AAI 627-A BIG DATA ANALYTICS
FINAL PROJECT

EMOTION BASED MUSIC RECOMMENDER USING DEEP LEARNING

NAME : MONISH KANNA SURESH
STEVENS ID : 20012129

OUTLINE

1. INTRODUCTION

1a. WHY FACIAL EMOTION RECOGNITION (FER)?

- The process of human communication is inextricably linked to the fluctuation of various emotions
- When people are experiencing basic emotions, their faces will display a variety of expression patterns, each with its own set of characteristics and distribution scale
- Facial expression recognition is a crucial part of human-computer interaction that allows computers to understand facial expressions based on human thinking
- According to the processing of facial expression recognition process can be divided into three important face detection, feature extraction and classification module
- Face detection as the key technology of face recognition with its rapid development has basic mature, which can effectively extracted from the original face image of excellent characteristics and the characteristics of correct classification becomes key factor affecting the recognition result

1b. WHY MUSIC RECOMMENDER?

- With commercial music streaming service which can be accessed from mobile devices, the availability of digital music currently is abundant compared to previous era
- It’s is very useful to develop a music recommender system that can search in the music libraries automatically and suggest suitable songs to users

2. RELATED WORK

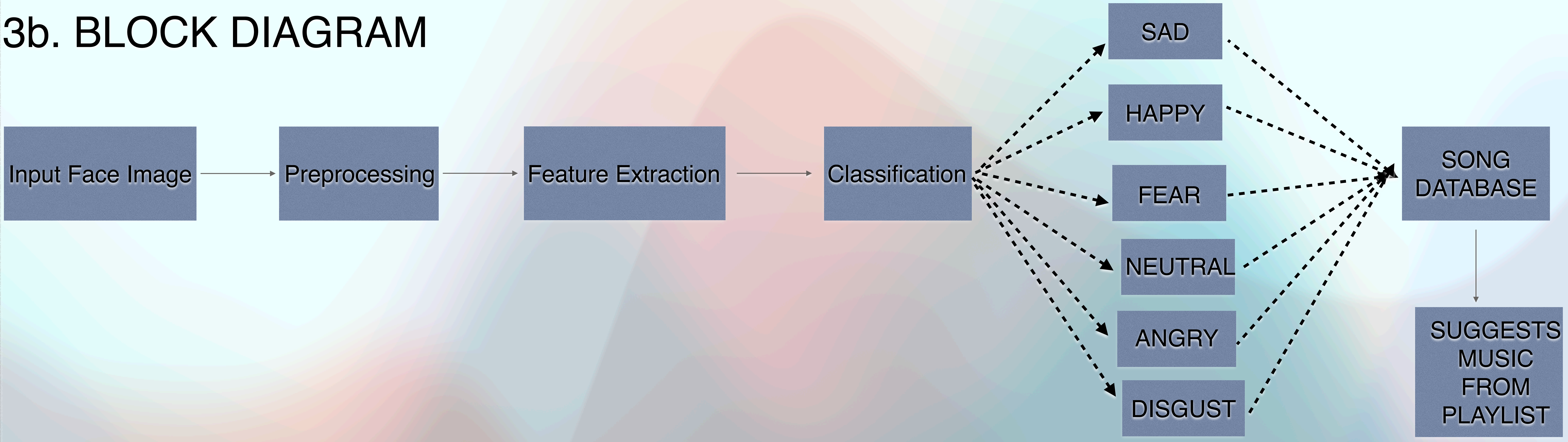
- A few methodologies have been proposed and embraced to group human feelings successfully. Most of the methodologies laid their emphasis on seven essential feelings which are steady over age culture or different characters.
- Describes the advantages of using OpenCV, especially the Adaboost algorithm, in the process of face recognition.
- Detecting and recognition of face in complicated colour images can be achieved using a combination of a particular algorithm with AdaBoost algorithm. It also talks about the disadvantages of using a timer in face detection.
- Proposes on utilising Support Vector Machines (SVM) as the primary characterisation technique to order eight facial feelings. The faces distinguished utilising channels in OpenCV and changed over to Greyscale. The paper likewise explains on robotized constant coding of outward appearances in nonstop video gushing, which is feasible for applications in which frontal perspectives can be accepted utilising webcam.
- The creator proposed a calculation to produce a subset of a unique playlist or a custom playlist related to the feeling perceived. The picture to be prepared was acquired from a web camera or the hard circle itself. The picture is expose to improvements, where a few mapping and upgrade procedures are connected to reestablish required differentiation of the picture. Preparing and arrangement are maintained by "one versus all" approach of SVM to encourage multi-class characterisation.
- Proposes on the utilisation of profound convolutional neural networks. It depends on solid face acknowledgment convolutional systems, which can be effectively tweaked to-play out the feeling acknowledgment task. Visual models are supplemented with sound highlights for better face acknowledgment.
- Aids in the music suggestion framework which is additionally a significant module of the proposed framework. It discusses highlights to be removed from the music to characterise its mind-set.

3. OBJECTIVE

3a. PROBLEM STATEMENT

The goal of this project is to learn about emotions of target using target's facial images and later categorise, label them with respective emotions and suggests some songs that alleviates the target's mood.

3b. BLOCK DIAGRAM



- In deep learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.
- Multi-label classification involves predicting one or more classes for each example and imbalanced classification refers to classification tasks where the distribution of examples across the classes is not equal.

3c. PROBLEM SOLUTION

- The software analyses the images of the user
- With the help of image segmentation and image processing techniques extracts features from the face of a target human being
- Later tries to detect the emotion that the person is trying to express
- The project further aims to lighten the mood of the user, by describing songs that match the requirements of the target by categorising the emotion of the target

+

4. TOOLS AND TECHNOLOGIES

4a. **GOOGLE COLAB NOTEBOOK** :

- Colab notebooks are Jupyter notebooks that run in the cloud and are highly integrated with Google Drive, making them easy to set up, access, and share.

4b. **PYTHON PROGRAMMING LANGUAGE** :

- Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.
- Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialised for any specific problems.

4c. **TENSORFLOW (deep learning framework)** :

- Open Source Software libraries used for Machine Learning and Artificial Intelligence. Tensorflow offers multiple data tools to help you consolidate, clean and preprocess data at scale.

4d. **KERAS (deep learning framework)** :

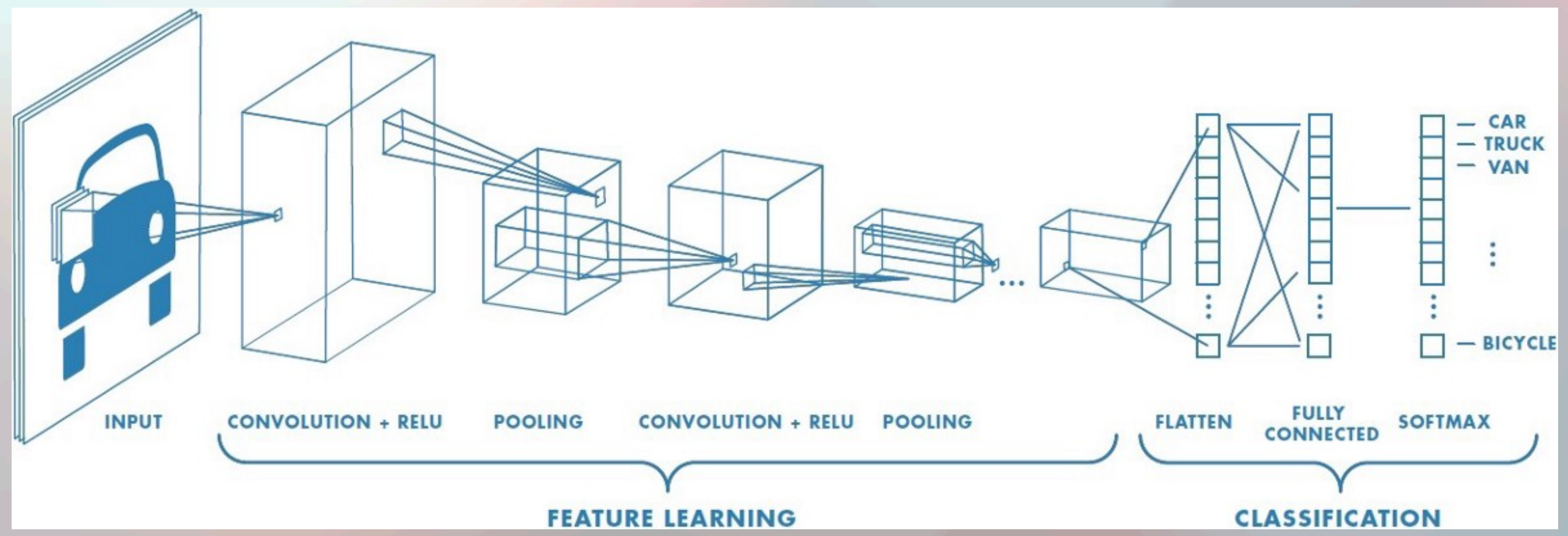
- Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

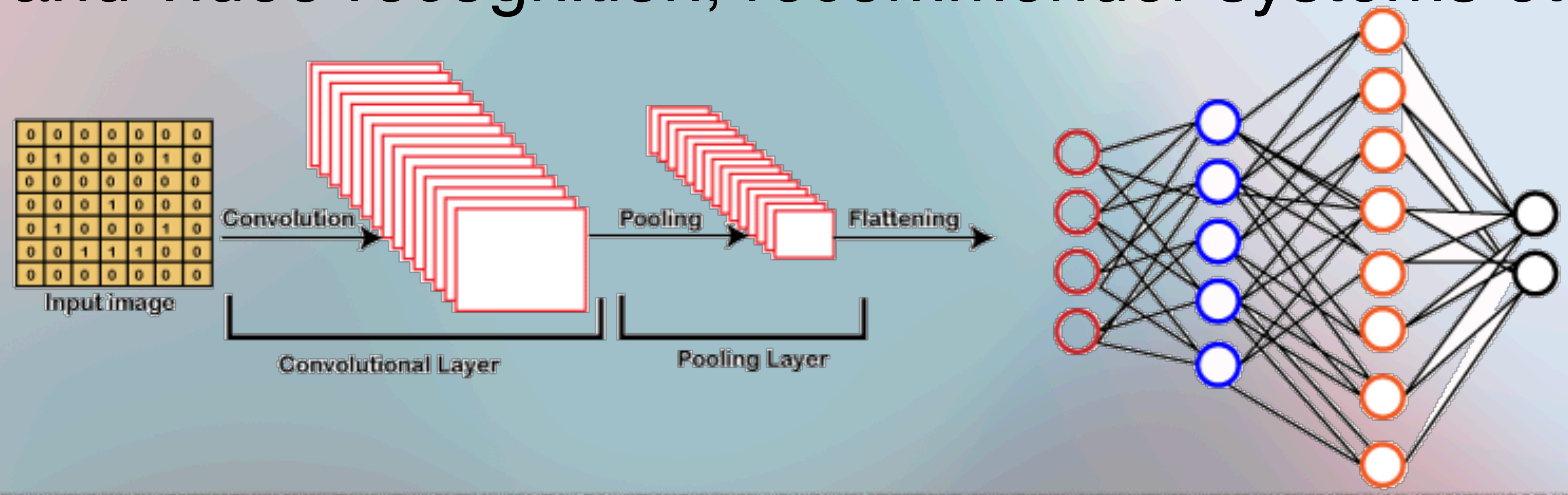
5. ALGORITHM

The approach to the problem solution is done by developing model using Convolutional Neural Networks(CNN) Algorithm and using pretrained ResNet50V2 model.

5a. What is CNN?

- In deep learning, a convolutional neural network (CNN or ConvNet) is a class of deep neural networks. CNNs, like neural networks, are made up of neurons with learnable weights and biases.
- Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output.
- A convolutional neural network (CNN) consists of an input and an output layer, as well as multiple hidden layers.
- The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product.
- CNNs have wide variety of applications in image and video recognition, recommender systems etc,..





5b. Why CNN?

- The CNNs have several different filters/kernels consisting of trainable parameters which can convolve on a given image spatially to detect features like edges and shapes
- These high number of filters essentially learn to capture spatial features from the image based on the learned weights through back propagation and stacked layers of filters can be used to detect complex spatial shapes from the spatial features at every subsequent level
- Hence they can successfully boil down a given image into a highly abstracted representation which is easy for predicting.
- Another advantage of CNN is that it automatically detects the important features without any human supervision

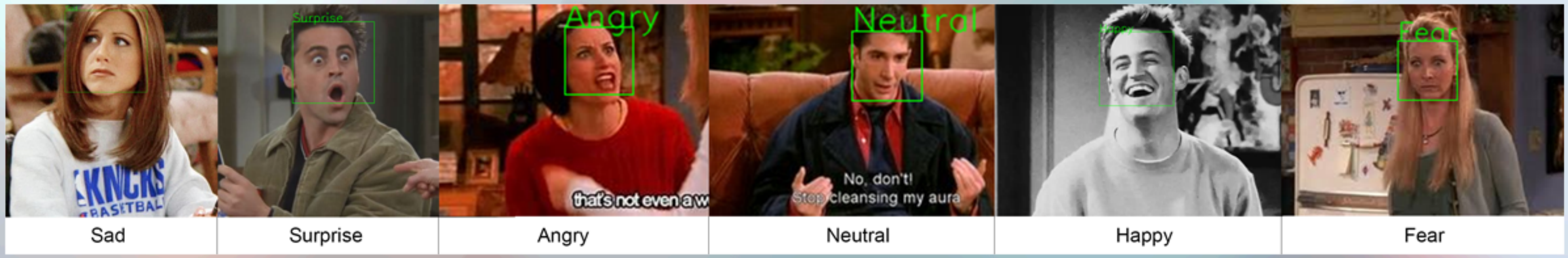
5c. Why ResNet50-V2?

- Resnet is short name for Residual Network that supports Residual Learning.
- The 50 indicates the number of layers that it has.
- Deep convolutional networks have led to number of breakthroughs for image classification.
- In general the trend is to go more deeper number of layers to solve complex tasks and to increase the classification and recognition accuracy.
- But as we go deeper with the neural networks the accuracy starts saturating and then degrades also. Residual training tries to solve this problem.

6. DATASET

FER-2013 (FACIAL EXPRESSION RECOGNITION 2013 DATASET)

- 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 categorise 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)
- The training set consists of 29163 examples
- The testing set consists of 7228 examples



7. IMPLEMENTATION AND PREDICTIONS

7a. IMPORTING LIBRARIES AND DEEP LEARNING FRAMEWORKS

```
from google.colab import drive
drive.mount('/content/drive')
```

```
    Drive already mounted at /content/drive; to attempt to forcibly remount, ca
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('default')
```

```
import os
import tensorflow as tf
import keras
import cv2
```

```
from sklearn.model_selection import train_test_split
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img, i
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLRO
from tensorflow.keras.utils import plot_model
from tensorflow.keras import layers , models, optimizers
```

```
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import *
from tensorflow.keras.applications import ResNet50V2
```

- PANDAS for analysis of dataset
- MATPLOTLIB for data visualisation and graphical plotting
- TENSORFLOW for data automation, model tracking, performance monitoring, and model retraining
- KERAS is used for rapid prototyping, and multiple back-end support for TENSORFLOW

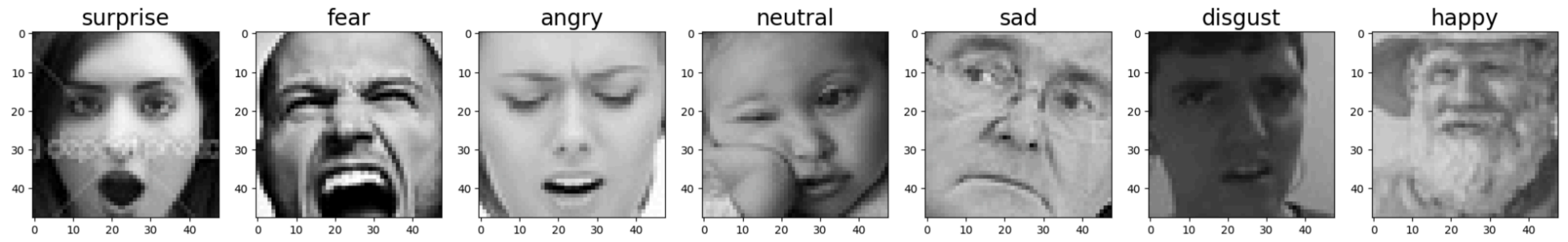
7b. LOADING DATASET

- FER-2013 Dataset was loaded into software
- Test Dataset and Train Dataset split are shown in Tabular and Graphical Format
- Every set are categorised into 7 emotions based on their facial expressions

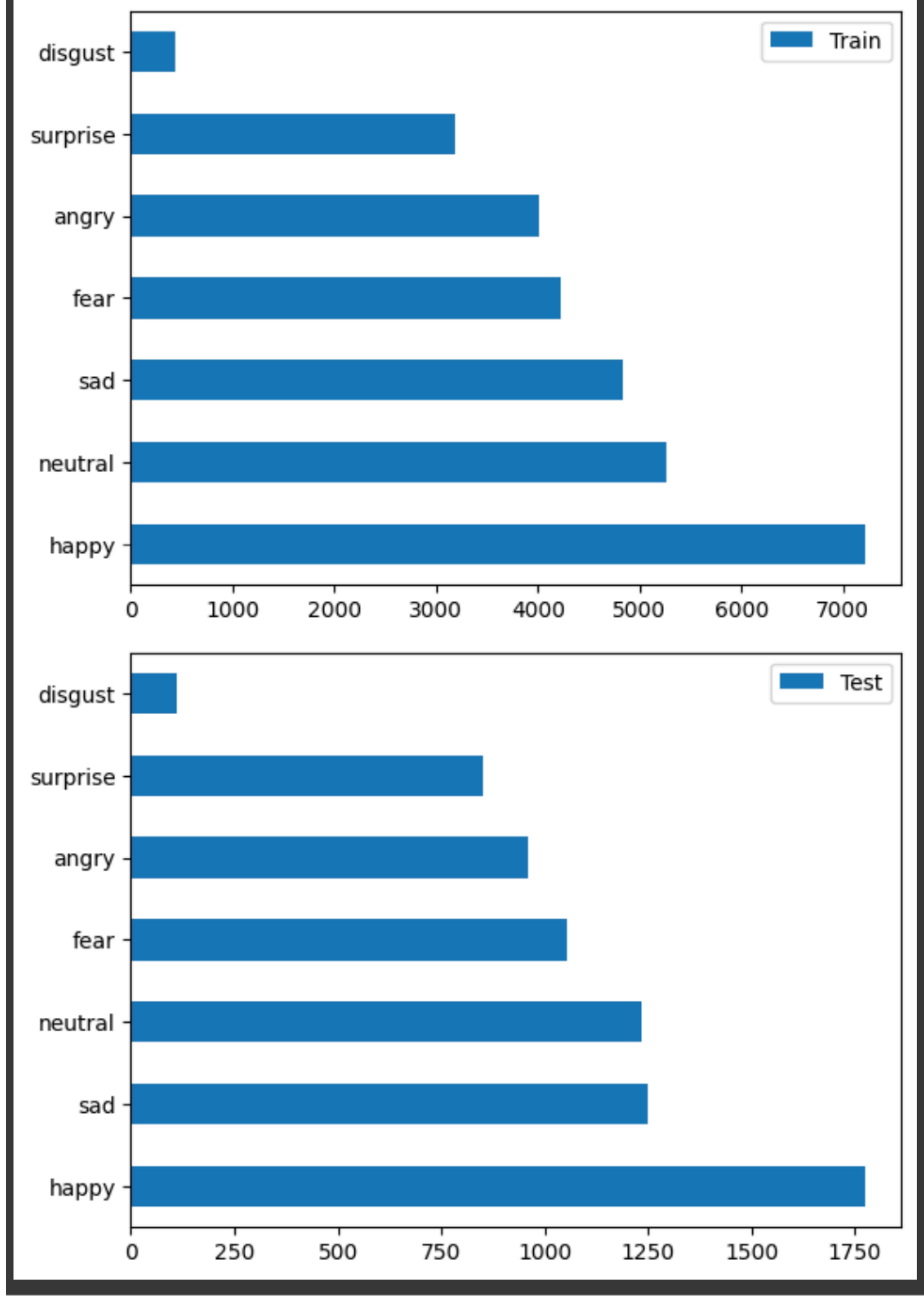
```
pd.concat([Train_Count,Test_Count] , axis=1)
```

	Train	Test
happy	7215	1774
neutral	5259	1233
sad	4830	1247
fear	4227	1054
angry	4015	958
surprise	3181	851
disgust	436	111

```
In [6]: plt.style.use('default')
        plt.figure(figsize = (25, 8))
        image_count = 1
        BASE_URL = '../input/fer2013/train/'

        for directory in os.listdir(BASE_URL):
            if directory[0] != '.':
                for i, file in enumerate(os.listdir(BASE_URL + directory)):
                    if i == 1:
                        break
                    else:
                        fig = plt.subplot(1, 7, image_count)
                        image_count += 1
                        image = cv2.imread(BASE_URL + directory + '/' + file)
                        plt.imshow(image)
                        plt.title(directory, fontsize = 20)
```



<matplotlib.axes._subplots.AxesSubplot at 0x7f0491447d10>



7c. CREATING CNN MODEL

- Designed 3 layer CNN Model for emotion recognition
- Model was designed using blocks of Conv2D layer, Batch-Normalisation, Max-Pooling2D, dropout, dense, flatten and stacking them together and at the end-use dense layer for output
- Conv2D layer: creates a convolutional kernel that is convolved with the layer input to produce a tensor of outputs
- Batch-Normalization: It’s is a technique for training very deep neural networks that normalizes the contributions to a layer for every mini-batch
- Max-Pooling2D: Max pooling operation for 2D spatial data. Downsamples the input along its spatial dimensions (height and width) by taking the maximum value over an input window (of size defined by pool_size) for each channel of the input. The window is shifted by strides along each dimension

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 32)	896
batch_normalization (BatchNo	(None, 46, 46, 32)	128
conv2d_1 (Conv2D)	(None, 46, 46, 64)	18496
batch_normalization_1 (Batch	(None, 46, 46, 64)	256
max_pooling2d (MaxPooling2D)	(None, 23, 23, 64)	0
dropout (Dropout)	(None, 23, 23, 64)	0
conv2d_2 (Conv2D)	(None, 21, 21, 64)	36928
batch_normalization_2 (Batch	(None, 21, 21, 64)	256
conv2d_3 (Conv2D)	(None, 21, 21, 128)	73856
batch_normalization_3 (Batch	(None, 21, 21, 128)	512
max_pooling2d_1 (MaxPooling2	(None, 11, 11, 128)	0
dropout_1 (Dropout)	(None, 11, 11, 128)	0
conv2d_4 (Conv2D)	(None, 9, 9, 128)	147584
batch_normalization_4 (Batch	(None, 9, 9, 128)	512
conv2d_5 (Conv2D)	(None, 9, 9, 256)	295168
batch_normalization_5 (Batch	(None, 9, 9, 256)	1024
max_pooling2d_2 (MaxPooling2	(None, 5, 5, 256)	0
dropout_2 (Dropout)	(None, 5, 5, 256)	0
flatten (Flatten)	(None, 6400)	0
dense (Dense)	(None, 1024)	6554624
batch_normalization_6 (Batch	(None, 1024)	4096
dropout_3 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 512)	524800
batch_normalization_7 (Batch	(None, 512)	2048
dropout_4 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
batch_normalization_8 (Batch	(None, 256)	1024
dropout_5 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
batch_normalization_9 (Batch	(None, 128)	512
dropout_6 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8256
batch_normalization_10 (Batc	(None, 64)	256
dropout_7 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 32)	2080
batch_normalization_11 (Batc	(None, 32)	128
dropout_8 (Dropout)	(None, 32)	0
dense_6 (Dense)	(None, 7)	231

Total params: 7,837,895
Trainable params: 7,832,519
Non-trainable params: 5,376

```
CNN_history = CNN_Model.fit( train_data , validation_data= test_data , epochs=50, batch_size= batch_size,
                            callbacks=callbacks, steps_per_epoch= steps_per_epoch, validation_steps=validation_steps
```

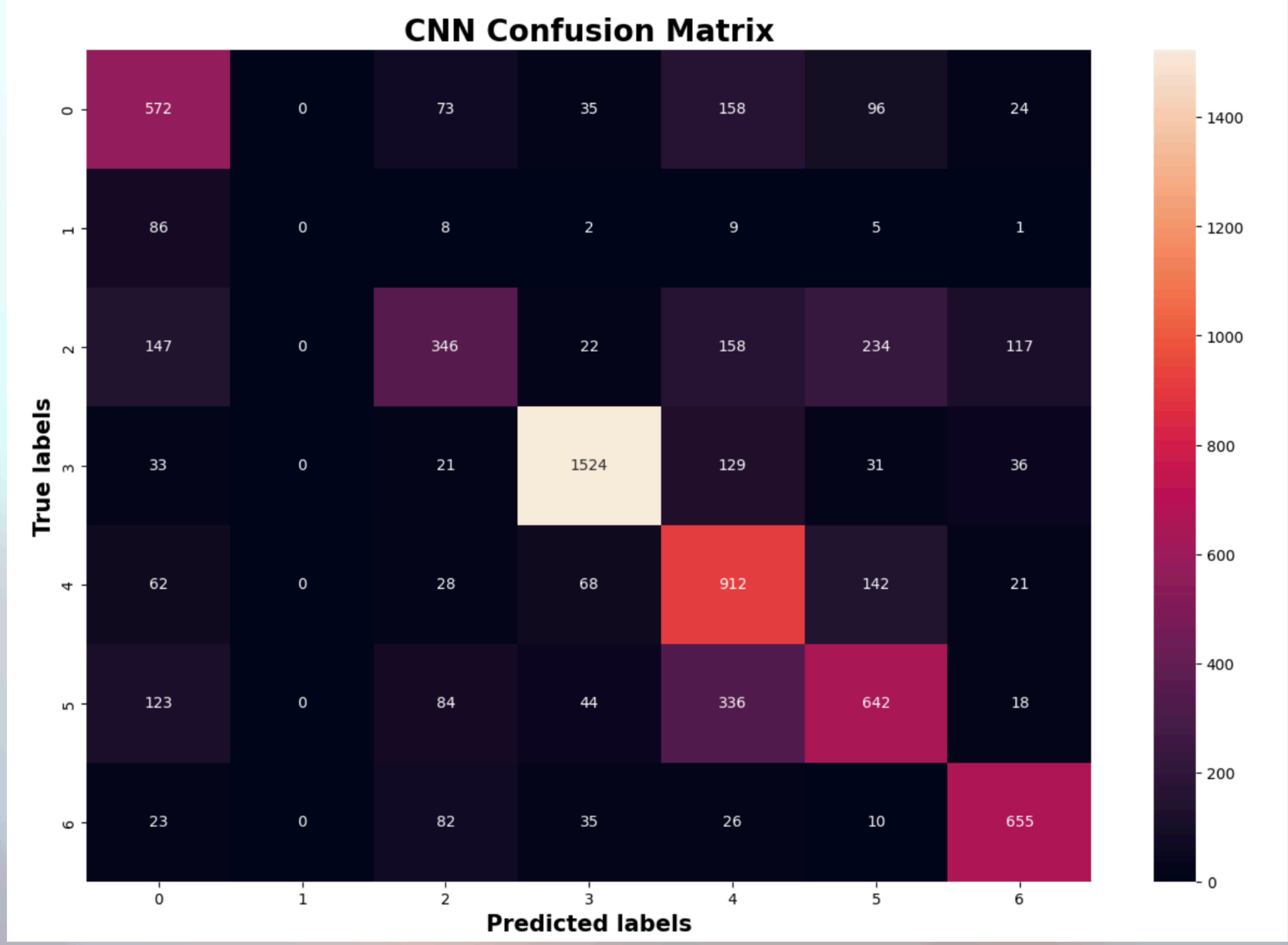
```
2022-11-16 22:56:39.667635: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimi
zation Passes are enabled (registered 2)

Epoch 1/50

2022-11-16 22:56:42.765207: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8005

448/448 [==============================] - 141s 296ms/step - loss: 2.0699 - accuracy: 0.2034 - val_loss: 1.8302 - v
al_accuracy: 0.2493
Epoch 2/50
448/448 [==============================] - 55s 122ms/step - loss: 1.7819 - accuracy: 0.2827 - val_loss: 1.6902 - va
l_accuracy: 0.3274
Epoch 3/50
448/448 [==============================] - 55s 122ms/step - loss: 1.6521 - accuracy: 0.3460 - val_loss: 1.5430 - va
l_accuracy: 0.4014
Epoch 4/50
448/448 [==============================] - 57s 128ms/step - loss: 1.5282 - accuracy: 0.4056 - val_loss: 1.3863 - va
l_accuracy: 0.4609
Epoch 5/50
448/448 [==============================] - 63s 141ms/step - loss: 1.4337 - accuracy: 0.4471 - val_loss: 1.7540 - va
l_accuracy: 0.3450
Epoch 6/50
448/448 [==============================] - 60s 135ms/step - loss: 1.3789 - accuracy: 0.4713 - val_loss: 1.2394 - va
l_accuracy: 0.5300
Epoch 7/50
448/448 [==============================] - 56s 125ms/step - loss: 1.3247 - accuracy: 0.4946 - val_loss: 1.5164 - va
l_accuracy: 0.3986
Epoch 8/50
448/448 [==============================] - 55s 123ms/step - loss: 1.2875 - accuracy: 0.5112 - val_loss: 1.1664 - va
l_accuracy: 0.5544
Epoch 9/50
448/448 [==============================] - 54s 122ms/step - loss: 1.2647 - accuracy: 0.5238 - val_loss: 1.2448 - va
l_accuracy: 0.5243
Epoch 10/50
448/448 [==============================] - 55s 122ms/step - loss: 1.2330 - accuracy: 0.5370 - val_loss: 1.2362 - va
l_accuracy: 0.5315

Epoch 00010: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
```

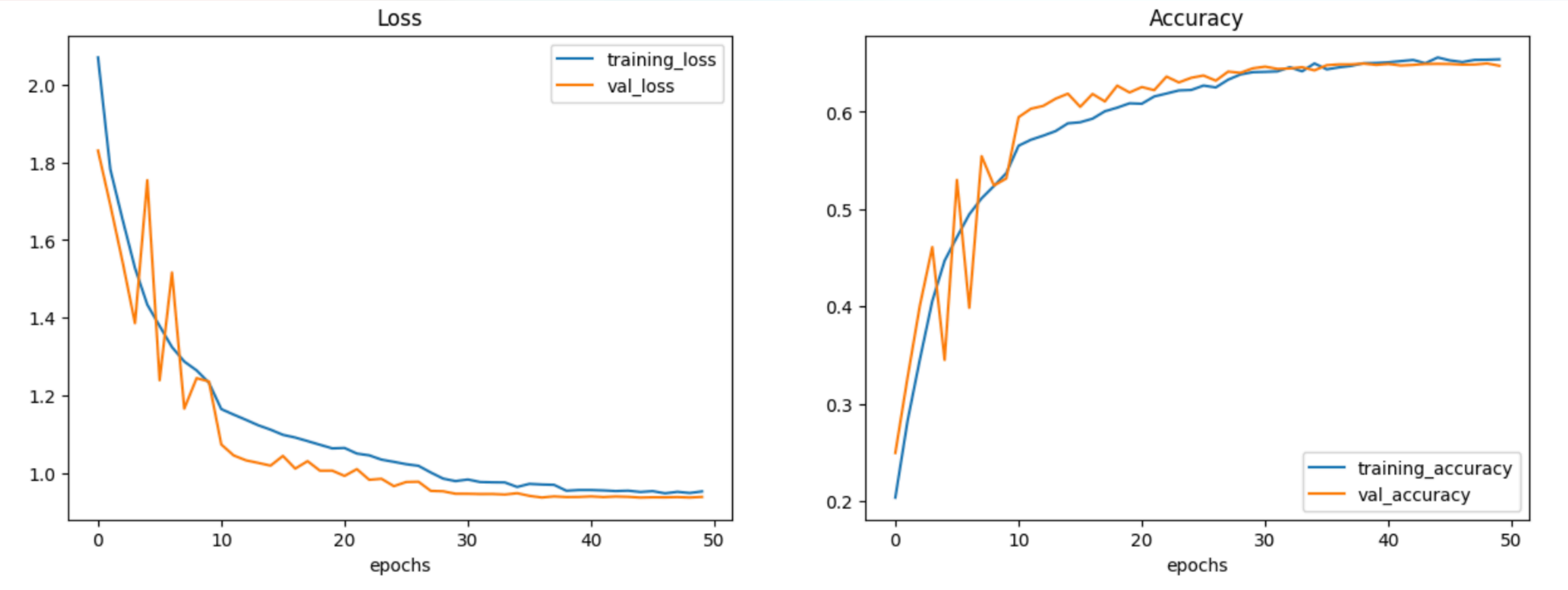
7d. EVALUATION OF CNN MODEL

- Loss value implies how poorly or well a model behaves after each iteration of optimisation
- Accuracy value of a model implies the proportion of true positive results in the selected population

TEST ACCURACY OF CNN MODEL : 64.80%

```
In [13]: CNN_Score = CNN_Model.evaluate(test_data)
         print("Test Loss: {:.5f}".format(CNN_Score[0]))
         print("Test Accuracy: {:.2f}%".format(CNN_Score[1] * 100))

         113/113 [==============================] - 6s 56ms/step - loss: 0.9381 - accuracy: 0.6480
             Test Loss: 0.93807
         Test Accuracy: 64.80%
```





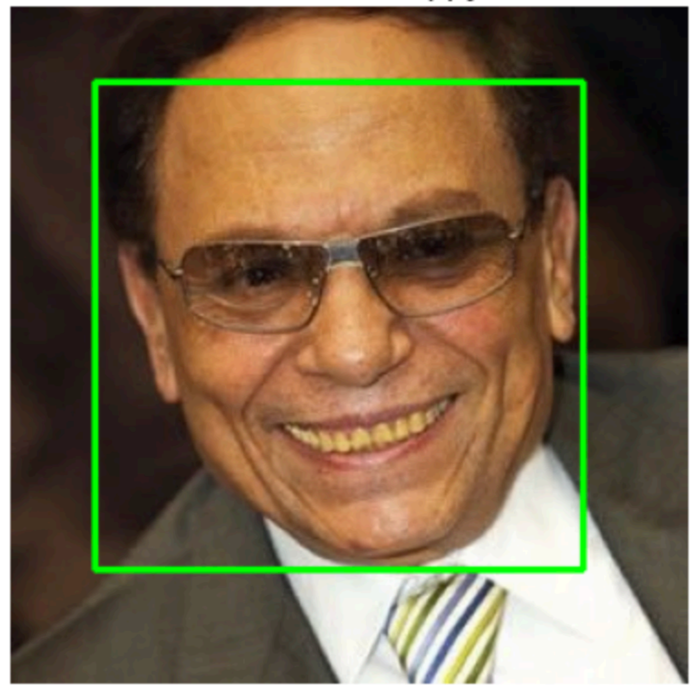
7e. PREDICTION BY CNN MODEL

	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Prediction: Happy

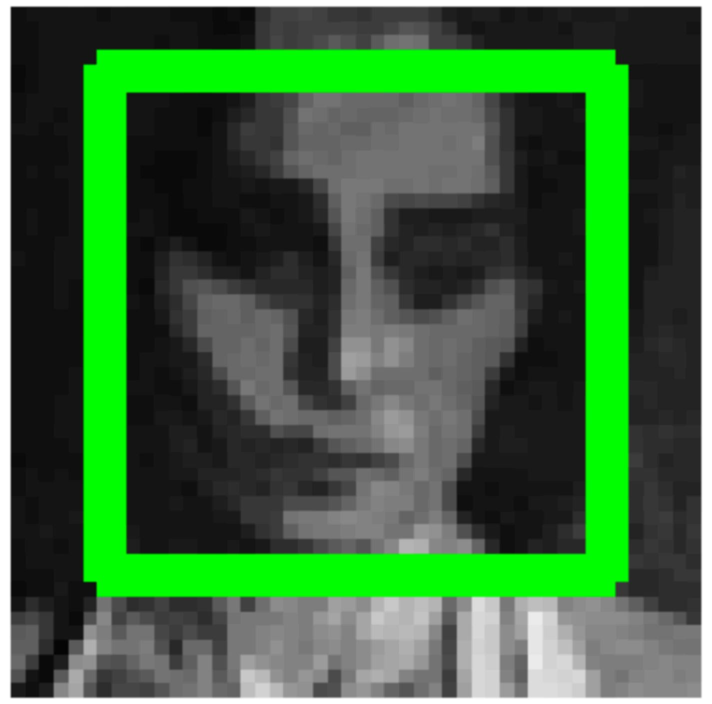
	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Prediction: Happy



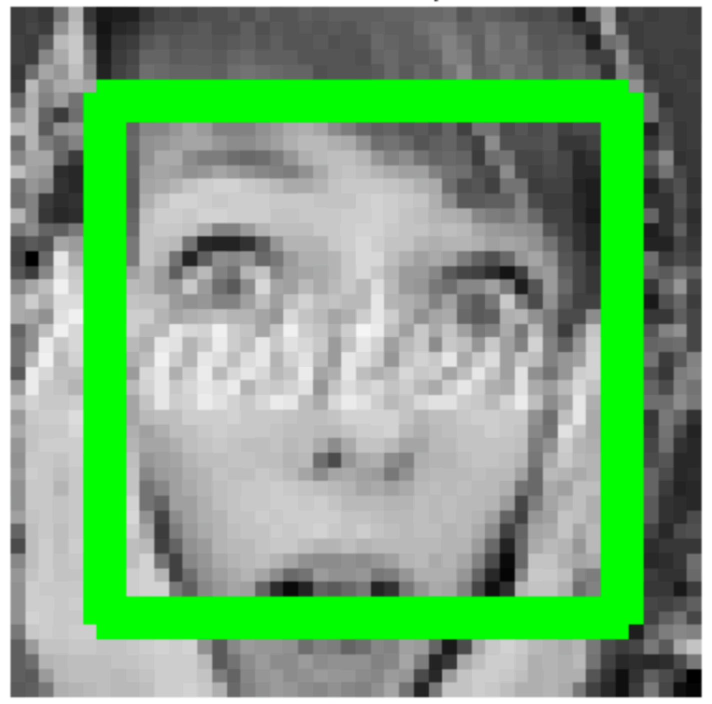
	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Prediction: Sad



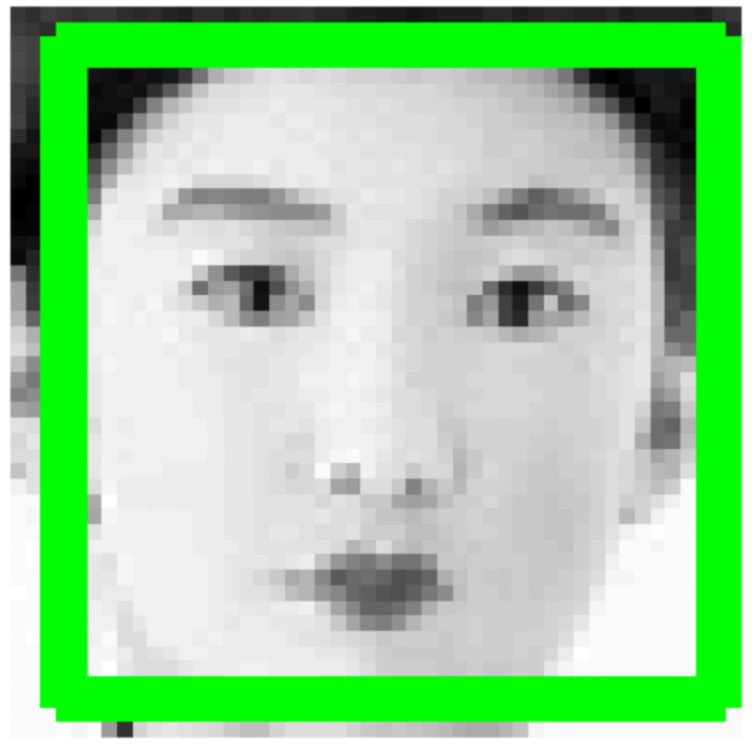
	name	artist	mood	popularity
0	Chop Suey!	System Of A Down	Energetic	79
1	Killing In The Name	Rage Against The Machine	Energetic	78
2	Dani California	Red Hot Chili Peppers	Energetic	77
3	Duality	Slipknot	Energetic	76
4	Uprising	Muse	Energetic	75

Prediction: Surprise



	name	artist	mood	popularity
0	Chop Suey!	System Of A Down	Energetic	79
1	Killing In The Name	Rage Against The Machine	Energetic	78
2	Dani California	Red Hot Chili Peppers	Energetic	77
3	Duality	Slipknot	Energetic	76
4	Uprising	Muse	Energetic	75

Prediction: Neutral



click to expand output; double click to hide output

	name	artist	mood	popularity
0	lovely (with Khalid)	Billie Eilish	Sad	88
1	you broke me first	Tate McRae	Sad	87
2	when the party's over	Billie Eilish	Sad	87
3	Falling	Harry Styles	Sad	87
4	Before You Go	Lewis Capaldi	Sad	85

Prediction: Disgust

	name	artist	mood	popularity
0	Lost	Annelie	Calm	64
1	Curiosity	Beau Projet	Calm	60
2	Escaping Time	Benjamin Martins	Calm	60
3	Just Look at You	369	Calm	59
4	Vague	Amaranth Cove	Calm	59

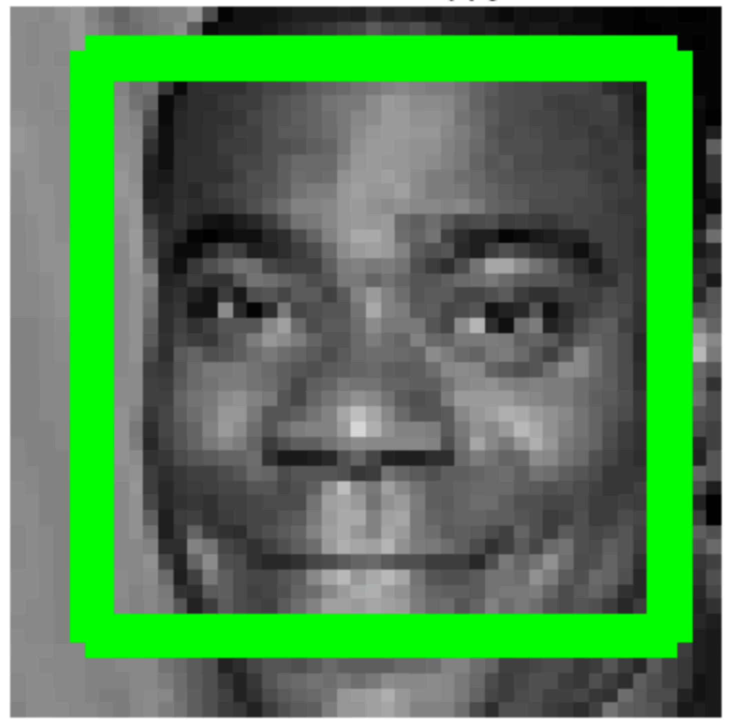
Prediction: Angry

	name	artist	mood	popularity
0	Lost	Annelie	Calm	64
1	Curiosity	Beau Projet	Calm	60
2	Escaping Time	Benjamin Martins	Calm	60
3	Just Look at You	369	Calm	59
4	Vague	Amaranth Cove	Calm	59

Prediction: Angry

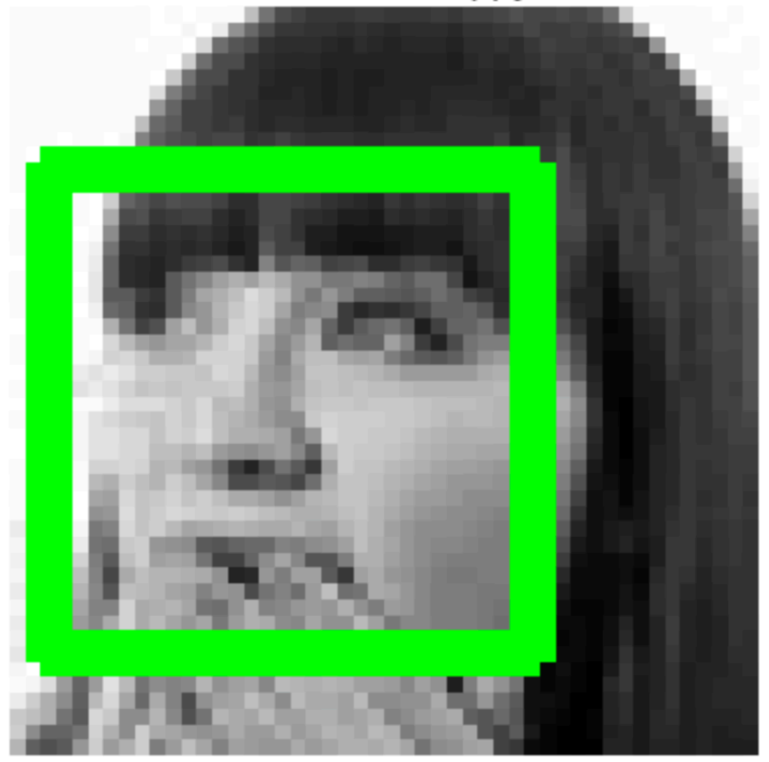
	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Prediction: Happy



	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Prediction: Happy



7f. DEFINING RESNET50-V2 MODEL

- ResNet50-V2 is a modified version of ResNet50 that performs better than ResNet50 and ResNet101 on the ImageNet dataset
- ResNet50-V2 modifications were made in the propagation formulation of the connections between blocks
- ResNet50-V2 also produces the same size of feature map on its final layer
- ResNet50 is a 50-layer convolutional neural network (48 convolutional layers, one maxpool layer, one average pool layer)
- Residual neural networks are a type of Artificial Neural Network that forms networks by stacking residual blocks

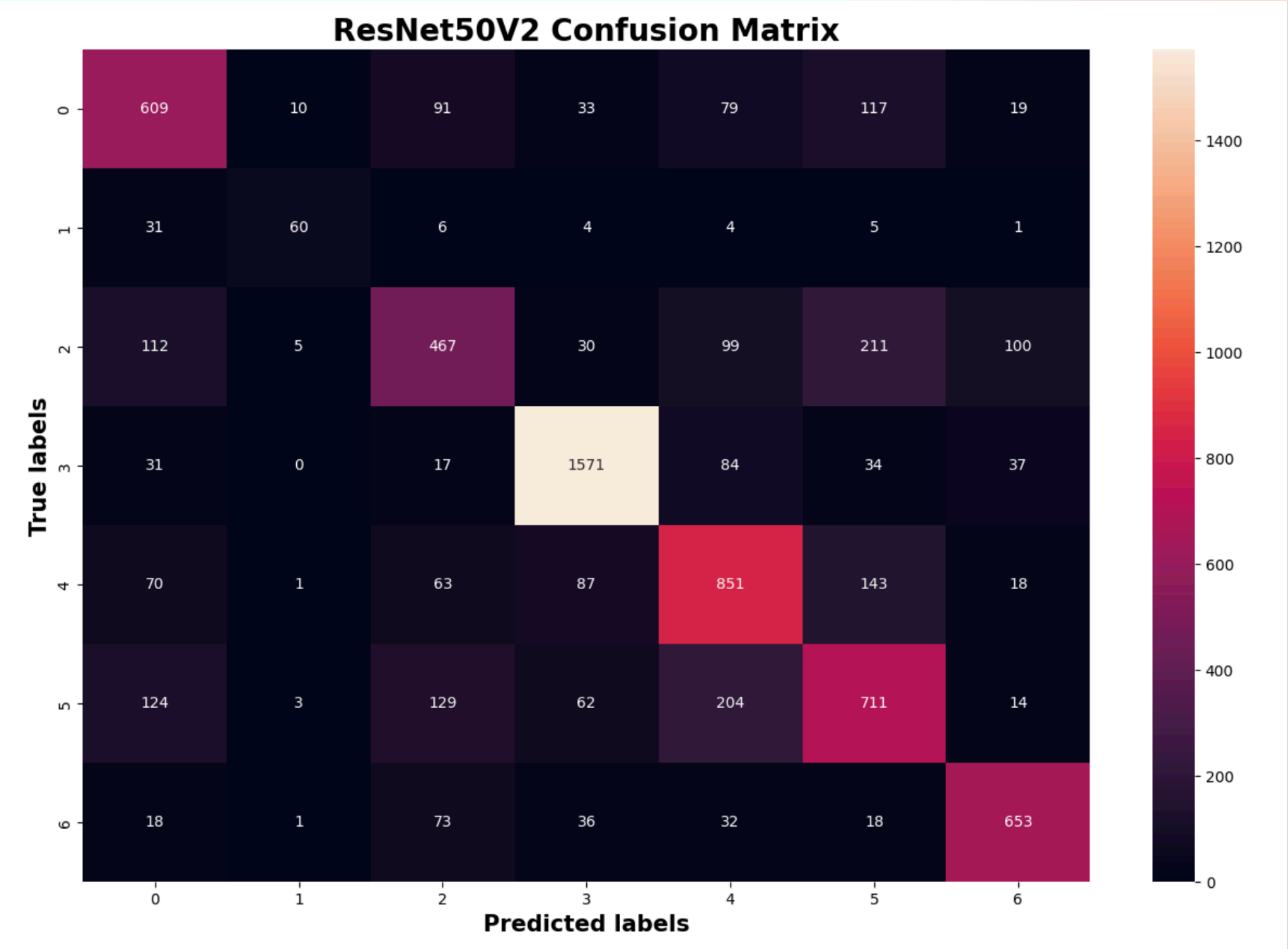
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50v2 (Functional)      (None, 7, 7, 2048)        23564800
_________________________________________________________________
dropout_9 (Dropout)          (None, 7, 7, 2048)        0
_________________________________________________________________
batch_normalization_12 (Batc (None, 7, 7, 2048)        8192
_________________________________________________________________
flatten_1 (Flatten)          (None, 100352)            0
_________________________________________________________________
dense_7 (Dense)              (None, 64)                6422592
_________________________________________________________________
batch_normalization_13 (Batc (None, 64)                256
_________________________________________________________________
dropout_10 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_8 (Dense)              (None, 7)                 455
=================================================================
Total params: 29,996,295
Trainable params: 22,779,527
Non-trainable params: 7,216,768
_________________________________________________________________
```

```
: ResNet50V2_history = ResNet50V2_Model.fit(train_data ,validation_data = test_data , epochs=50, batch_size=batch_size
                                          callbacks = callbacks, steps_per_epoch=steps_per_epoch, validation_steps=va

Epoch 1/30
448/448 [==============================] - 371s 817ms/step - loss: 1.5742 - accuracy: 0.4648 - val_loss: 1.5658 - v
al_accuracy: 0.4547
Epoch 2/30
448/448 [==============================] - 365s 814ms/step - loss: 1.2277 - accuracy: 0.5574 - val_loss: 1.1444 - v
al_accuracy: 0.5787
Epoch 3/30
448/448 [==============================] - 361s 806ms/step - loss: 1.1163 - accuracy: 0.5922 - val_loss: 1.0438 - v
al_accuracy: 0.6116
Epoch 4/30
448/448 [==============================] - 359s 800ms/step - loss: 1.0586 - accuracy: 0.6140 - val_loss: 1.0848 - v
al_accuracy: 0.6123
Epoch 5/30
448/448 [==============================] - 361s 806ms/step - loss: 1.0466 - accuracy: 0.6218 - val_loss: 1.0666 - v
al_accuracy: 0.6112

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
Epoch 6/30
448/448 [==============================] - 362s 809ms/step - loss: 0.9474 - accuracy: 0.6593 - val_loss: 0.9481 - v
al_accuracy: 0.6507
Epoch 7/30
448/448 [==============================] - 365s 814ms/step - loss: 0.8972 - accuracy: 0.6771 - val_loss: 0.9253 - v
al_accuracy: 0.6560
Epoch 8/30
448/448 [==============================] - 364s 813ms/step - loss: 0.8864 - accuracy: 0.6806 - val_loss: 0.9490 - v
al_accuracy: 0.6501
Epoch 9/30
448/448 [==============================] - 358s 800ms/step - loss: 0.8530 - accuracy: 0.6916 - val_loss: 0.9252 - v
al_accuracy: 0.6621
Epoch 10/30
448/448 [==============================] - 361s 806ms/step - loss: 0.8316 - accuracy: 0.7012 - val_loss: 0.9203 - v
al_accuracy: 0.6670
```

7g. EVALUATION OF ResNet50-V2 MODEL

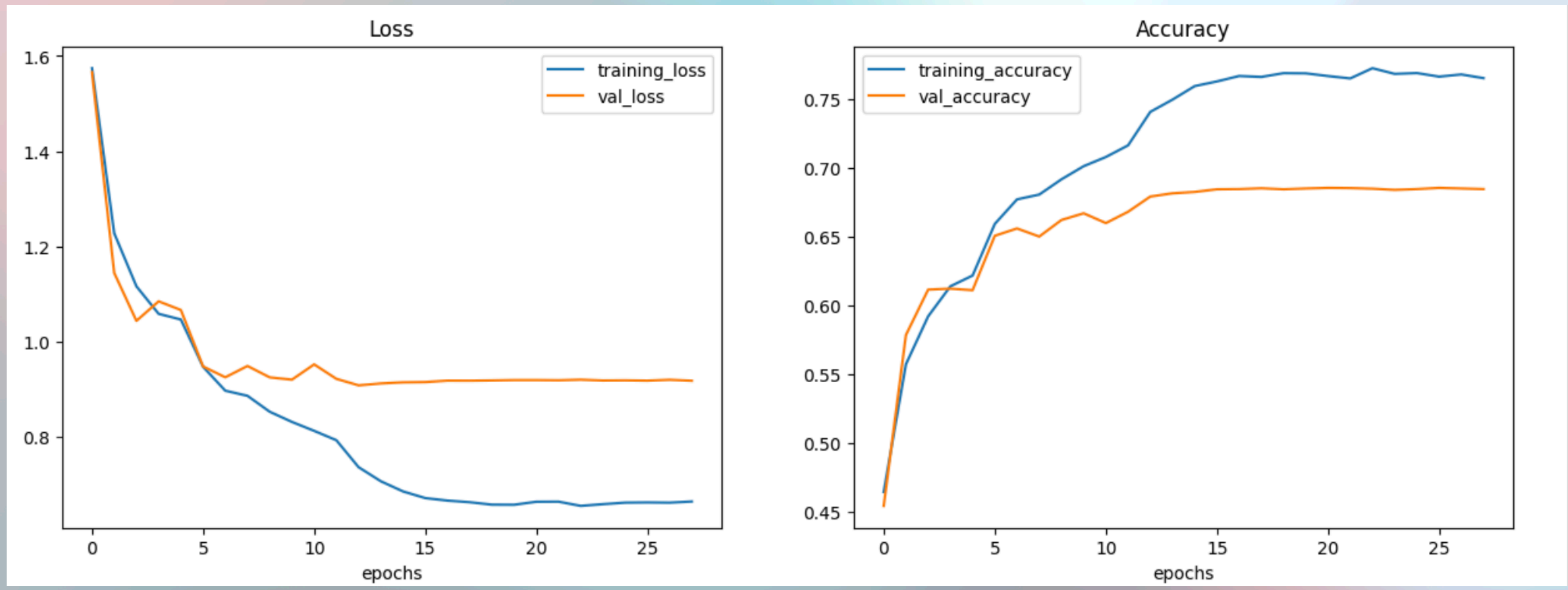
- Confusion matrix is a table used to define the preformation of classification algorithm
- Correlation matrix is simply a table which displays the correlation coefficients for different variables
- Accuracy curve is the one with both training and validation accuracy
- Gap between training and validation accuracies is a clear indication of overfitting
- Larger the gap higher the overfitting and smaller the gap lesser the overfitting

TEST ACCURACY OF RESNET50-V2 MODEL : 68.57%




```
ResNet50V2_Score = ResNet50V2_Model.evaluate(test_data)
print("Test Loss: {:.5f}".format(ResNet50V2_Score[0]))
print("Test Accuracy: {:.2f}%".format(ResNet50V2_Score[1] * 100))
```

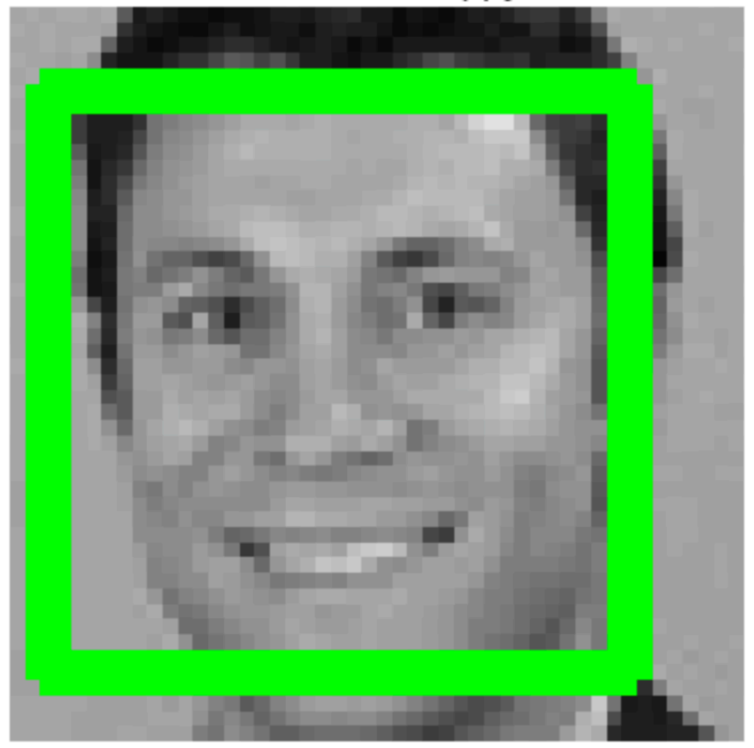
```
113/113 [==============================] - 17s 149ms/step - loss: 0.9187 - accuracy: 0.6857
    Test Loss: 0.91872
Test Accuracy: 68.57%
```



7h. PREDICTION BY ResNet50V2 MODEL

	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Prediction: Happy



	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

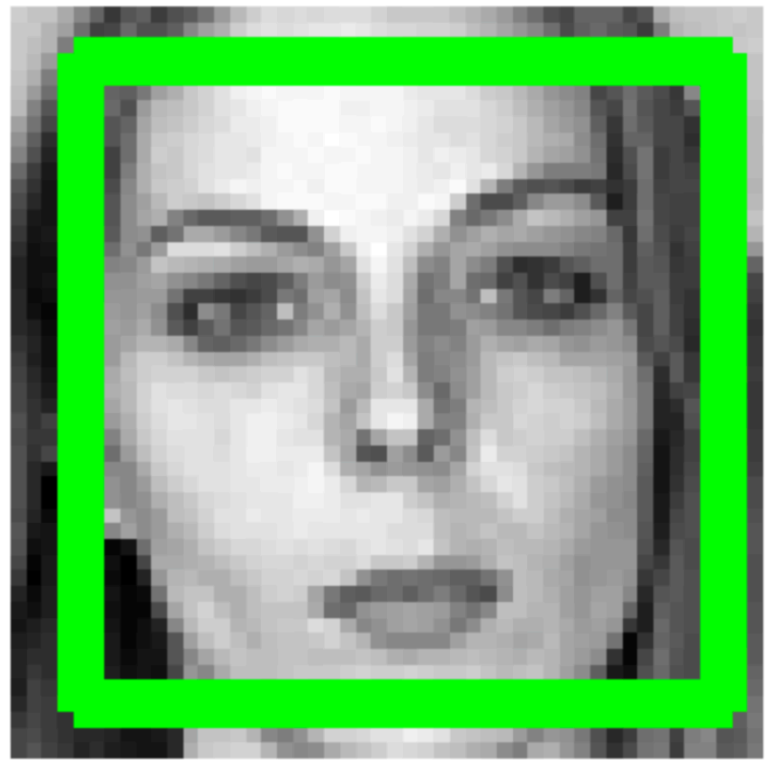
Prediction: Happy

	name	artist	mood	popularity
0	Lost	Annelie	Calm	64
1	Curiosity	Beau Projet	Calm	60
2	Escaping Time	Benjamin Martins	Calm	60
3	Just Look at You	369	Calm	59
4	Vague	Amaranth Cove	Calm	59

Prediction: Fear

	name	artist	mood	popularity
0	Chop Suey!	System Of A Down	Energetic	79
1	Killing In The Name	Rage Against The Machine	Energetic	78
2	Dani California	Red Hot Chili Peppers	Energetic	77
3	Duality	Slipknot	Energetic	76
4	Uprising	Muse	Energetic	75

Prediction: Neutral



	name	artist	mood	popularity
0	Chop Suey!	System Of A Down	Energetic	79
1	Killing In The Name	Rage Against The Machine	Energetic	78
2	Dani California	Red Hot Chili Peppers	Energetic	77
3	Duality	Slipknot	Energetic	76
4	Uprising	Muse	Energetic	75

Prediction: Surprise

	name	artist	mood	popularity
0	Lost	Annelie	Calm	64
1	Curiosity	Beau Projet	Calm	60
2	Escaping Time	Benjamin Martins	Calm	60
3	Just Look at You	369	Calm	59
4	Vague	Amaranth Cove	Calm	59

Prediction: Angry

	name	artist	mood	popularity
0	Lost	Annelie	Calm	64
1	Curiosity	Beau Projet	Calm	60
2	Escaping Time	Benjamin Martins	Calm	60
3	Just Look at You	369	Calm	59
4	Vague	Amaranth Cove	Calm	59

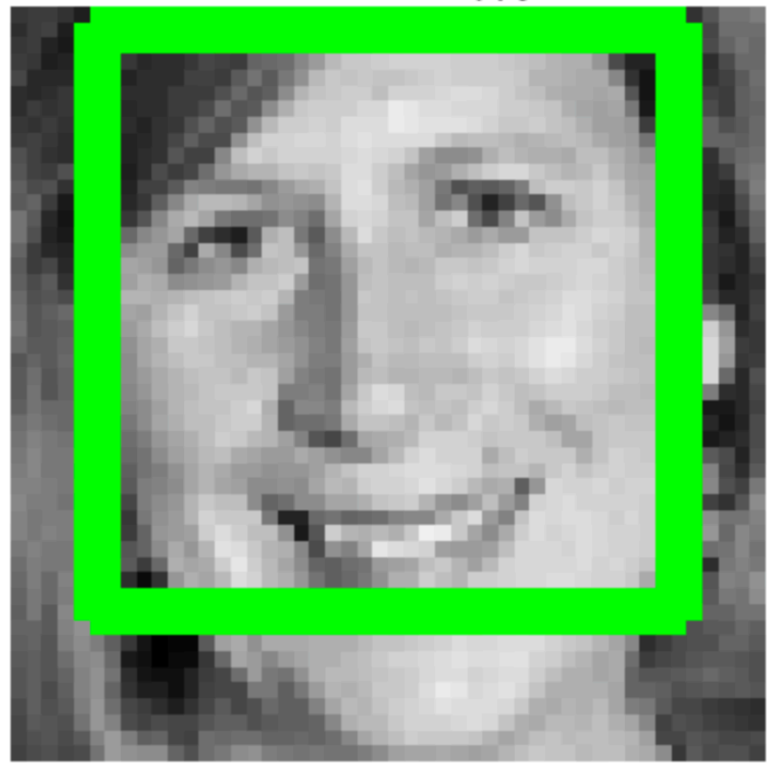
Prediction: Angry

	name	artist	mood	popularity
0	Lost	Annelie	Calm	64
1	Curiosity	Beau Projet	Calm	60
2	Escaping Time	Benjamin Martins	Calm	60
3	Just Look at You	369	Calm	59
4	Vague	Amaranth Cove	Calm	59

Prediction: Fear

	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Prediction: Happy



	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Prediction: Sad

8. CONCLUSION

CNN	64.80
ResNet50-V2	68.57

- In this project a music recommendation system based on two models was proposed based on facial emotions of target
- The human face is given as input, from which facial emotion is detected and based on the emotions music is suggested automatically
- Music are the one that has the power to heal any stress or any kind of emotions
- Recent development promises a wide scope in developing emotion based music recommendation system
- For the proposed project it was evident that ResNet50-V2 works better as the pre-trained model has 50 CNN layers inside the model which solves complex task and provides better accuracy

9. REFERENCES

- Abdat, F. et al. (2011). Human-Computer Interaction Using Emotion Recognition from Facial Expression. In: 2011 UKSim 5th European Symposium on Com- puter Modeling and Simulation. IEEE. doi: 10 . 1109/ems.2011.20.
- Barrett, Lisa Feldman et al. (2019). Emotional Expres- sions Reconsidered: Challenges to Inferring Emo- tion From Human Facial Movements. In: Psy- chological Science in the Public Interest 20.1. PMID: 31313636, pp. 1–68. doi: 10 . 1177 / 1529100619832930.
- Du, Shichuan et al. (2014). Compound facial expressions of emotion. In: Proceedings of the National Academy of Sciences 111.15, E1454–E1462. issn: 0027-8424. doi: 10.1073/ pnas.1322355111. eprint: https://www.pnas.org/content/111/15/E1454.full. pdf.
- Cowie, R. et al. (2001). Emotion recognition in human- computer interaction. In: IEEE Signal Processing Magazine 18.1, pp. 32–80. doi: 10.1109/79.911197.
- Jacintha, V et al. (2019). A Review on Facial Emotion Recognition Techniques. In: 2019 International Con- ference on Communication and Signal Processing (ICCSP). IEEE, pp. 0517–0521. doi: 10.1109/ICCSP. 2019.8698067.