

DOCUMENTATION

How the model works?

- Initially rescaling of the data is done and then it goes through first convolution layer.
- In this layer, a convolution operation is applied to the input, passing the result to the next layer. A convolution converts all the pixels in its receptive field into a single value. We use the Relu activation function for this.
- The next step is batch normalization. This is done so that it normalizes the output of the previous layers.
- Next one is the second convolution layer. Here, every single filter gets applied separately to each of the feature maps. Then, batch normalization is done.
- After that is the pooling layer where it is used to reduce the dimensions of the feature maps.
- Later on, a series of these convolutional layers and pooling layers are applied so that they try to find out the suitable weights by backpropagation and reducing the loss.
- A dropout layer is also used in the process to avoid overfitting.
- At the end, we get model weights which are used later on.
- The model is trained for 10 epochs.

The following are the screenshots which depicts the loss and accuracy of training data and test data while the model is trained for 10 epochs.

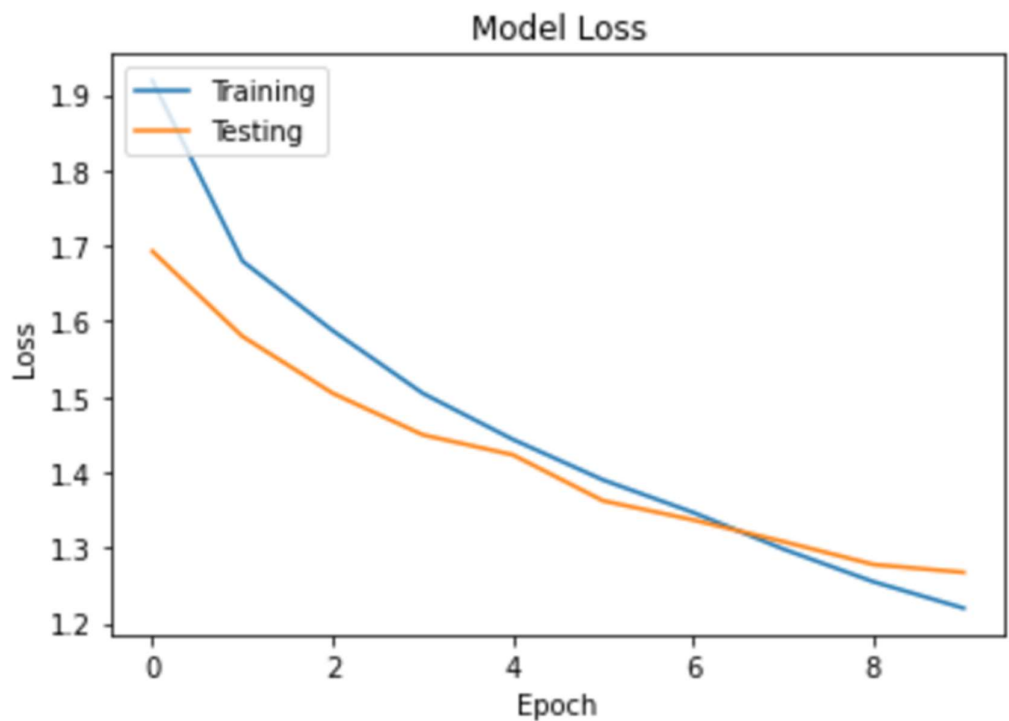
```
Epoch 1/10
898/898 [=====] - 978s 1s/step - loss: 1.9190 - accuracy: 0.2533 - val_loss: 1.6928 - val_accuracy: 0.3123
Epoch 2/10
898/898 [=====] - 986s 1s/step - loss: 1.6796 - accuracy: 0.3310 - val_loss: 1.5799 - val_accuracy: 0.3865
Epoch 3/10
898/898 [=====] - 842s 938ms/step - loss: 1.5877 - accuracy: 0.3744 - val_loss: 1.5047 - val_accuracy: 0.4224
Epoch 4/10
898/898 [=====] - 788s 877ms/step - loss: 1.5045 - accuracy: 0.4123 - val_loss: 1.4495 - val_accuracy: 0.4386
Epoch 5/10
898/898 [=====] - 751s 837ms/step - loss: 1.4434 - accuracy: 0.4386 - val_loss: 1.4230 - val_accuracy: 0.4508
Epoch 6/10
898/898 [=====] - 742s 826ms/step - loss: 1.3898 - accuracy: 0.4617 - val_loss: 1.3622 - val_accuracy: 0.4731
Epoch 7/10
898/898 [=====] - 742s 826ms/step - loss: 1.3467 - accuracy: 0.4810 - val_loss: 1.3368 - val_accuracy: 0.4838
Epoch 8/10
898/898 [=====] - 734s 817ms/step - loss: 1.2982 - accuracy: 0.5003 - val_loss: 1.3081 - val_accuracy: 0.4961
Epoch 9/10
898/898 [=====] - 740s 824ms/step - loss: 1.2548 - accuracy: 0.5189 - val_loss: 1.2777 - val_accuracy: 0.5057
```

```
Epoch 4/10
898/898 [=====] - 788s 877ms/step - loss: 1.5045 - accuracy: 0.4123 - val_loss: 1.4495 - val_accuracy: 0.4386
Epoch 5/10
898/898 [=====] - 751s 837ms/step - loss: 1.4434 - accuracy: 0.4386 - val_loss: 1.4230 - val_accuracy: 0.4508
Epoch 6/10
898/898 [=====] - 742s 826ms/step - loss: 1.3898 - accuracy: 0.4617 - val_loss: 1.3622 - val_accuracy: 0.4731
Epoch 7/10
898/898 [=====] - 742s 826ms/step - loss: 1.3467 - accuracy: 0.4810 - val_loss: 1.3368 - val_accuracy: 0.4838
Epoch 8/10
898/898 [=====] - 734s 817ms/step - loss: 1.2982 - accuracy: 0.5003 - val_loss: 1.3081 - val_accuracy: 0.4961
Epoch 9/10
898/898 [=====] - 740s 824ms/step - loss: 1.2548 - accuracy: 0.5189 - val_loss: 1.2777 - val_accuracy: 0.5057
Epoch 10/10
898/898 [=====] - 709s 790ms/step - loss: 1.2199 - accuracy: 0.5339 - val_loss: 1.2673 - val_accuracy: 0.5121
```

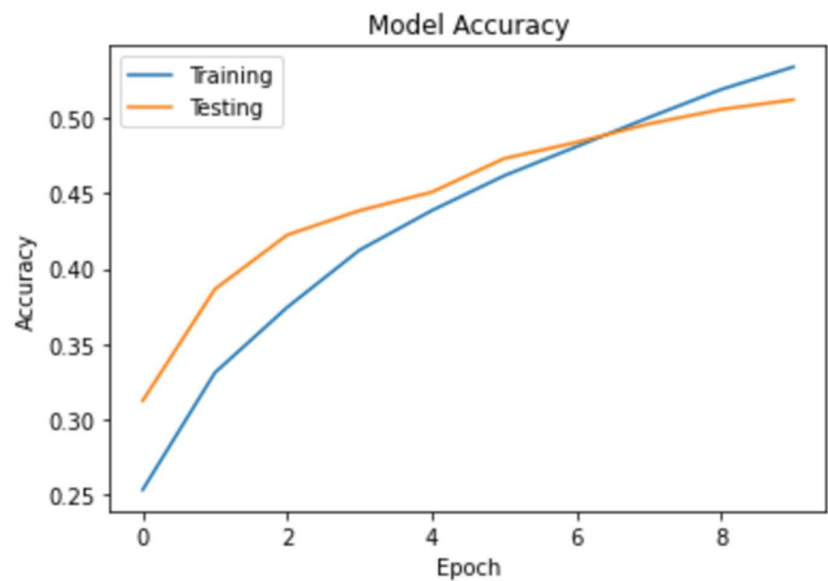
After running it for 10 epochs, the training accuracy obtained is **53.39%** and the test accuracy is **51.21%**.

The following graphs depict the model loss and model accuracy:

MODEL LOSS graph:

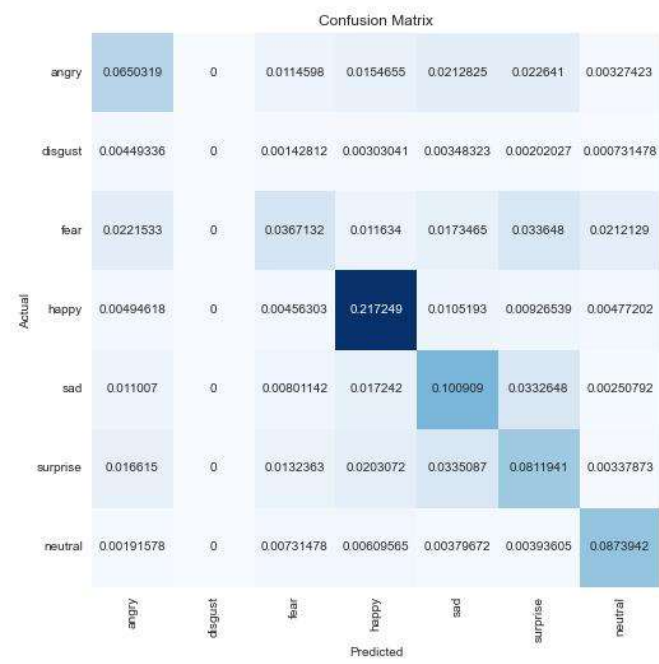


MODEL ACCURACY graph:



CONFUSION MATRIX:

Train data:



Test data:

		Confusion Matrix						
Actual	angry	0.0484815	0	0.0130956	0.0197827	0.0213151	0.0263305	0.00445807
	disgust	0.00585121	0	0.00139315	0.00306492	0.00334355	0.00125383	0.000557258
	fear	0.0190861	0	0.0281415	0.017275	0.0243801	0.0320424	0.0217331
	happy	0.00780162	0	0.00849819	0.194762	0.0151853	0.0143494	0.00654778
	sad	0.0135135	0	0.00933408	0.0229869	0.0883254	0.0330176	0.00459738
	surprise	0.0207579	0	0.0143494	0.0261911	0.0363611	0.0704932	0.00557258
	neutral	0.00390081	0	0.0118417	0.00738367	0.00585121	0.00487601	0.081917
		Predicted						
		angry	disgust	fear	happy	sad	surprise	neutral

Flowchart of the model process

