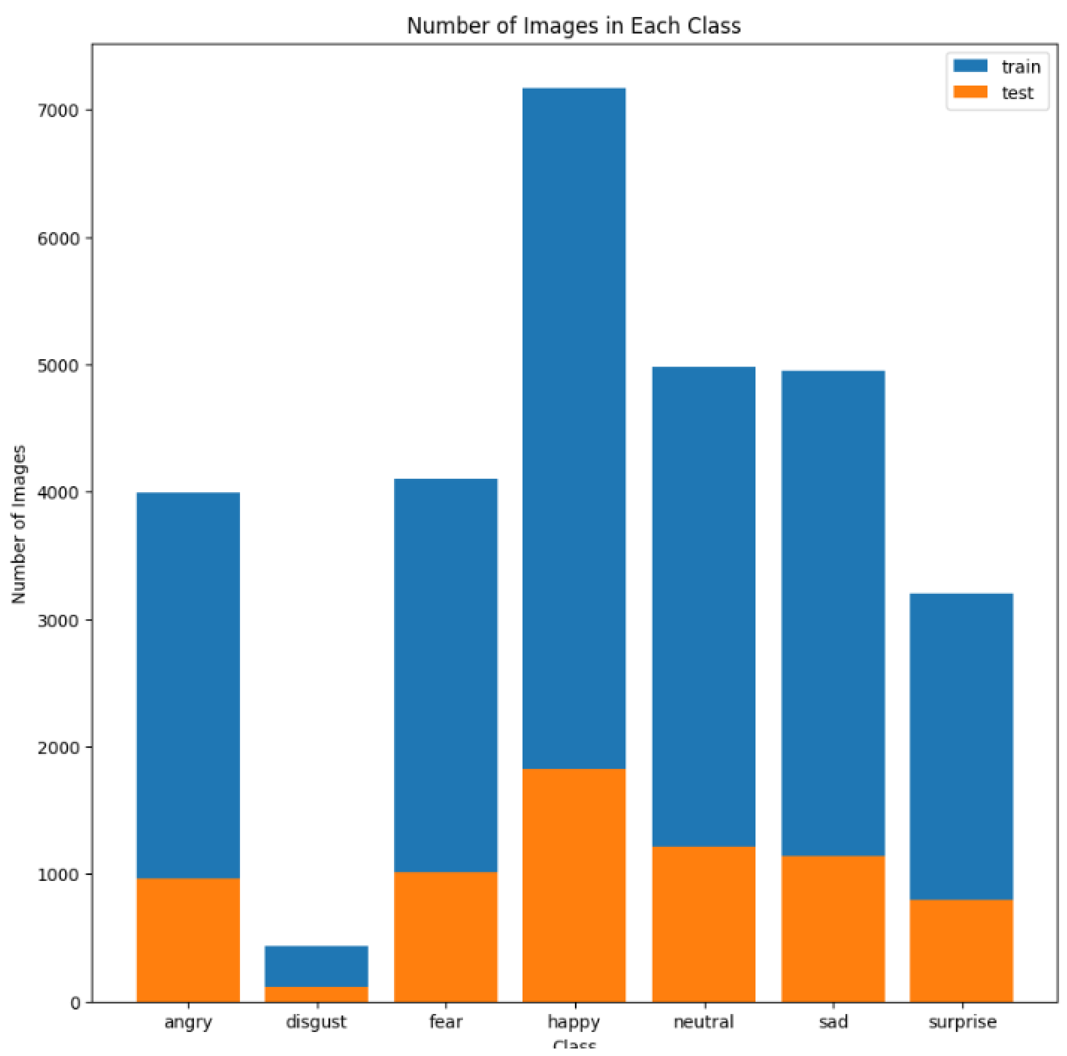
Write – Up for Emotion Recognition Project

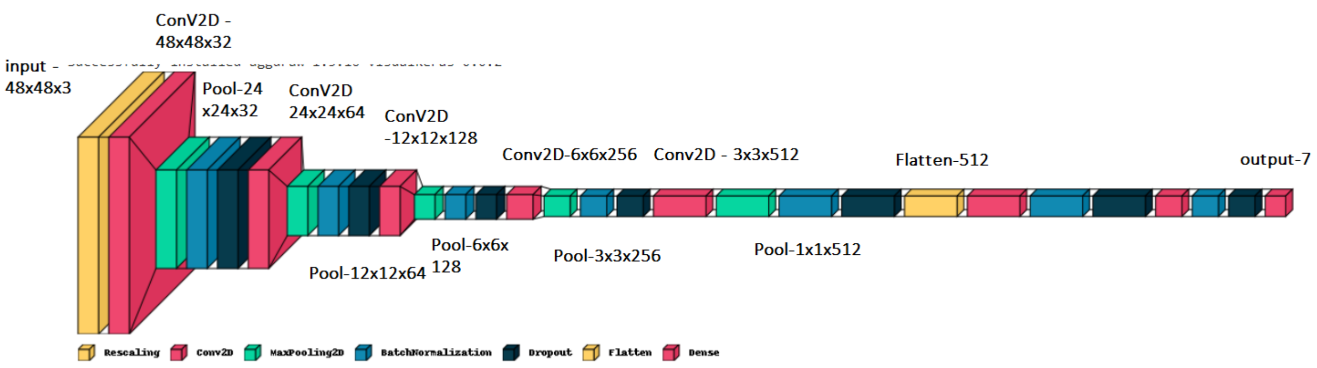
Number of images per class :



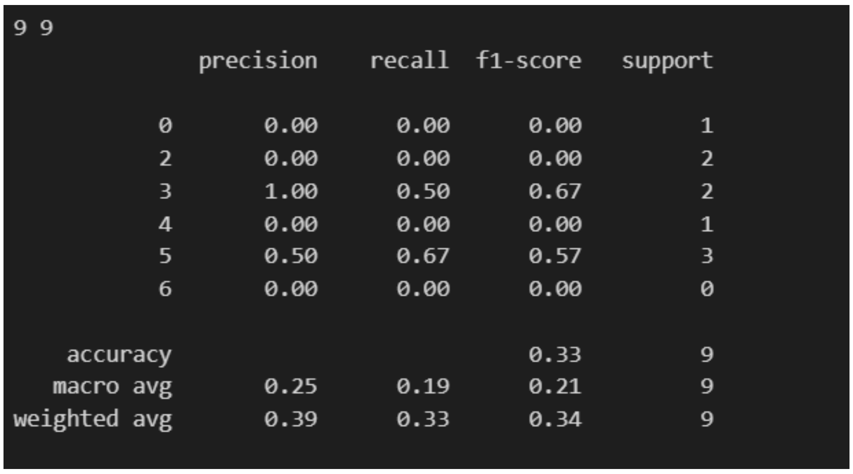
Sample classes :

```
['angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', 'surprise']
```

1. Customised CNN with additional layers (Emotion1.ipynb):



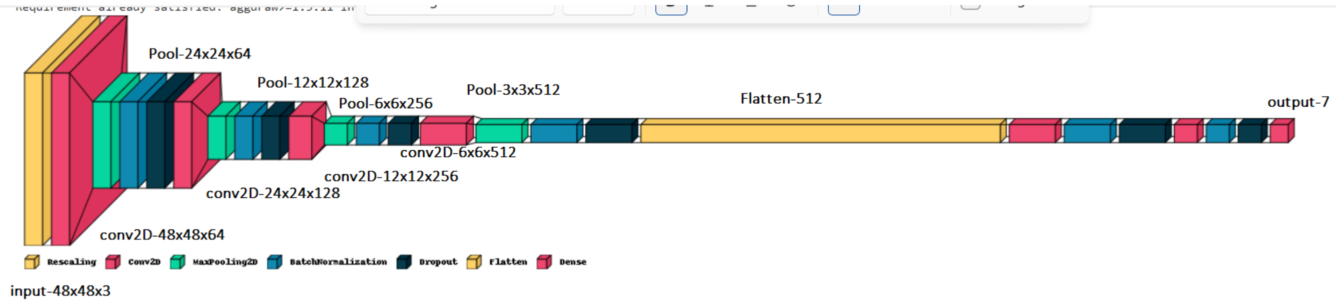
Scores:


```
9 9
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           2       0.00      0.00      0.00         2
           3       1.00      0.50      0.67         2
           4       0.00      0.00      0.00         1
           5       0.50      0.67      0.57         3
           6       0.00      0.00      0.00         0

    accuracy                           0.33         9
   macro avg       0.25      0.19      0.21         9
weighted avg       0.39      0.33      0.34         9
```

2.Base CNN(Emotion2.ipynb):



Scores:

```
9 9
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           2       1.00      0.50      0.67         2
           3       1.00      0.50      0.67         2
           4       0.33      1.00      0.50         1
           5       0.00      0.00      0.00         2
           6       0.67      1.00      0.80         2

    accuracy                           0.56         9
   macro avg       0.50      0.50      0.44         9
weighted avg       0.63      0.56      0.53         9
```

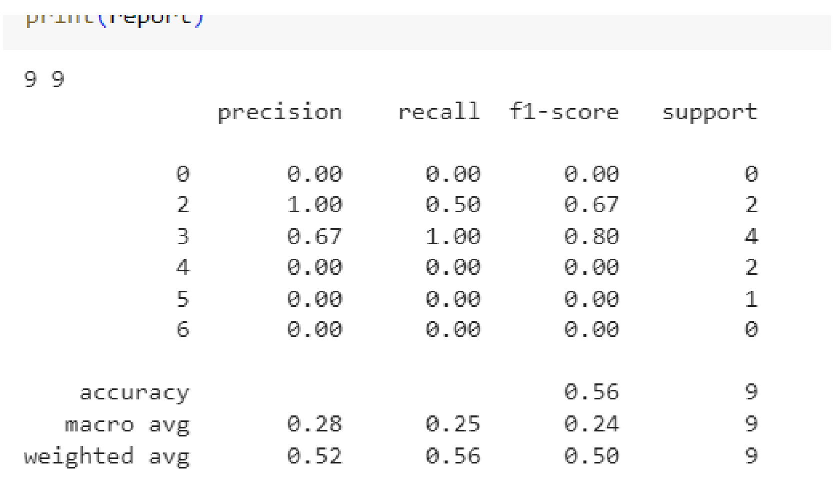
3. Transfer Learning with EfficientNetB2 :

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 rescaling_3 (Rescaling)     (None, 48, 48, 3)         0         
                                                                 
 conv2d_14 (Conv2D)          (None, 48, 48, 64)        1792      
                                                                 
 max_pooling2d_14 (MaxPoolin  (None, 24, 24, 64)       0         
 g2D)                                                            
                                                                 
 batch_normalization_20 (Bat  (None, 24, 24, 64)       256       
 chNormalization)                                                
                                                                 
 dropout_20 (Dropout)        (None, 24, 24, 64)        0         
                                                                 
 conv2d_15 (Conv2D)          (None, 24, 24, 128)       204928    
                                                                 
 max_pooling2d_15 (MaxPoolin  (None, 12, 12, 128)      0         
 g2D)                                                            
                                                                 
 batch_normalization_21 (Bat  (None, 12, 12, 128)      512       
 chNormalization)                                                
                                                                 
 dropout_21 (Dropout)        (None, 12, 12, 128)       0         
                                                                 
 conv2d_16 (Conv2D)          (None, 12, 12, 256)       295168    
                                                                 
 max_pooling2d_16 (MaxPoolin  (None, 6, 6, 256)        0         
 g2D)                                                            
                                                                 
 batch_normalization_22 (Bat  (None, 6, 6, 256)        1024      
 chNormalization)                                                
                                                                 
 dropout_22 (Dropout)        (None, 6, 6, 256)         0         
                                                                 
 conv2d_17 (Conv2D)          (None, 6, 6, 512)         1180160   
                                                                 
 max_pooling2d_17 (MaxPoolin  (None, 3, 3, 512)        0         
 g2D)                                                            
                                                                 
 batch_normalization_23 (Bat  (None, 3, 3, 512)        2048      
 chNormalization)                                                
                                                                 
 dropout_23 (Dropout)        (None, 3, 3, 512)         0         
                                                                 
 flatten_3 (Flatten)         (None, 4608)              0         
                                                                 
 dense_6 (Dense)             (None, 512)               2359808   
                                                                 
 batch_normalization_24 (Bat  (None, 512)              2048      
 chNormalization)                                                
                                                                 
 dropout_24 (Dropout)        (None, 512)               0         
                                                                 
 dense_7 (Dense)             (None, 256)               131328    
                                                                 
 batch_normalization_25 (Bat  (None, 256)              1024      
 chNormalization)                                                
                                                                 
 dropout_25 (Dropout)        (None, 256)               0         
 outputs (Dense)              (None, 7)                 1799

=================================================================
Total params: 4,181,895
Trainable params: 4,178,439
Non-trainable params: 3,456
```

```
The last fully connected layers are customised as per the dataset given.
```

Scores :



```
print(report)

9 9
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           2       1.00      0.50      0.67         2
           3       0.67      1.00      0.80         4
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1
           6       0.00      0.00      0.00         0

    accuracy                           0.56         9
   macro avg       0.28      0.25      0.24         9
weighted avg       0.52      0.56      0.50         9
```

Final Steps:

1.Accuracy for the basic CNN and transfer model seems to be 56% and is lower for the one with more layers. Also, recall, precision and F1 scores are better for basic CNN and EfficientNet compared to Customised CNN.

2. To improve model performance, more intricate data augmentation methods, weight initialisation methods can be implemented. Also, regularisation can help reduce the loss and improve accuracy. Shuffling can also be employed to improve randomness.