Real Time Facial Expression Recognition

Description

Computer animated agents and robots bring new dimension in human computer interaction whicl computers can affect our social life in day-to-day activities. Face to face communication is a real- a time scale in the order of milliseconds. The level of uncertainty at this time scale is considerabl humans and machines to rely on sensory rich perceptual primitives rather than slow symbolic infe

In this project we are presenting the real time facial expression recognition of seven most basic h DISGUST, FEAR, HAPPY, NEUTRAL SAD, SURPRISE.

This model can be used for prediction of expressions of both still images and real time video. Hov have to provide image to the model. In case of real time video the image should be taken at any p the model for prediction of expression. The system automatically detects face using HAAR casca resize the image to a specific size and give it to the model for prediction. The model will generate corresponding to seven expressions. The highest probability value to the corresponding expressic expression for that image.

Business Problem

However, our goal here is to predict the human expressions, but we have trained our model on bot images. Since, we had only approx 1500 human images which are very less to make a good mode 9000 animated images and leverage those animated images for training the model and ultimately expressions on human images.

For better prediction we have decided to keep the size of each image **350∗350**.

For any image our goal is to predict the expression of the face in that image out of seven basic h

Problem Statement

CLASSIFY THE EXPRESSION OF FACE IN IMAGE OUT OF SEVEN BASIC HUMAN EXPRESSION

Performance Metric

This is a multi-class classification problem with 7 different classes, so we have considered three p

1. Multi-Class Log-loss
2. Accuracy
3. Confusion Metric

Source Data

We have downloaded data from 4 different sources.

1. Human Images Source-1: http://www.consortium.ri.cmu.edu/ckagree/
2. Human Images Source-2: http://app.visgraf.impa.br/database/faces/
3. Human Images Source-3: http://www.kasrl.org/jaffe.html
4. Animated Images Source: https://grail.cs.washington.edu/projects/deepexpr/ferg-db.html

Real-World Business Objective & Constraints

1. Low-latency is required.
2. Interpretability is important for still images but not in real time. For still images, probability c can be given.
3. Errors are not costly.

Y- Encoded Labels

Angry--1
Disgust --2
Fear--3
Happy--4
Neutral--5
Sad--6
Surprise--7

Mapping real-world to ML Problem

```
!git clone https://chethan1996:Chethan%409066@github.com/chethan1996/Dataset.gi
```

```
fatal: destination path 'Dataset' already exists and is not an empty director
```

```
!ls
```

```
BEFinalProject  Dataset  sample_data
```

```
import os
import numpy as np
import pandas as pd
```

```
import pandas as pd
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
from PIL import Image
import glob
import cv2
from sklearn.model_selection import train_test_split
from keras.layers import Dropout, Dense
from keras.layers.normalization import BatchNormalization
from keras.models import Sequential, load_model
from keras.applications import VGG16
from keras.layers import Dense, Conv2D, MaxPool2D , Flatten
from keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import accuracy_score, confusion_matrix
base_path="Dataset/Data/face-expression-dataset/images/images/"
print("All libraries imported successfully")
```

```
All libraries imported successfully
```

1. Reading the Data of Human Images

Angry

```
#human_angry1 = glob.glob('../Data/face-expression-recognition-dataset/images/i
human_angry = glob.glob(base_path+"train/angry/*")
#human_angry.remove('../Data/face-expression-recognition-dataset/images/images/
print("Number of images in Angry emotion = "+str(len(human_angry)))
#print(human_angry)
```

```
Number of images in Angry emotion = 3911
```

```
human_angry_folderName = [os.path.dirname(i)+"/" for i in human_angry]
human_angry_imageName = [os.path.basename(i) for i in human_angry]
human_angry_emotion = [["Angry"]*len(human_angry)][0]
human_angry_label = [1]*len(human_angry)
len(human_angry_folderName), len(human_angry_imageName), len(human_angry_emotio
```

```
(3911, 3911, 3911, 3911)
```

```
df_angry = pd.DataFrame()
p1=df_angry["folderName"] = human_angry_folderName
p2=df_angry["imageName"] = human_angry_imageName
df_angry["Emotion"] = human_angry_emotion
df_angry["Labels"] = human_angry_label
#print(p1)
#print(p2)
df_angry.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	31931.jpg	Angry	1
1	Dataset/Data/face-expression-dataset/images/im...	19243.jpg	Angry	1
2	Dataset/Data/face-expression-dataset/images/im...	30886.jpg	Angry	1
3	Dataset/Data/face-expression-dataset/images/im...	31213.jpg	Angry	1
4	Dataset/Data/face-expression-dataset/images/im...	35808.jpg	Angry	1

Disgust

```
human_disgust = glob.glob(base_path+"train/disgust/*")
print("Number of images in Disgust emotion = "+str(len(human_disgust)))
```

```
Number of images in Disgust emotion = 431
```

```
human_disgust_folderName = [os.path.dirname(i)+"/" for i in human_disgust]
human_disgust_imageName = [os.path.basename(i) for i in human_disgust]
human_disgust_emotion = [["Disgust"]*len(human_disgust)][0]
human_disgust_label = [2]*len(human_disgust)

len(human_disgust_folderName), len(human_disgust_imageName), len(human_disgust_
```

```
(431, 431, 431, 431)
```

```
df_disgust = pd.DataFrame()
#print(human_disgust_folderName)
df_disgust["folderName"] = human_disgust_folderName
df_disgust["imageName"] = human_disgust_imageName
df_disgust["Emotion"] = human_disgust_emotion
df_disgust["Labels"] = human_disgust_label
df_disgust.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	20703.jpg	Disgust	2
1	Dataset/Data/face-expression-dataset/images/im...	5964.jpg	Disgust	2
2	Dataset/Data/face-expression-dataset/images/im...	10236.jpg	Disgust	2
3	Dataset/Data/face-expression-dataset/images/im...	34622.jpg	Disgust	2
4	Dataset/Data/face-expression-dataset/images/im...	23683.jpg	Disgust	2

Fear

```
human_fear = glob.glob(base_path+"train/fear/*")
print("Number of images in Fear emotion = "+str(len(human_fear)))
```

```
Number of images in Fear emotion = 4061
```

```
human_fear_folderName = [os.path.dirname(i)+"/" for i in human_fear]
human_fear_imageName = [os.path.basename(i)  for i in human_fear]
human_fear_emotion = [["Fear"]*len(human_fear)][0]
human_fear_label = [3]*len(human_fear)

len(human_fear_folderName), len(human_fear_imageName), len(human_fear_emotion),
```

```
(4061, 4061, 4061, 4061)
```

```
df_fear = pd.DataFrame()
df_fear["folderName"] = human_fear_folderName
df_fear["imageName"] = human_fear_imageName
df_fear["Emotion"] = human_fear_emotion
df_fear["Labels"] = human_fear_label
df_fear.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	9137.jpg	Fear	3
1	Dataset/Data/face-expression-dataset/images/im...	29659.jpg	Fear	3
2	Dataset/Data/face-expression-dataset/images/im...	5551.jpg	Fear	3
3	Dataset/Data/face-expression-dataset/images/im...	22908.jpg	Fear	3
4	Dataset/Data/face-expression-dataset/images/im...	15386.jpg	Fear	3

Happy

```
human_happy = glob.glob(base_path+'train/happy/*')
print("Number of images in Happy emotion = "+str(len(human_happy)))
```

```
Number of images in Happy emotion = 6921
```

```
human_happy_folderName = [os.path.dirname(i)+"/" for i in human_happy]
human_happy_imageName = [os.path.basename(i)  for i in human_happy]
human_happy_emotion = [["Happy"]*len(human_happy)][0]
human_happy_label = [4]*len(human_happy)

len(human_happy_folderName), len(human_happy_imageName), len(human_happy_emotio
```

```
(6921, 6921, 6921, 6921)
```

```
df_happy = pd.DataFrame()
df_happy["folderName"] = human_happy_folderName
df_happy["imageName"] = human_happy_imageName
df_happy["Emotion"] = human_happy_emotion
df_happy["Labels"] = human_happy_label
df_happy.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	30186.jpg	Happy	4
1	Dataset/Data/face-expression-dataset/images/im...	223.jpg	Happy	4
2	Dataset/Data/face-expression-dataset/images/im...	8180.jpg	Happy	4
3	Dataset/Data/face-expression-dataset/images/im...	34027.jpg	Happy	4
4	Dataset/Data/face-expression-dataset/images/im...	30372.jpg	Happy	4

Neutral

```
human_neutral = glob.glob(base_path+'train/neutral/*')
print("Number of images in Neutral emotion = "+str(len(human_neutral)))
```

```
Number of images in Neutral emotion = 4869
```

```
human_neutral_folderName = [os.path.dirname(i)+"/" for i in human_neutral]
human_neutral_imageName = [os.path.basename(i)  for i in human_neutral]
human_neutral_emotion = [["Neutral"]*len(human_neutral)][0]
human_neutral_label = [5]*len(human_neutral)

len(human_neutral_folderName), len(human_neutral_imageName), len(human_neutral_
```

```
(4869, 4869, 4869, 4869)
```

```
df_neutral = pd.DataFrame()
df_neutral["folderName"] = human_neutral_folderName
df_neutral["imageName"] = human_neutral_imageName
df_neutral["Emotion"] = human_neutral_emotion
df_neutral["Labels"] = human_neutral_label
df_neutral.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	25499.jpg	Neutral	5
1	Dataset/Data/face-expression-dataset/images/im...	15826.jpg	Neutral	5
2	Dataset/Data/face-expression-dataset/images/im...	15485.jpg	Neutral	5
3	Dataset/Data/face-expression-dataset/images/im...	9427.jpg	Neutral	5
4	Dataset/Data/face-expression-dataset/images/im...	6234.jpg	Neutral	5

Sad

```
human_sad = glob.glob(base_path+'train/sad/*')
print("Number of images in Sad emotion = "+str(len(human_sad)))
```

```
Number of images in Sad emotion = 4897
```

```
human_sad_folderName = [os.path.dirname(i)+"/" for i in human_sad]
human_sad_imageName = [os.path.basename(i)  for i in human_sad]
human_sad_emotion = [["Sad"]*len(human_sad)][0]
human_sad_label = [6]*len(human_sad)

len(human_sad_folderName), len(human_sad_imageName), len(human_sad_emotion), le
```

```
(4897, 4897, 4897, 4897)
```

```
df_sad = pd.DataFrame()
df_sad["folderName"] = human_sad_folderName
df_sad["imageName"] = human_sad_imageName
df_sad["Emotion"] = human_sad_emotion
df_sad["Labels"] = human_sad_label
df_sad.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	7477.jpg	Sad	6
1	Dataset/Data/face-expression-dataset/images/im...	9282.jpg	Sad	6
2	Dataset/Data/face-expression-dataset/images/im...	28129.jpg	Sad	6
3	Dataset/Data/face-expression-dataset/images/im...	3342.jpg	Sad	6
4	Dataset/Data/face-expression-dataset/images/im...	31281.jpg	Sad	6

Surprise

```
human_surprise = glob.glob(base_path+'train/surprise/*')
#human_surprise.remove('../Data/Human/Surprise\\Thumbs.db')
print("Number of images in Surprise emotion = "+str(len(human_surprise)))
```

```
Number of images in Surprise emotion = 3173
```

```
human_surprise_folderName = [os.path.dirname(i) +"/" for i in human_surprise]
human_surprise_imageName = [os.path.basename(i)  for i in human_surprise]
human_surprise_emotion = [["Surprise"]*len(human_surprise)][0]
human_surprise_label = [7]*len(human_surprise)

len(human_surprise_folderName), len(human_surprise_imageName), len(human_surpri
```

```
(3173, 3173, 3173, 3173)
```

```
df_surprise = pd.DataFrame()
df_surprise["folderName"] = human_surprise_folderName
df_surprise["imageName"] = human_surprise_imageName
df_surprise["Emotion"] = human_surprise_emotion
df_surprise["Labels"] = human_surprise_label
```

```
df_surprise.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	15545.jpg	Surprise	7
1	Dataset/Data/face-expression-dataset/images/im...	35445.jpg	Surprise	7
2	Dataset/Data/face-expression-dataset/images/im...	14921.jpg	Surprise	7
3	Dataset/Data/face-expression-dataset/images/im...	18420.jpg	Surprise	7
4	Dataset/Data/face-expression-dataset/images/im...	22575.jpg	Surprise	7

```
length = df_angry.shape[0] + df_disgust.shape[0] + df_fear.shape[0] + df_happy.
print("Total number of images in all the emotions = "+str(length))
```

```
Total number of images in all the emotions = 28263
```

Concatenating all dataframes

```
frames = [df_angry, df_disgust, df_fear, df_happy, df_neutral, df_sad, df_surpr
Final_human = pd.concat(frames)
Final_human.shape
```

```
(28263, 4)
```

```
Final_human.reset_index(inplace = True, drop = True)
Final_human = Final_human.sample(frac = 1.0)   #shuffling the dataframe
Final_human.reset_index(inplace = True, drop = True)
Final_human.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	14746.jpg	Happy	4
1	Dataset/Data/face-expression-dataset/images/im...	13685.jpg	Fear	3
2	Dataset/Data/face-expression-dataset/images/im...	8643.jpg	Neutral	5
3	Dataset/Data/face-expression-dataset/images/im...	25523.jpg	Happy	4
4	Dataset/Data/face-expression-dataset/images/im...	17806.jpg	Surprise	7

2. Train, CV and Test Split for Human Images

```
df_human_train_data, df_human_test = train_test_split(Final_human, stratify=Fin
df_human_train, df_human_cv = train_test_split(df_human_train_data, stratify=df
df_human_train.shape, df_human_cv.shape, df_human_test.shape
```

```
((18891, 4), (3779, 4), (5593, 4))
```

```
df_human_train.reset_index(inplace = True, drop = True)
df_human_train.to_pickle(base_path+"train.pkl")

df_human_test.reset_index(inplace = True, drop = True)
df_human_test.to_pickle(base_path+"validation.pkl")
```

```
df_human_train = pd.read_pickle(base_path+"train.pkl")
df_human_train.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	28446.jpg	Happy	4
1	Dataset/Data/face-expression-dataset/images/im...	34684.jpg	Sad	6
2	Dataset/Data/face-expression-dataset/images/im...	19320.jpg	Happy	4
3	Dataset/Data/face-expression-dataset/images/im...	1187.jpg	Happy	4
4	Dataset/Data/face-expression-dataset/images/im...	32770.jpg	Sad	6

```
df_human_train.shape
```

```
(18891, 4)
```

```
df_human_test = pd.read_pickle(base_path+"validation.pkl")
df_human_test.head()
#print(df_human_test["folderName"][1])
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	3966.jpg	Fear	3
1	Dataset/Data/face-expression-dataset/images/im...	31581.jpg	Angry	1
2	Dataset/Data/face-expression-dataset/images/im...	4117.jpg	Happy	4
3	Dataset/Data/face-expression-dataset/images/im...	11839.jpg	Sad	6
4	Dataset/Data/face-expression-dataset/images/im...	3642.jpg	Happy	4

```
df_human_test.shape
```

```
(5593, 4)
```

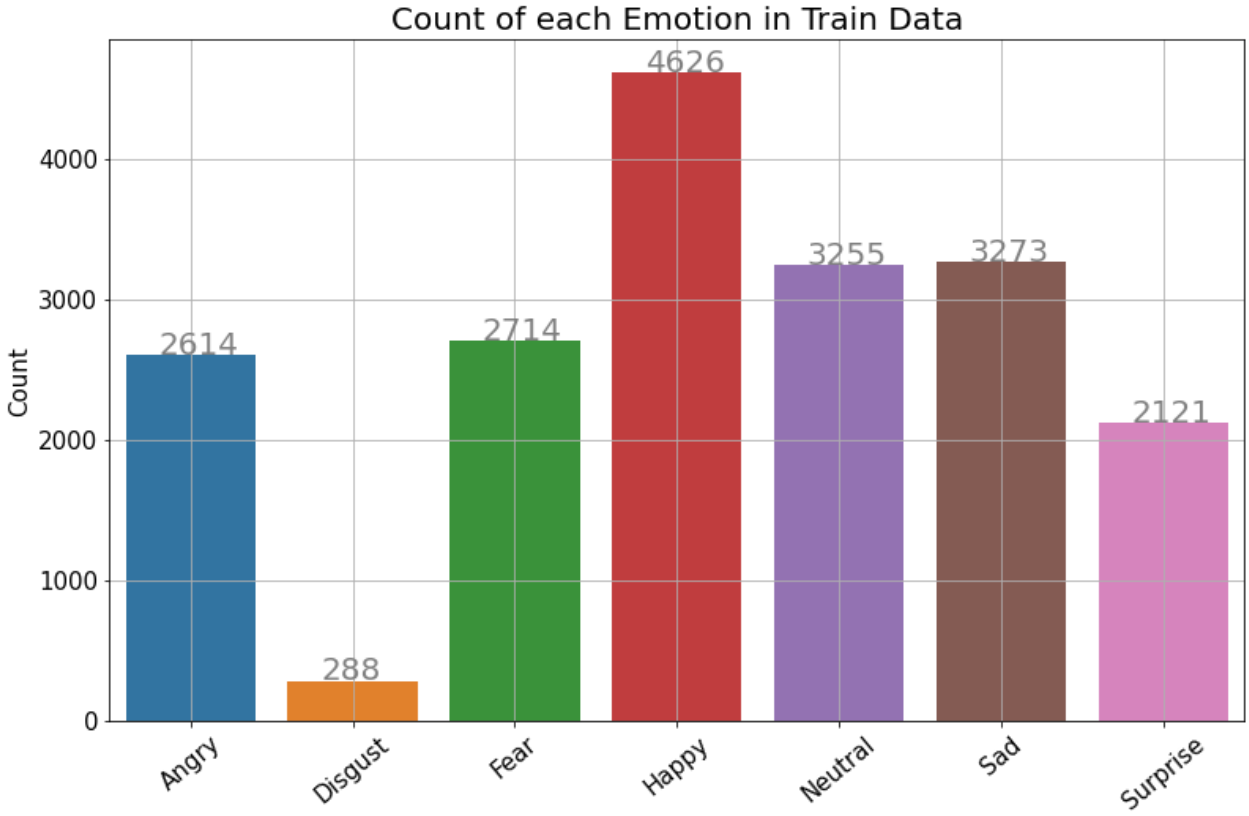
3. Analysing Data of Human Images

Distribution of class labels in Train and Test

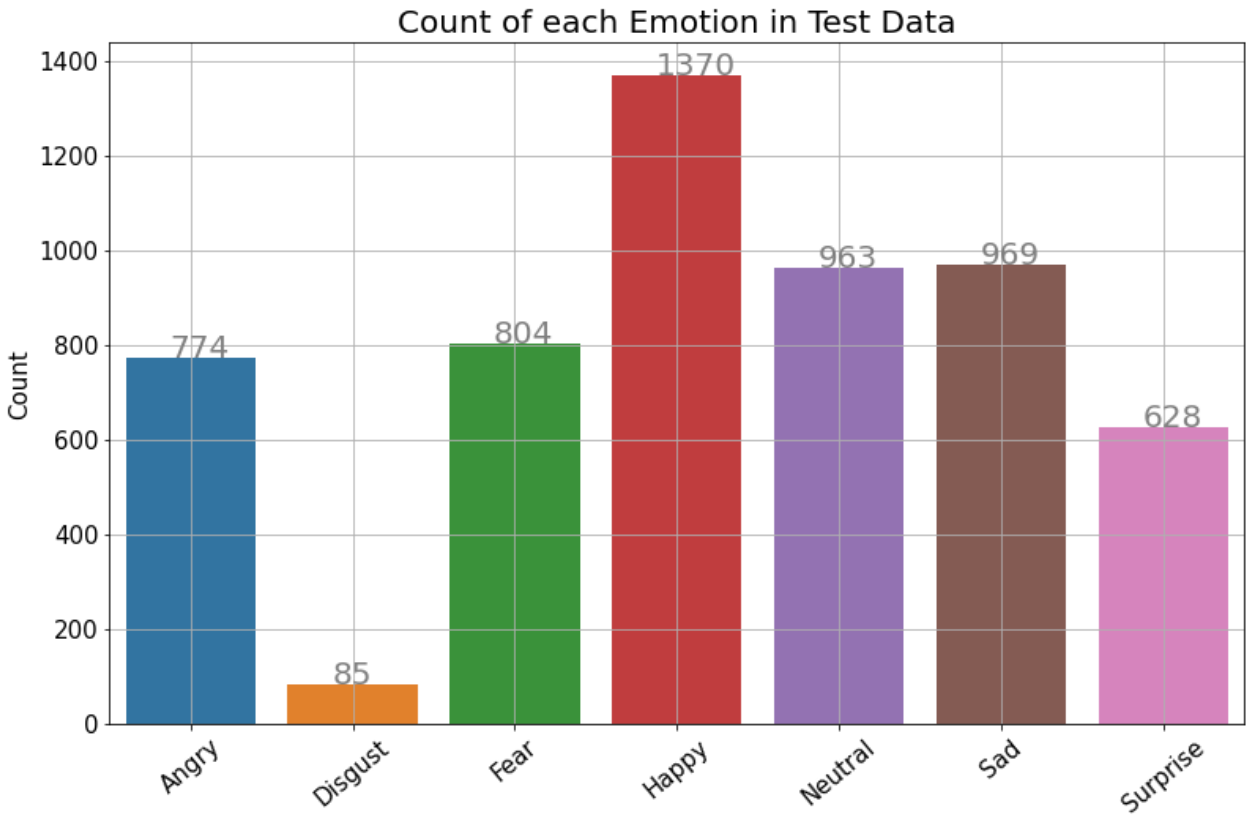
```
df_temp_train = df_human_train.sort_values(by = "Labels", inplace = False)
df_temp_test = df_human_test.sort_values(by = "Labels", inplace = False)

TrainData_distribution = df_human_train["Emotion"].value_counts().sort_index()
TestData_distribution = df_human_test["Emotion"].value_counts().sort_index()
```

```

TrainData_distribution_sorted = sorted(TrainData_distribution.items(), key = la
TestData_distribution_sorted = sorted(TestData_distribution.items(), key = lamb
```

```
fig = plt.figure(figsize = (10, 6))
ax = fig.add_axes([0,0,1,1])
ax.set_title("Count of each Emotion in Train Data", fontsize = 20)
sns.countplot(x = "Emotion", data = df_temp_train)
plt.grid()
for i in ax.patches:
    ax.text(x = i.get_x() + 0.2, y = i.get_height()+1.5, s = str(i.get_height()
plt.xlabel("")
plt.ylabel("Count", fontsize = 15)
plt.tick_params(labelsize = 15)
plt.xticks(rotation = 40)
plt.show()

for i in TrainData_distribution_sorted:
    print("Number of training data points in class "+str(i[0])+" = "+str(i[1])+

print("-"*80)


fig = plt.figure(figsize = (10, 6))
ax = fig.add_axes([0,0,1,1])
ax.set_title("Count of each Emotion in Test Data", fontsize = 20)
sns.countplot(x = "Emotion", data = df_temp_test)
plt.grid()
for i in ax.patches:
    ax.text(x = i.get_x() + 0.27, y = i.get_height()+0.2, s = str(i.get_height(
plt.xlabel("")
plt.ylabel("Count", fontsize = 15)
plt.tick_params(labelsize = 15)
plt.xticks(rotation = 40)
plt.show()

for i in TestData_distribution_sorted:
    print("Number of training data points in class "+str(i[0])+" = "+str(i[1])+
```




```
Number of training data points in class Happy = 4626(24.4879%)
Number of training data points in class Sad = 3273(17.3257%)
Number of training data points in class Neutral = 3255(17.2304%)
Number of training data points in class Fear = 2714(14.3666%)
Number of training data points in class Angry = 2614(13.8373%)
Number of training data points in class Surprise = 2121(11.2276%)
Number of training data points in class Disgust = 288(1.5245%)
--------------------------------------------------------------------------------
```




```
Number of training data points in class Happy = 1370(24.4949%)
Number of training data points in class Sad = 969(17.3252%)
Number of training data points in class Neutral = 963(17.218%)
Number of training data points in class Fear = 804(14.3751%)
Number of training data points in class Angry = 774(13.8387%)
```

```
Number of training data points in class Angry = 774(13.8587%)
Number of training data points in class Surprise = 628(11.2283%)
Number of training data points in class Disgust = 85(1.5198%)
```

4. Pre-Processing Human Images

4.1 Converting all the images to grayscale and save them

```
def convt_to_gray(df):
    count = 0
    for i in range(len(df)):
        path1 = df["folderName"][i]
        path2 = df["imageName"][i]
        #print(os.path.join(path1, path2))
        img = cv2.imread(os.path.join(path1, path2))
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        cv2.imwrite(os.path.join(path1, path2), gray)
        count += 1
    print("Total number of images converted and saved = "+str(count))
```

```
convt_to_gray(df_human_train)
```

```
Total number of images converted and saved = 18891
```

```
convt_to_gray(df_human_test)
```

```
Total number of images converted and saved = 5593
```

4.2 Detecting face in image using HAAR then crop it then resize then save the

```
!git clone https://github.com/chethan1996/BEFinalProject.git
```

```
fatal: destination path 'BEFinalProject' already exists and is not an empty d
```

```
#detect the face in image using HAAR cascade then crop it then resize it and fi
face_cascade = cv2.CascadeClassifier('BEFinalProject/haarcascade_frontalface_de
```

```
#download this xml file from link: https://github.com/opencv/opencv/tree/master
def face_det_crop_resize(img_path):
    img = cv2.imread(img_path)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:
        face_clip = img[y:y+h, x:x+w]  #cropping the face in image
        cv2.imwrite(img_path, cv2.resize(face_clip, (48, 48)))  #resizing image
```

```
for i, d in df_human_train.iterrows():
    img_path = os.path.join(d["folderName"], d["imageName"])
    face_det_crop_resize(img_path)
```

```
for i, d in df_human_cv.iterrows():
    img_path = os.path.join(d["folderName"], d["imageName"])
    face_det_crop_resize(img_path)
```

```
for i, d in df_human_test.iterrows():
    img_path = os.path.join(d["folderName"], d["imageName"])
    face_det_crop_resize(img_path)
```

10. Creating bottleneck features from VGG-16 model. Here, we are u learning.

```
Train_Data = pd.read_pickle(base_path+"train.pkl")
Test_Data = pd.read_pickle(base_path+"validation.pkl")
Train_Data.shape, Test_Data.shape
```

```
((18891, 4), (5593, 4))
```

```
Train_Data.head()
```

	folderName	imageName	Emotion	Labels
0	Dataset/Data/face-expression-dataset/images/im...	28446.jpg	Happy	4
1	Dataset/Data/face-expression-dataset/images/im...	34684.jpg	Sad	6
2	Dataset/Data/face-expression-dataset/images/im...	19320.jpg	Happy	4
3	Dataset/Data/face-expression-dataset/images/im...	1187.jpg	Happy	4
4	Dataset/Data/face-expression-dataset/images/im...	32770.jpg	Sad	6

```
```

```
TrainData_batch_pointer = 0
TestData_batch_pointer = 0
```

10.1 Bottleneck features for Train Data

```
TrainData_Labels = pd.get_dummies(Train_Data["Labels"]).as_matrix()
TrainData_Labels.shape
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: FutureWarning
  """Entry point for launching an IPython kernel.
(18891, 7)
```

```
trdata = ImageDataGenerator()
traindata = trdata.flow_from_directory(directory=base_path+"/train",target_size
tsdata = ImageDataGenerator()
testTestdata = tsdata.flow_from_directory(directory=base_path+"/validation", ta
```

```
Found 28263 images belonging to 7 classes.
Found 7066 images belonging to 7 classes.
```

```
Train_Data = Train_Data.dropna(how='any',axis=0)
print(int(len(Train_Data)))
```

```
18892
```

```
batch_images = []
batch_labels = []
for i in range(len(Test_Data)):
    #print(i)
    path1 = Test_Data.iloc[i]["folderName"]
    path2 = Test_Data.iloc[ i]["imageName"]
    read_image = cv2.imread(os.path.join(path1, path2))
    #read_image_final = read_image/255.0  #here, we are normalizing the images
    try:
        if(read_image.shape!=(48,48,3)):
            print("Removed {}, {}".format(os.path.join(path1, path2), read_imag
            os.remove(os.path.join(path1, path2))
        else:
            print(read_image.shape)
    except AttributeError :
        print(read_image)
        print("gotti")
        os.remove(os.path.join(path1, path2))
```

```
None
gotti
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-220-0c668b4ea0ca> in <module>()
      9     try:
---> 10         if(read_image.shape!=(48,48,3)):
     11             print("Removed {}, {}".format(os.path.join(path1, path2),

AttributeError: 'NoneType' object has no attribute 'shape'

During handling of the above exception, another exception occurred:
```

```
TestData_batch_pointer=0
print(TrainData_batch_pointer)
print(TestData_batch_pointer)
```

```
18890
0
```

SEARCH STACK OVERFLOW

```
def loadTrainBatch(batch_size):
    global TrainData_batch_pointer
    batch_images = []
    batch_labels = []
    for i in range(batch_size):
        path1 = Train_Data.iloc[TrainData_batch_pointer + i]["folderName"]
        path2 = Train_Data.iloc[TrainData_batch_pointer + i]["imageName"]
        read_image = cv2.imread(os.path.join(path1, path2))
        #read_image_final = read_image/255.0  #here, we are normalizing the ima
        #print(read_image_final)
        batch_images.append(read_image)
        try:
            print(read_image.shape)
        except AttributeError :
            print("gotti")
            print(read_image)
            print(os.path.join(path1, path2))
            os.remove(os.path.join(path1, path2))
        batch_labels.append(TrainData_Labels[TrainData_batch_pointer + i]) #app

    TrainData_batch_pointer += batch_size

    return np.array(batch_images), np.array(batch_labels)
```

```
#creating bottleneck features for train data using VGG-16- Image-net model
model = VGG16(weights='imagenet', include_top=False)
SAVEDIR = "Dataset/Data/Bottleneck_Features/Bottleneck_TrainData/"
SAVEDIR_LABELS = "Dataset/Data/Bottleneck_Features/TrainData_Labels/"
batch_size =10
for i in range(int(len(Train_Data)/batch_size)):
    x, y = loadTrainBatch(batch_size)
    print("Batch {} loaded".format(i+1))

    np.save(os.path.join(SAVEDIR_LABELS, "bottleneck_labels_{}".format(i+1)), y
```

```

print("Creating bottleneck features for batch {}". format(i+1))
bottleneck_features = model.predict(x)

np.save(os.path.join(SAVEDIR, "bottleneck_{}".format(i+1)), bottleneck_feat
print("Bottleneck features for batch {} created and saved\n".format(i+1))
```

Streaming output truncated to the last 5000 lines.

```
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1533 loaded
Creating bottleneck features for batch 1533
Bottleneck features for batch 1533 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1534 loaded
Creating bottleneck features for batch 1534
Bottleneck features for batch 1534 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1535 loaded
Creating bottleneck features for batch 1535
Bottleneck features for batch 1535 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1536 loaded
Creating bottleneck features for batch 1536
Bottleneck features for batch 1536 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
```

```
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1537 loaded
Creating bottleneck features for batch 1537
Bottleneck features for batch 1537 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1538 loaded
Creating bottleneck features for batch 1538
Bottleneck features for batch 1538 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1539 loaded
Creating bottleneck features for batch 1539
Bottleneck features for batch 1539 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1540 loaded
Creating bottleneck features for batch 1540
Bottleneck features for batch 1540 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
```

```
(48, 48, 3)
(48, 48, 3)
Batch 1541 loaded
Creating bottleneck features for batch 1541
Bottleneck features for batch 1541 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1542 loaded
Creating bottleneck features for batch 1542
Bottleneck features for batch 1542 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1543 loaded
Creating bottleneck features for batch 1543
Bottleneck features for batch 1543 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1544 loaded
Creating bottleneck features for batch 1544
Bottleneck features for batch 1544 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1545 loaded
Creating bottleneck features for batch 1545
Bottleneck features for batch 1545 created and saved
```

```
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1546 loaded
Creating bottleneck features for batch 1546
Bottleneck features for batch 1546 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1547 loaded
Creating bottleneck features for batch 1547
Bottleneck features for batch 1547 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1548 loaded
Creating bottleneck features for batch 1548
Bottleneck features for batch 1548 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1549 loaded
Creating bottleneck features for batch 1549
Bottleneck features for batch 1549 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
```

```
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1550 loaded
Creating bottleneck features for batch 1550
Bottleneck features for batch 1550 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1551 loaded
Creating bottleneck features for batch 1551
Bottleneck features for batch 1551 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1552 loaded
Creating bottleneck features for batch 1552
Bottleneck features for batch 1552 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1553 loaded
Creating bottleneck features for batch 1553
Bottleneck features for batch 1553 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1554 loaded
```

```
Creating bottleneck features for batch 1554
Bottleneck features for batch 1554 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1555 loaded
Creating bottleneck features for batch 1555
Bottleneck features for batch 1555 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1556 loaded
Creating bottleneck features for batch 1556
Bottleneck features for batch 1556 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1557 loaded
Creating bottleneck features for batch 1557
Bottleneck features for batch 1557 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1558 loaded
Creating bottleneck features for batch 1558
Bottleneck features for batch 1558 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
```

```
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1559 loaded
Creating bottleneck features for batch 1559
Bottleneck features for batch 1559 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1560 loaded
Creating bottleneck features for batch 1560
Bottleneck features for batch 1560 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1561 loaded
Creating bottleneck features for batch 1561
Bottleneck features for batch 1561 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1562 loaded
Creating bottleneck features for batch 1562
Bottleneck features for batch 1562 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
```

```
(48, 48, 3)
(48, 48, 3)
Batch 1563 loaded
Creating bottleneck features for batch 1563
Bottleneck features for batch 1563 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1564 loaded
Creating bottleneck features for batch 1564
Bottleneck features for batch 1564 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1565 loaded
Creating bottleneck features for batch 1565
Bottleneck features for batch 1565 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1566 loaded
Creating bottleneck features for batch 1566
Bottleneck features for batch 1566 created and saved

(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
(48, 48, 3)
Batch 1567 loaded
Creating bottleneck features for batch 1567
Bottleneck features for batch 1567 created and saved
```