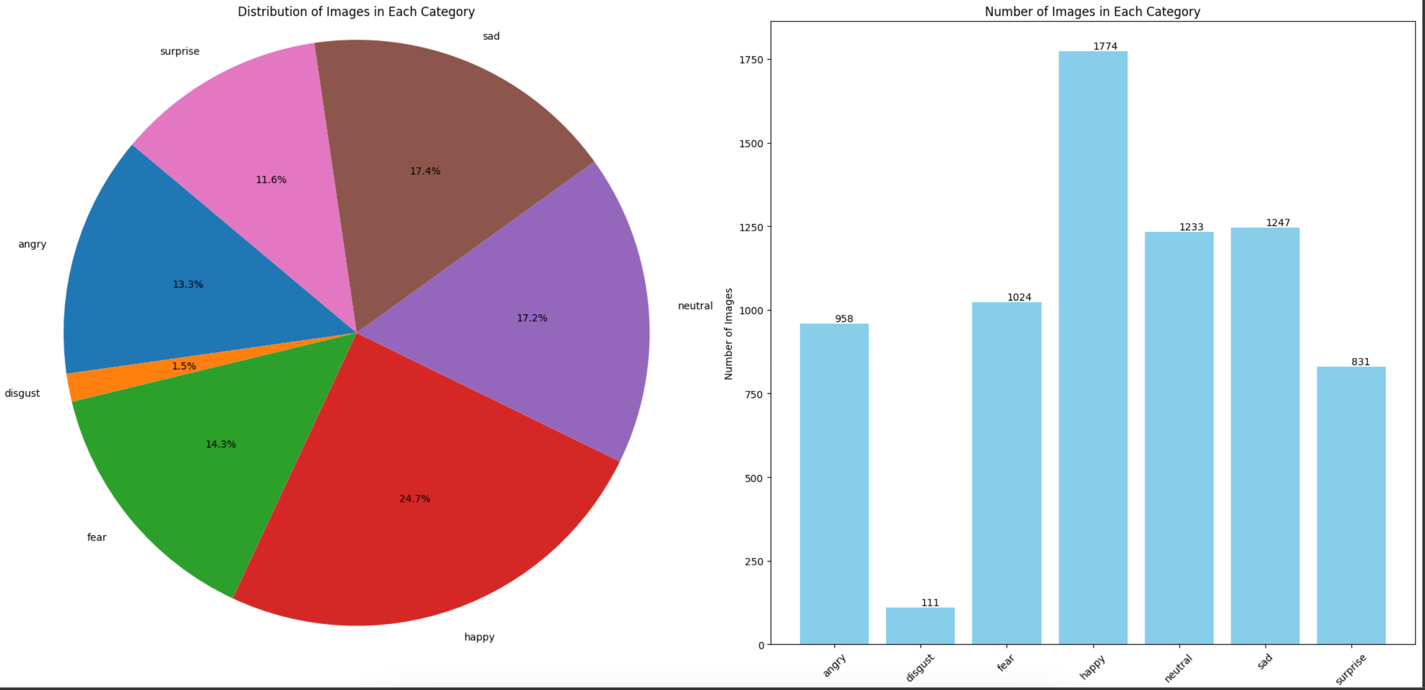
Factors that affect emotion recognition:

The influence of age, education, and fitness in the accuracy of recognizing compound emotions varied for each emotion. Some emotions such as Cruelty were better recognized by the 0-20 age group while the 60+ aged participants were more accurate in recognizing emotions like Devastation.

Usage:

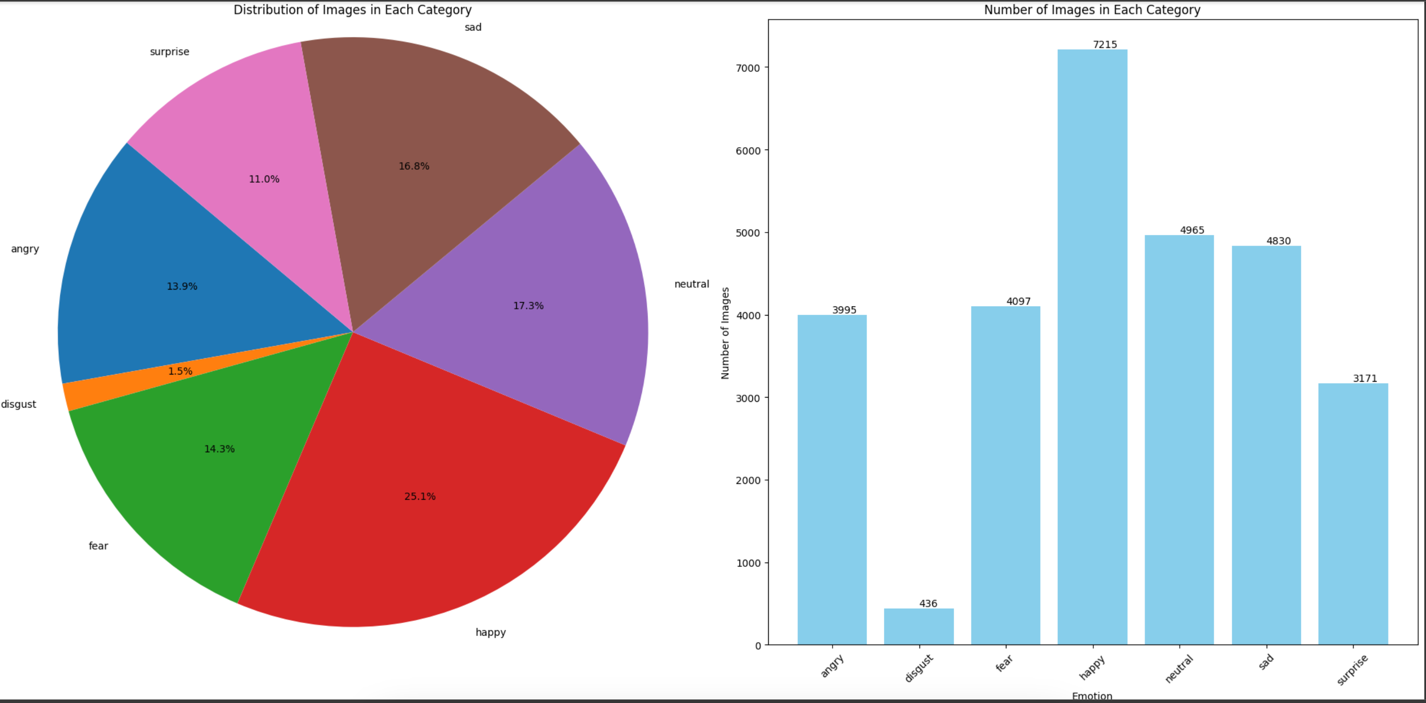
FRT can decrease the need for human interaction and thus increase efficiency during, for example, border checks in airports. FRT can even be employed in medicine, such as in identifying subtle facial traits to determine genetic disorders. A large number of tech companies are developing facial biometric systems.

**Image Distribution in train and test dataset:**



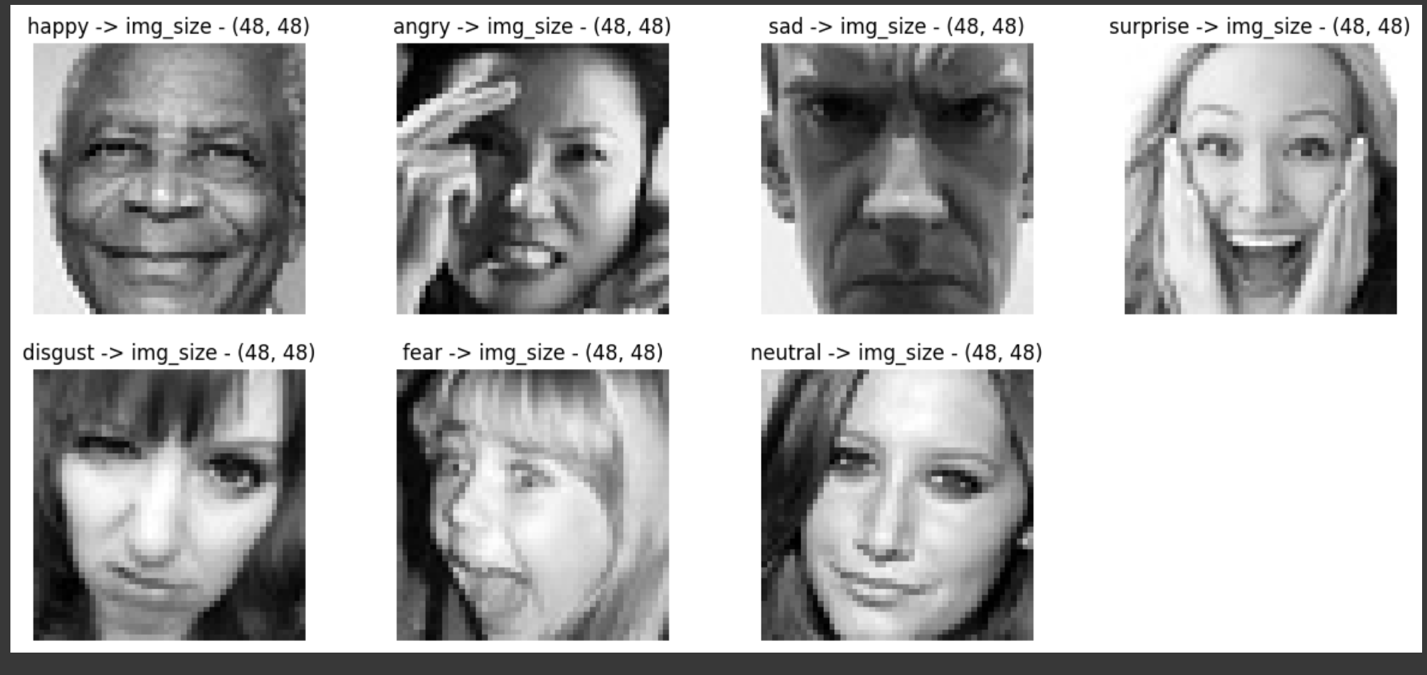
**Train dataset**

If the data distribution in each class is imbalanced, we will give class weights according to the number of images in each category.

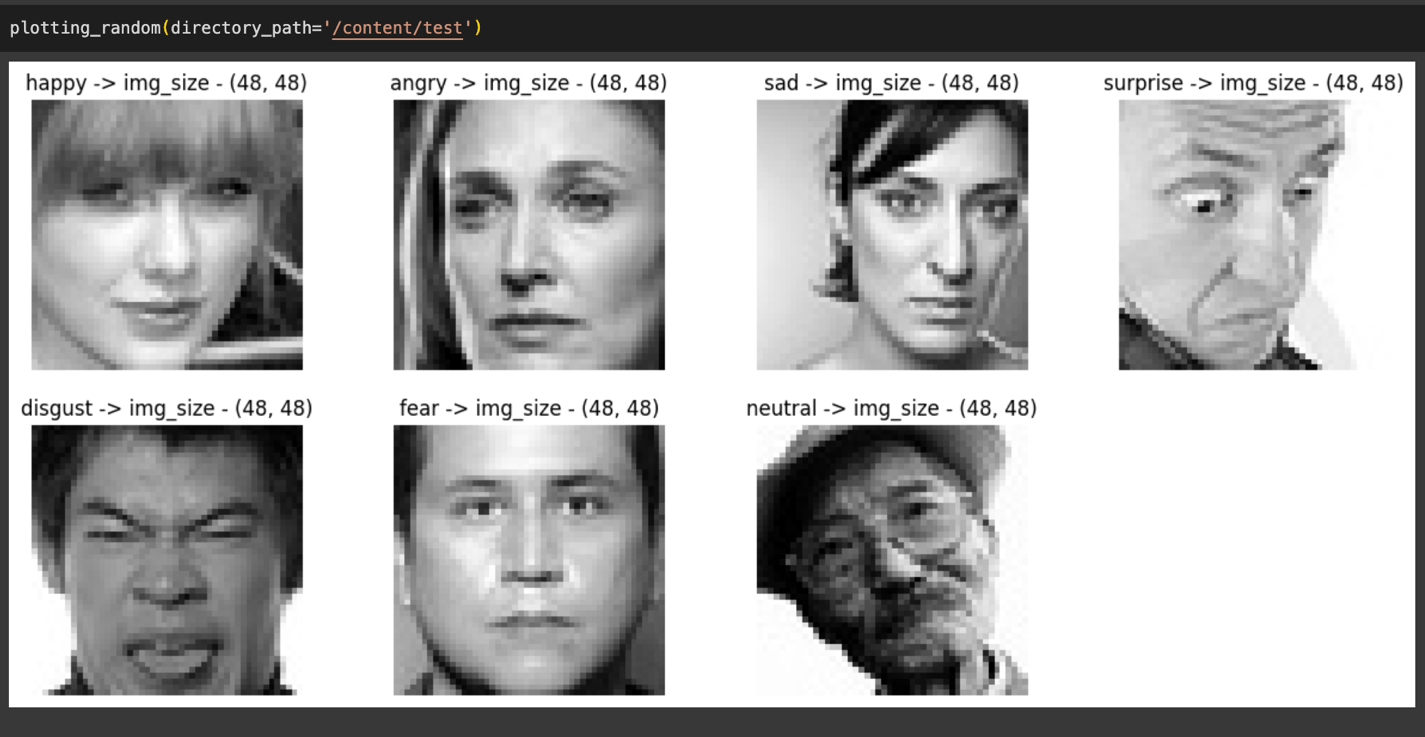


**Test dataset**

**Image Size details for each classes:**



**Train data**



**Test Data**

**Displaying Random images from each category**



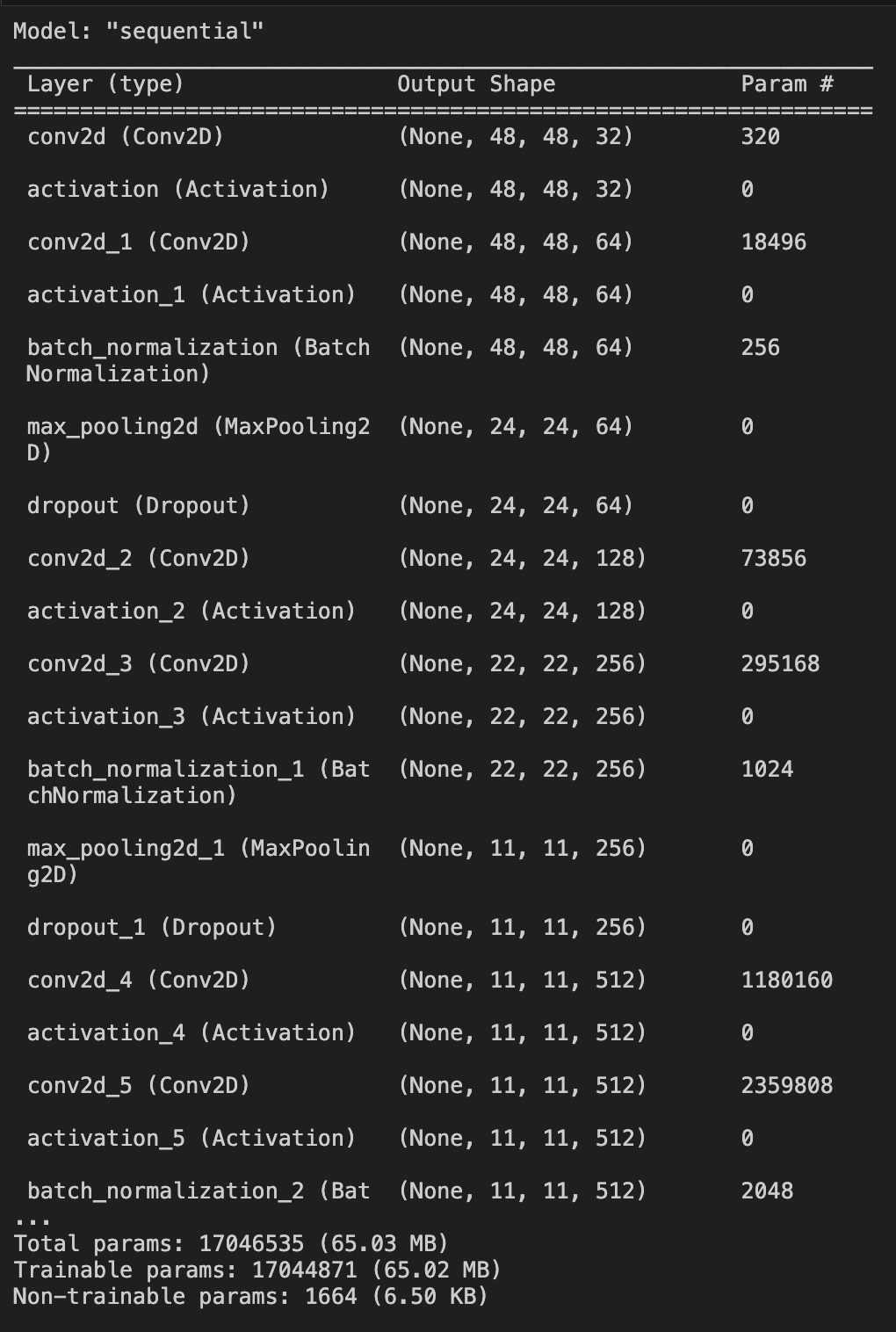
We are building 4 CNN models

1. CNN Model from Scratch
2. CNN model with Image Augmentation
3. CNN model using Transfer Learning (VGG16)
4. CNN model using Transfer Learning (ResNet50)

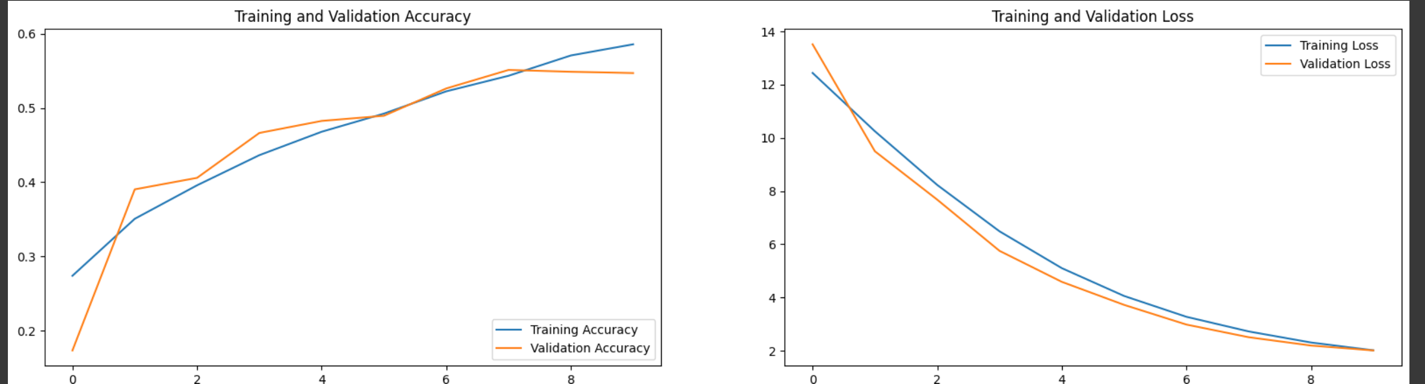
whichever is performing best we can pick that model and do prediction on test data using that model.

1. **CNN Model from Scratch**

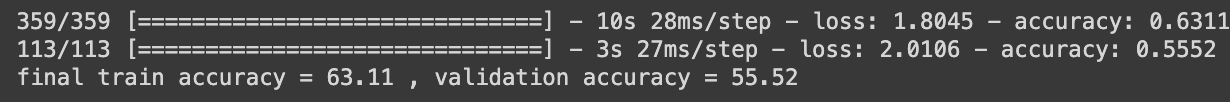
CNN Model from Scratch Model Summary:



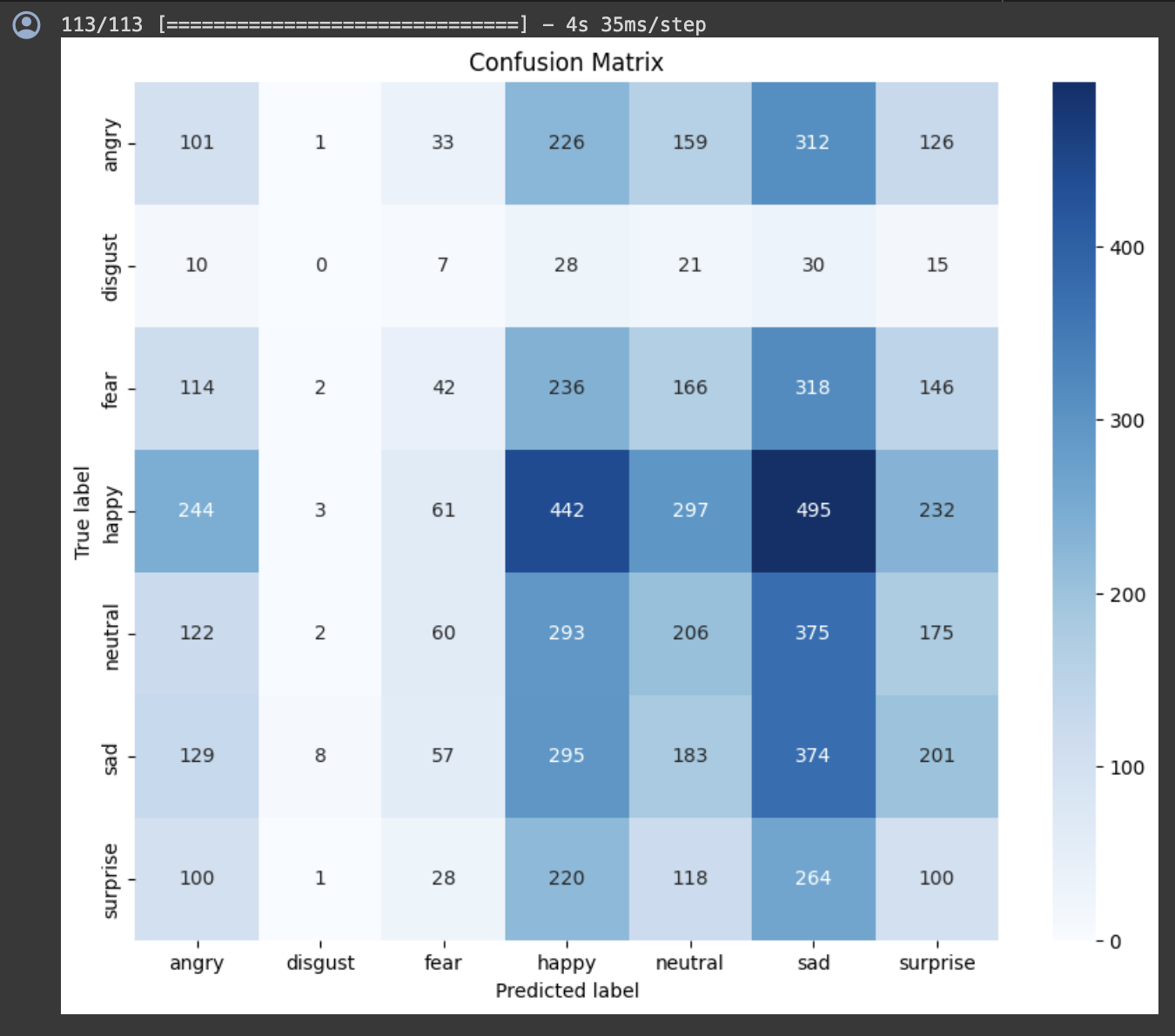
There are 17,046,535 total parameters in the CNN model, out of which 17,044,871 are trainable and 1664 are non-trainable parameters.



The above graph shows training and Validation accuracy and loss after 10 epochs.

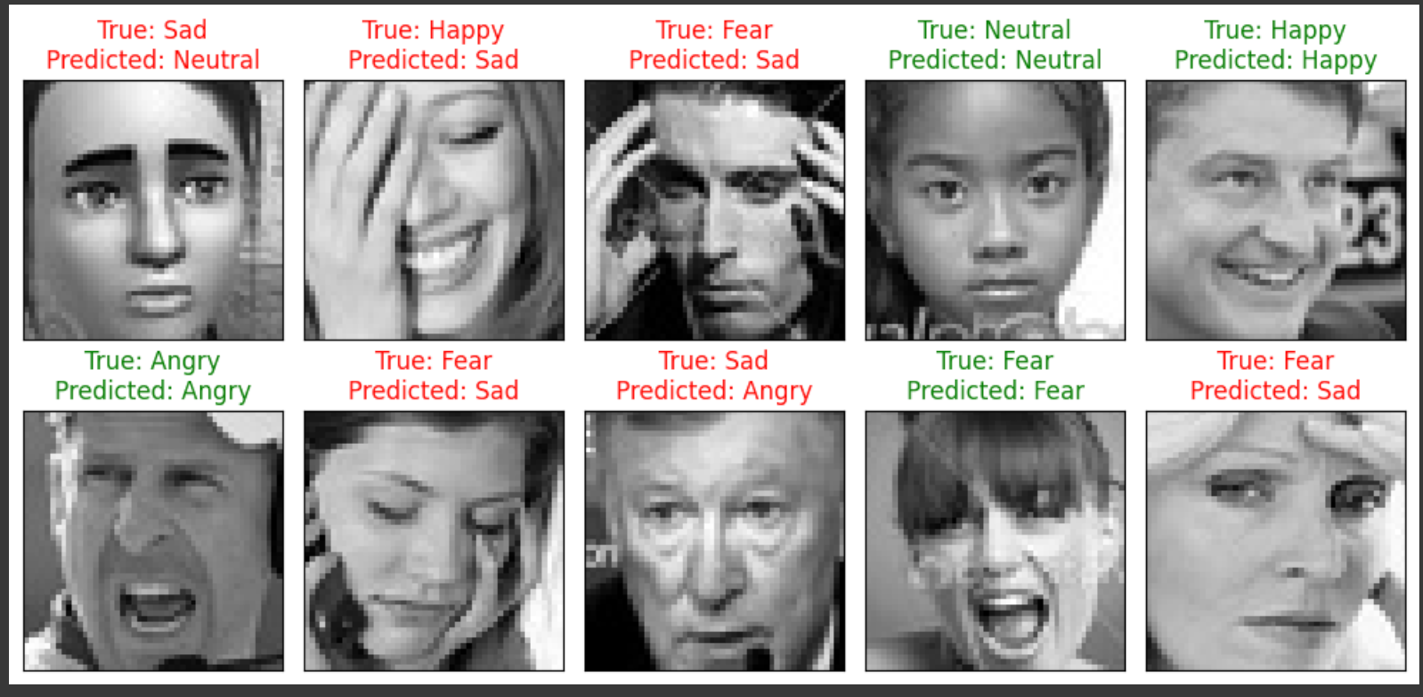


**Confusion Matrix:**



Since the disgust category contains least no of images, the confusion matrix shows that the model is unable to predict disgust class.

Besides the model is not that accurate in predicting actual classes for a given image.

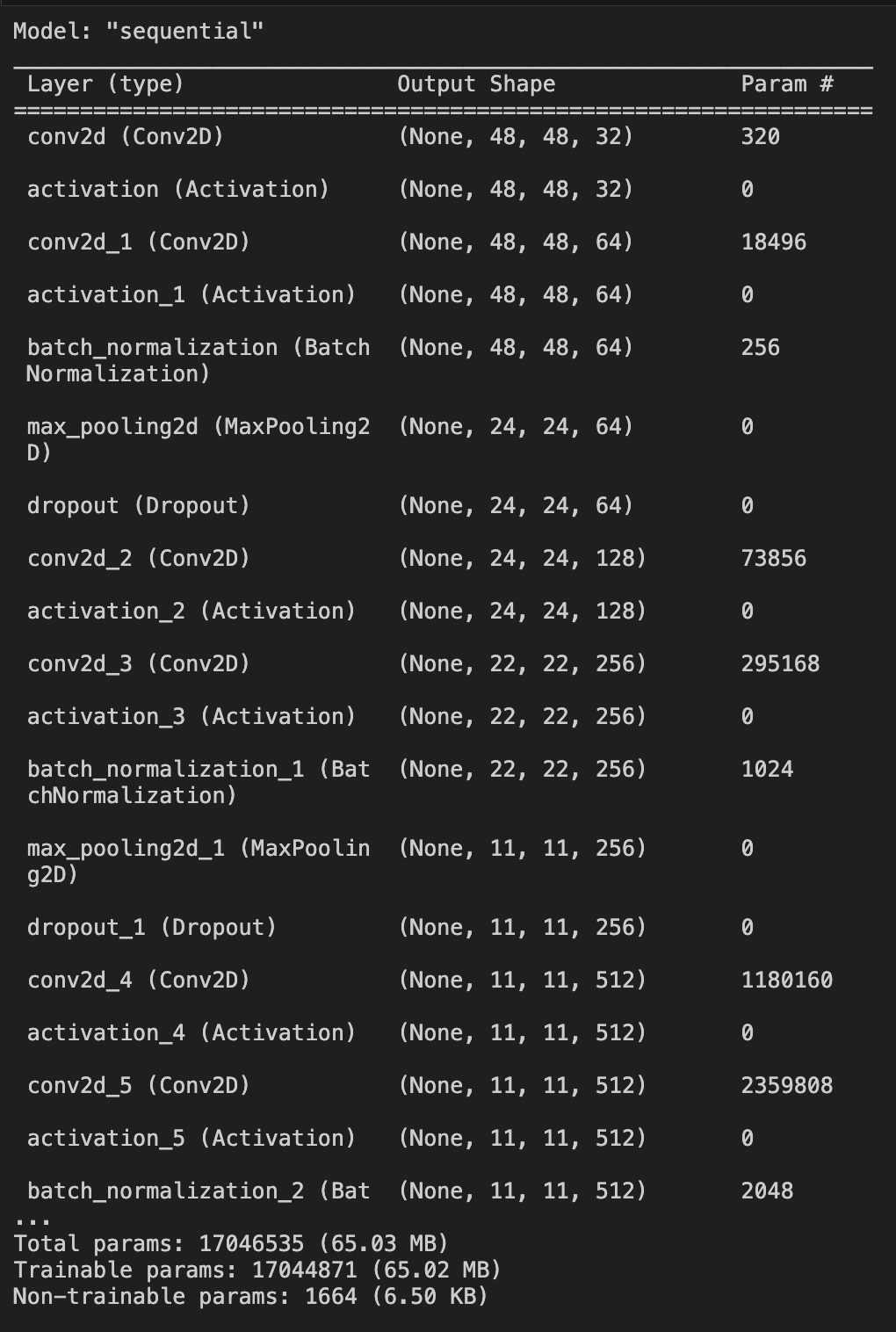


These are some of the images from the test data. Clearly, it is seen that the model is making errors in predicting the actual true value for a given image.

Therefore, we can conclude that the model is not suitable for us in predicting class of any images.

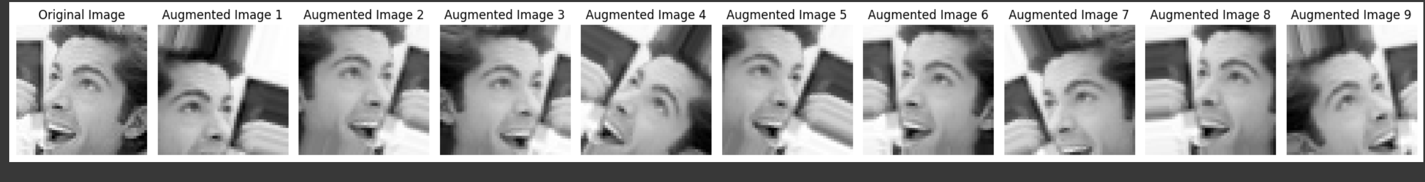
1. **CNN Model with Image Augmentation**

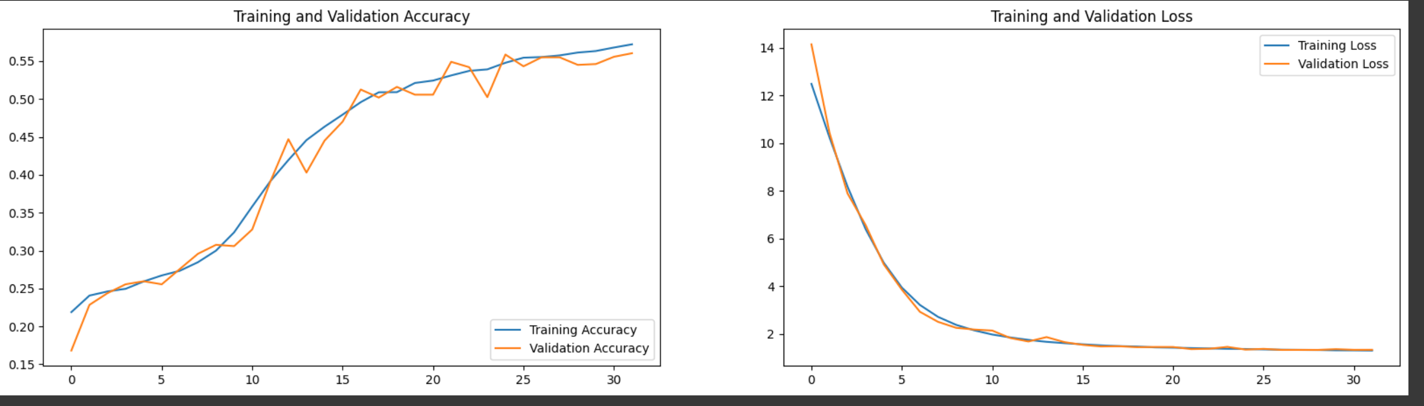
CNN Model with Image Augmentation Model Summary:



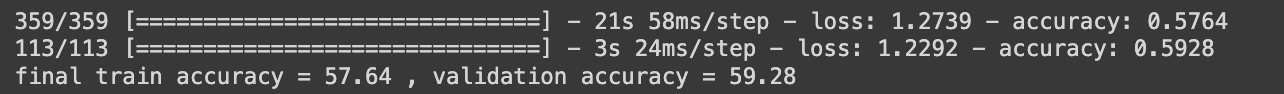
There are 17,046,535 total parameters in the CNN model, out of which 17,044,871 are trainable and 1664 are non-trainable parameters.

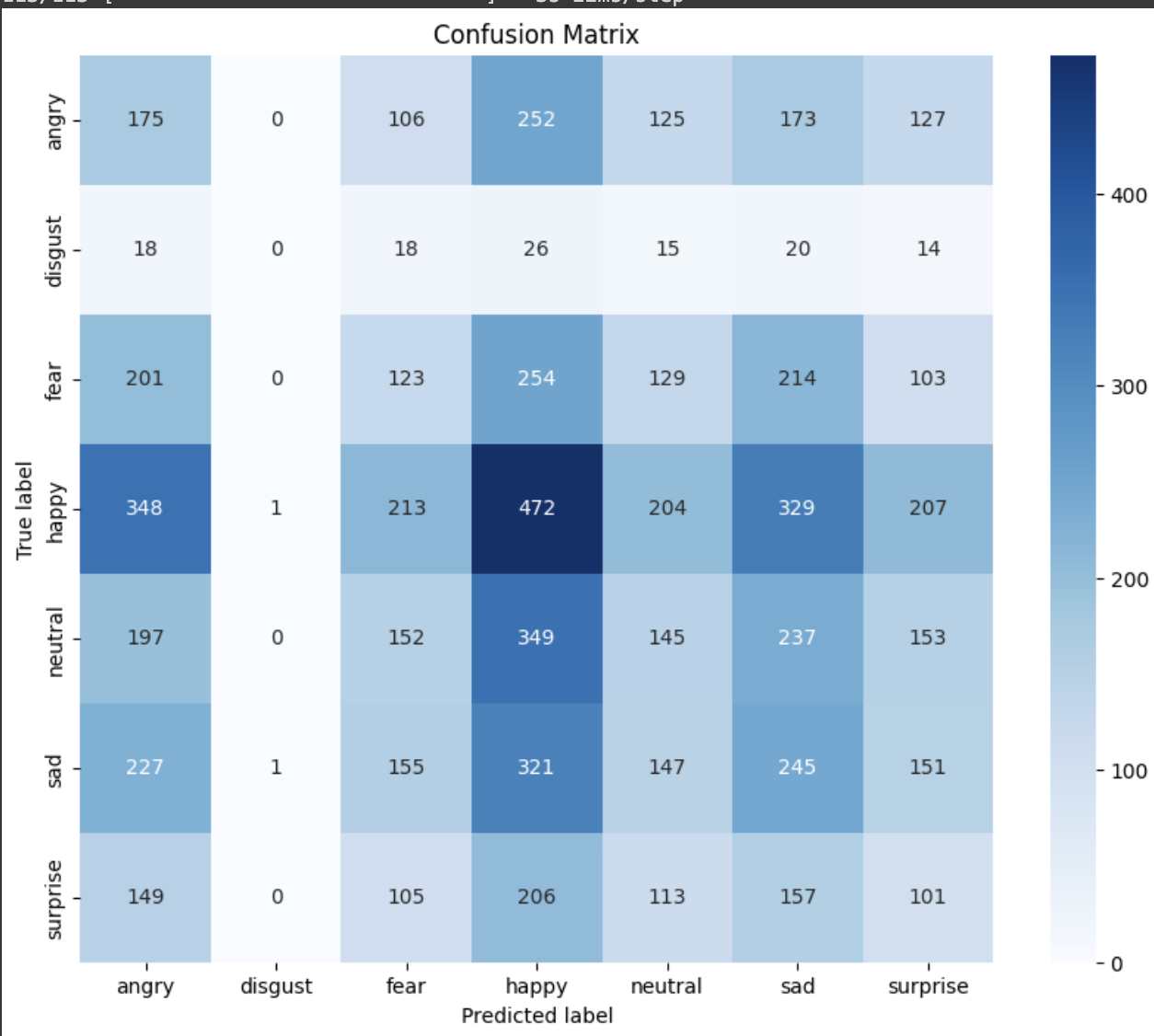
Images generated due to Image Augmentation method





As compare to previous model that is (CNN model from Scrtach) the performance of the model increase, as loss decreases.

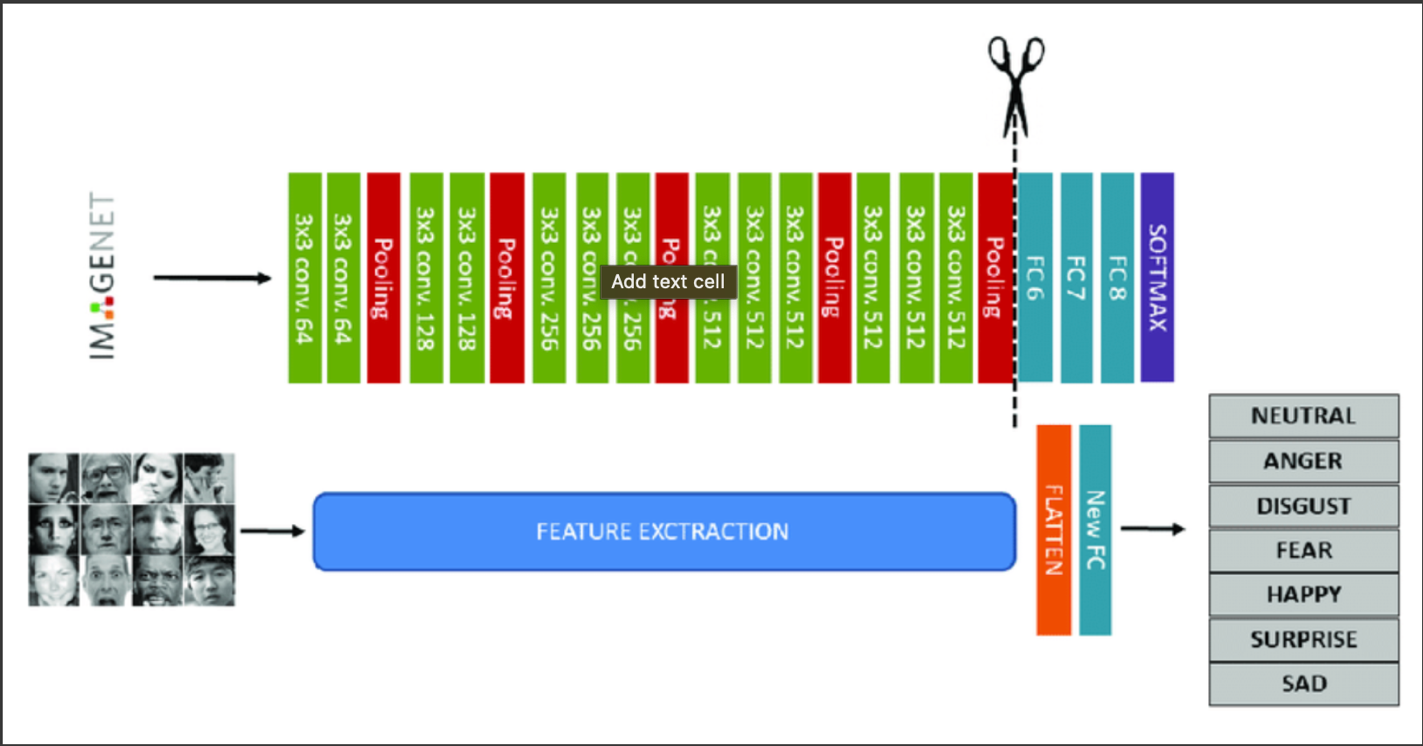




However, still the model is unable to predict disgust class.

Therefore we cannot rely on this model.

1. **CNN Model with VGG16**



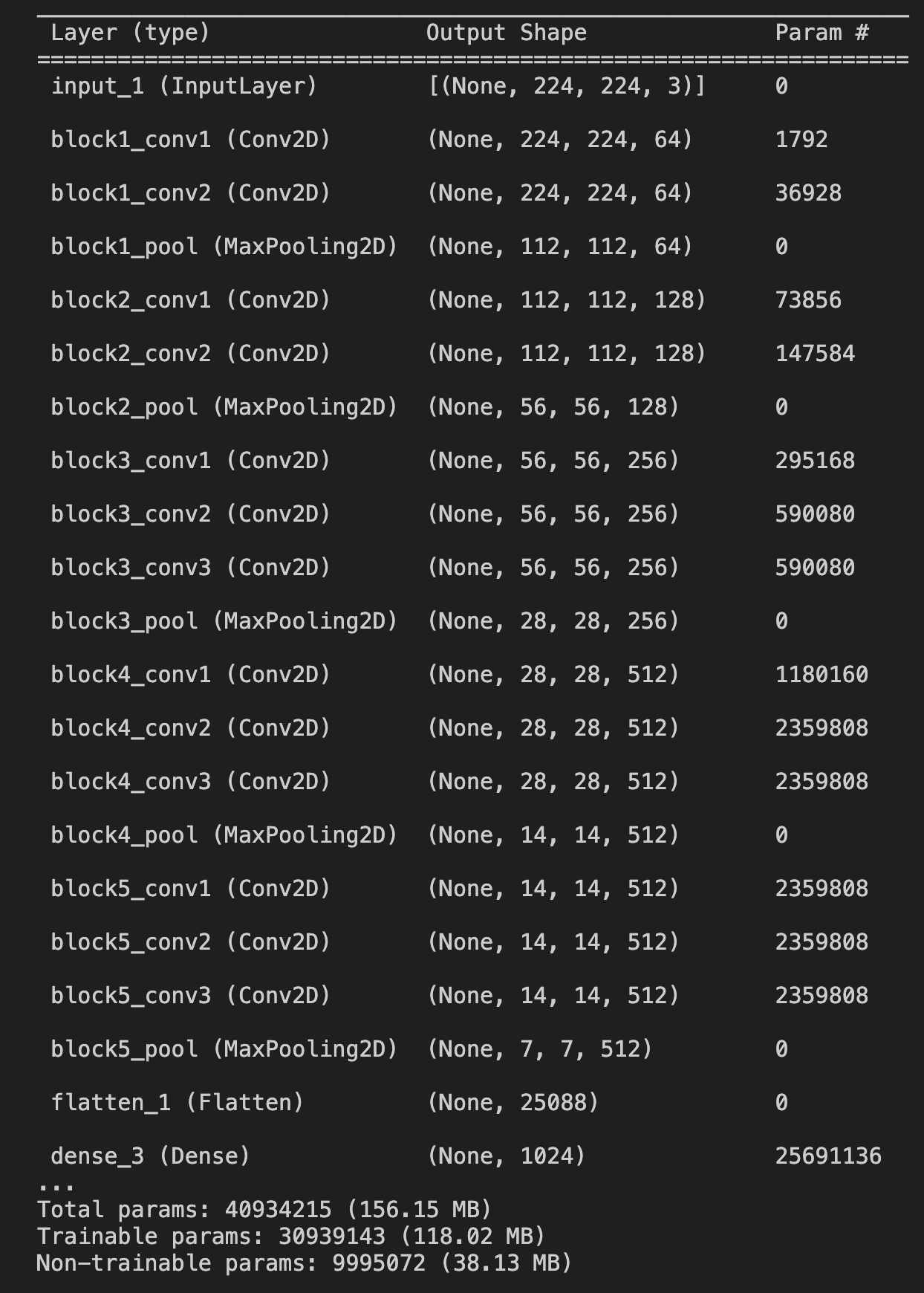
VGG16 Architecture

During the train, first of all, we need to change the size of the image from (48,48,1) to (224,224,3) in order to train the model, since these pre-trained model only accept images in

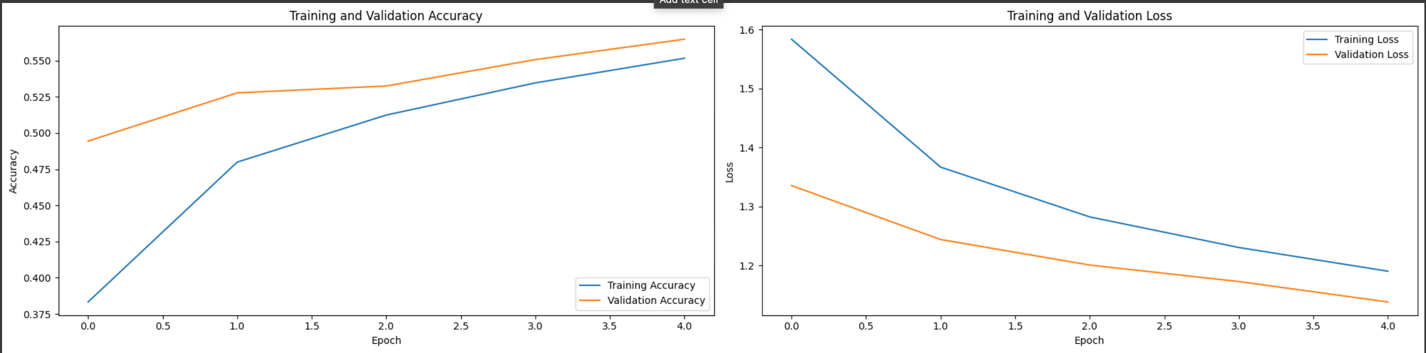
(224,224,3) format.

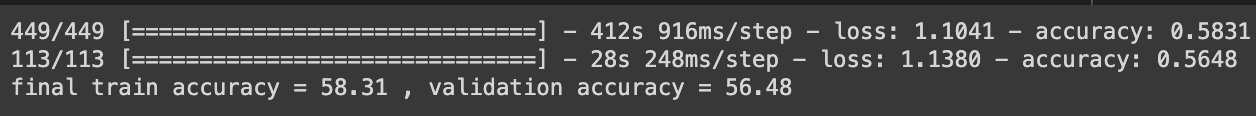
To train VGG16 for custom data, we are removing last 3 layers from this model. Instead, we will build our own layers to train the model, according to our dataset/requirements.

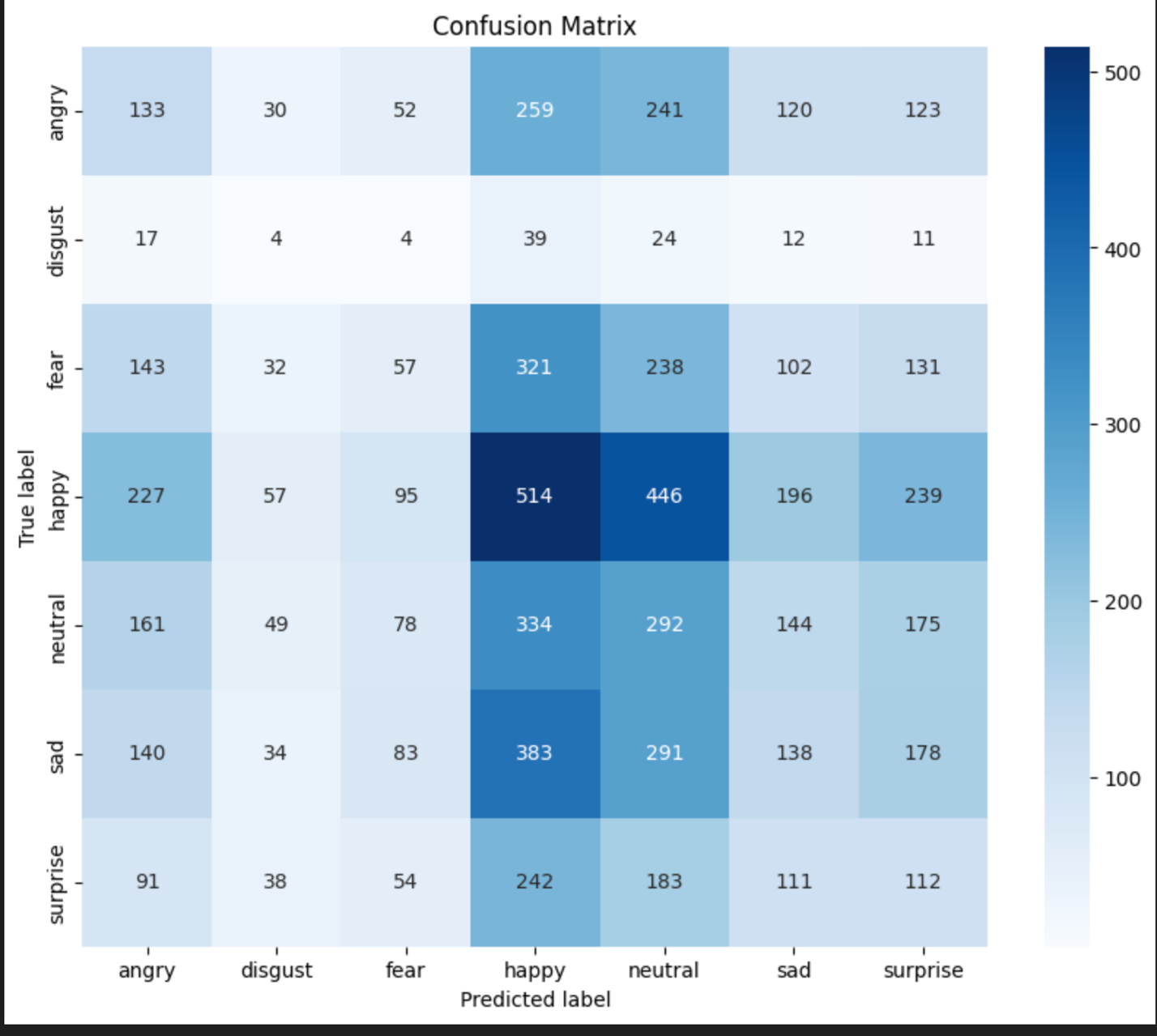
VGG16 Model Summary:



There are 40,934,215 parameters, out of which 30,939,143 are trainable parameters and 9995072 are non-trainable parameters.

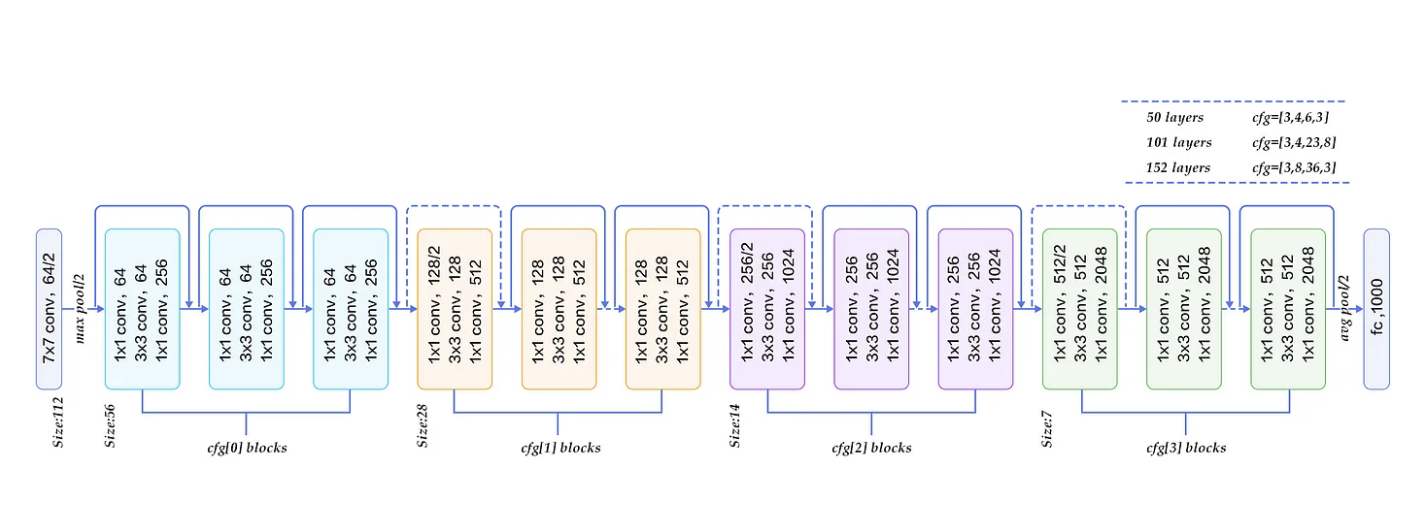






Here the accuracy of the model increases. As seen in the above matrix, the model is able to predict some disgust images.

1. **CNN Model with ResNet50**

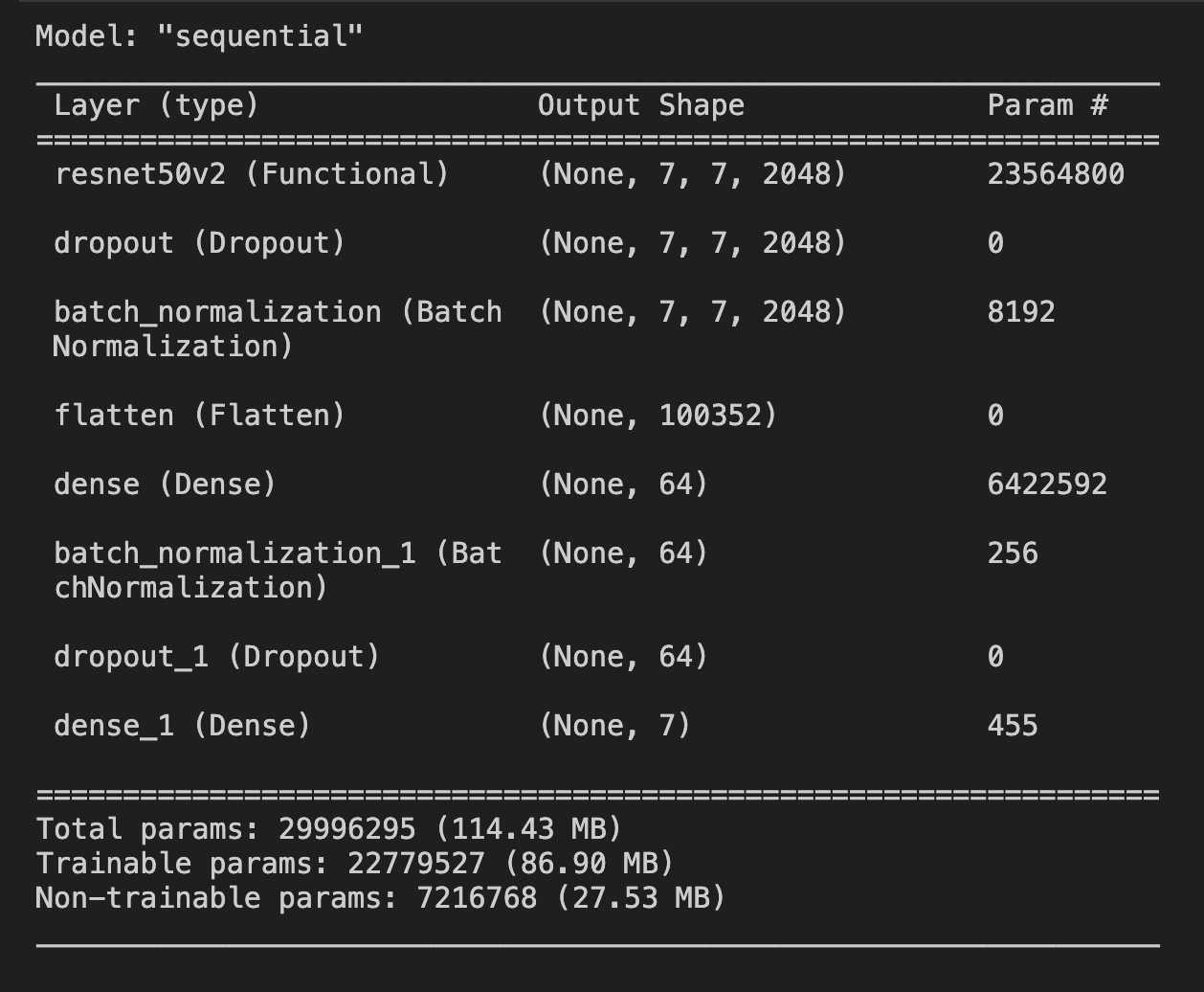


During the train, first of all, we need to change the size of the image from (48,48,1) to (224,224,3) in order to train the model, since these pre-trained model only accept images in

(224,224,3) format.

To train ResNet50 for custom data, we are removing last 50 layers from this model. Instead, we will build our own layers to train the model, according to our dataset/requirements.

ResNet50 Model Summary:



In total, there are 29996295 parameters, out of which 22,779,527 are trainable parameters and 7,216,768 are non-trainable parameters in this ResNet50 model.

