

EMOTION DETECTION

Steps for Data Download (Google Colab):

- 1) Install Kaggle using “!pip install kaggle”.
- 2) Download the required dataset after importing kaggle json.
- 3) Unzip the dataset (training and test) and place it in required path.
- 4) Define the path for Training and Test Data.

Total Train Data: 28709

Total Test Data: 7178

Models Used for Training:

Four different models were trained in multiple iterations.

- 1) Pre trained Model- VGG16 Model
- 2) Pre trained Model- RESNET50
- 3) Pre trained Model- InceptionV3
- 4) User Defined Model

Steps for Model Training (Pre-trained Model):

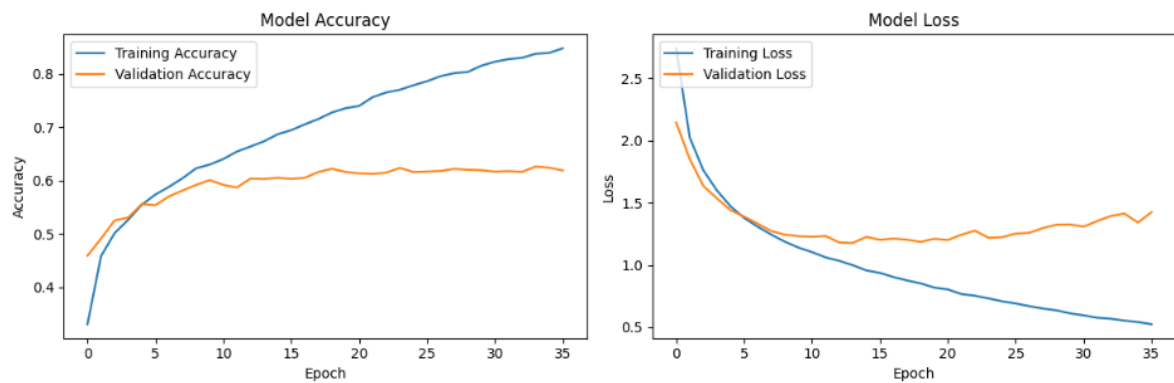
- 1) Data Augmentation and rescaling the data using Image Generator.
- 2) Load Images for Training and Validation (20% of Images from Training Dataset reserved for Validation) and define the batch size.
- 3) Define the Pre trained Model as base model and freeze its layers from getting trained again.
- 4) If required can train upper layers of the Pre trained model alone by unfreezing specific layers alone.
- 5) Add custom layers to fit the Classes based on the Dataset.
- 6) Compile the model with the specified loss function, optimizer, and metrics.
- 7) Define Early Stopping, Reduce Learning Rate (ReduceLROnPlateau) and number of epochs with required configuration.
- 8) Train the model in multiple Iterations to improve the Accuracy and reduce Loss.
- 9) Define Image Generator for Test and Load Test Dataset and use the Model for Prediction on Test Dataset.

Steps for Model Training (User Defined Model):

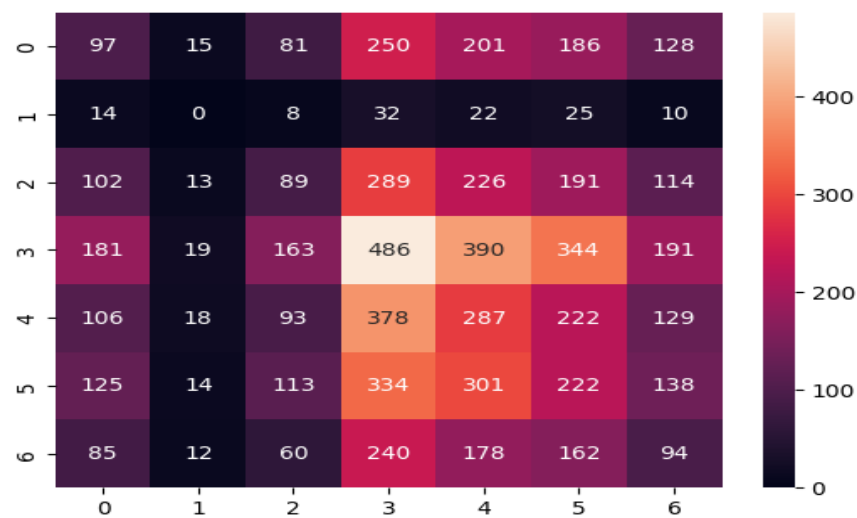
- 1) Data Augmentation and rescaling the data using Image Generator.
- 2) Load Images for Training and Validation (20% of Images from Training Dataset reserved for Validation) and define the batch size.
- 3) Define the layers to be added in the CNN.
- 4) Compile the model with the specified loss function, optimizer, and metrics.
- 5) Define Early Stopping, Reduce Learning Rate (ReduceLROnPlateau) and number of epochs with required configuration.
- 6) Train the model in multiple Iterations to improve the Accuracy and reduce Loss.
- 7) Define Image Generator for Test and Load Test Dataset and use the Model for Prediction on Test Dataset.

ANALYSIS and INFERENCE

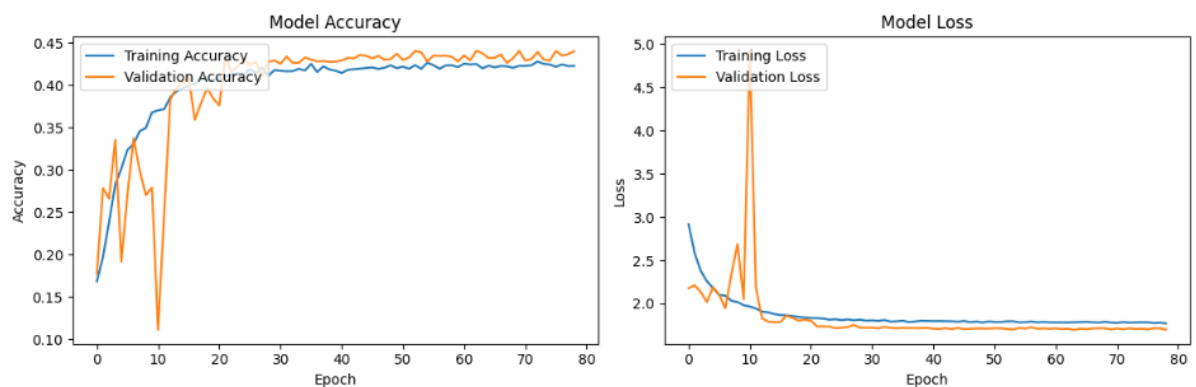
VGG16 Model



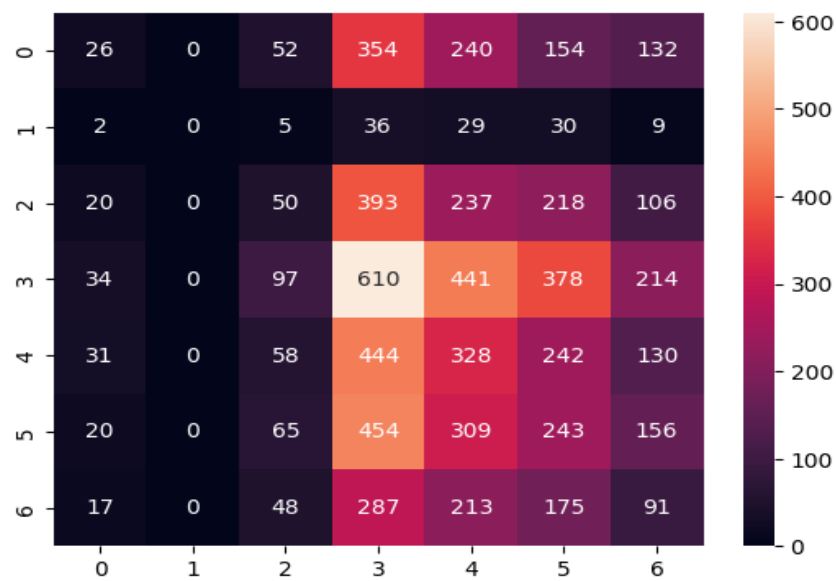
Prediction using Test data:



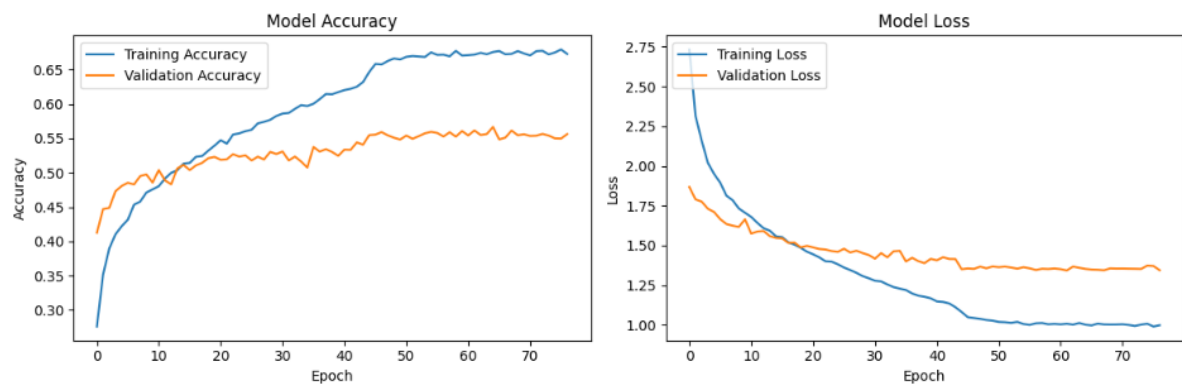
RESNET50 Model



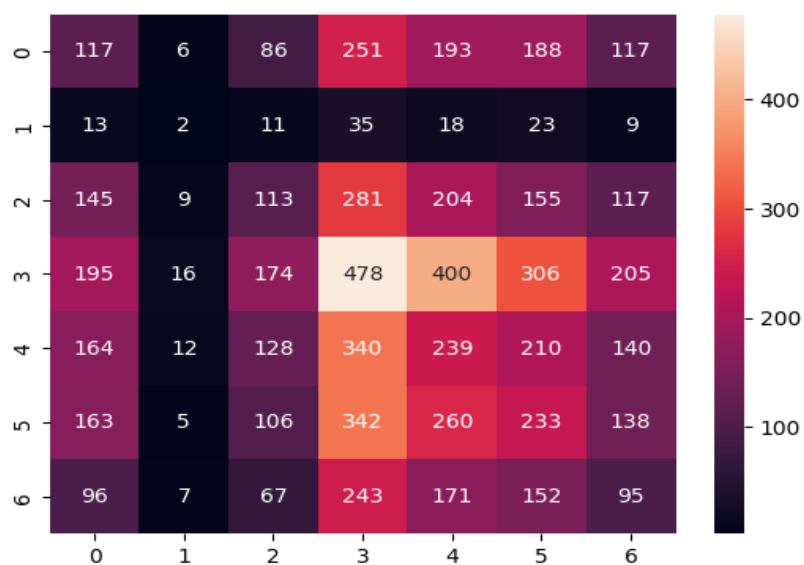
Prediction using Test data:



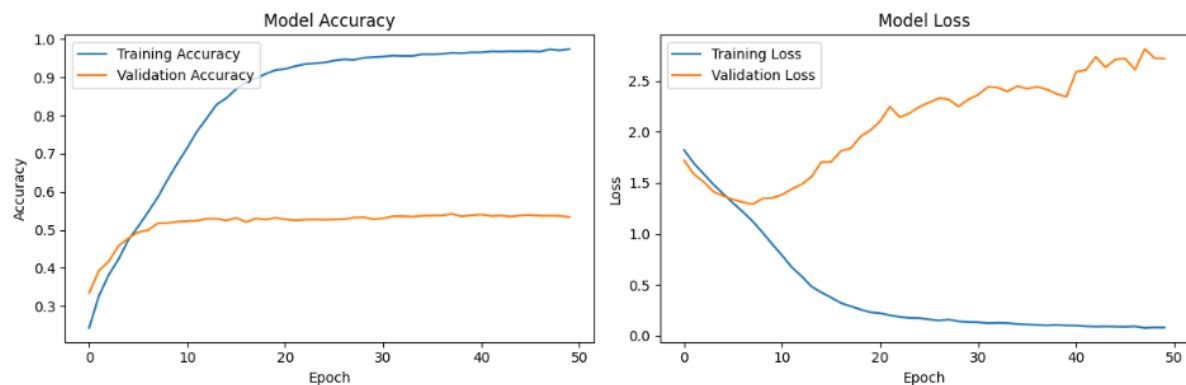
InceptionV3



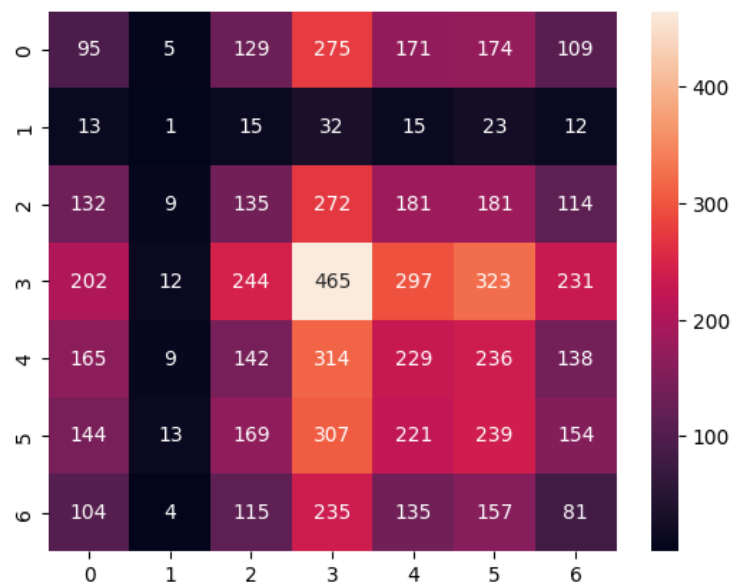
Prediction using Test data:



User Defined Model



Prediction using Test data:



Inference Points:

- 1) VGG16 gave a better accuracy when compared to other 2 pre trained models.
- 2) Inspite of multiple tuning iterations, RESNET provided the least accuracy among all the models.
- 3) Accuracy was highest for the user defined model.
- 4) The training of user defined model took a long time as it needed all the layers to be trained from scratch.
- 5) The class type disgust had the less number of true positives across the models.
- 6) The class type Happy had the highest true positives across the models.