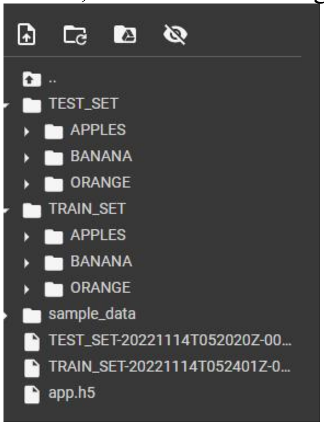
IMAGE PROCESSING

Date	13 NOVEMBER 2022
Team ID	PNT2022TMID48373
Project Name	AI-Powered Nutrition Analyzer for fitness enthusiasts

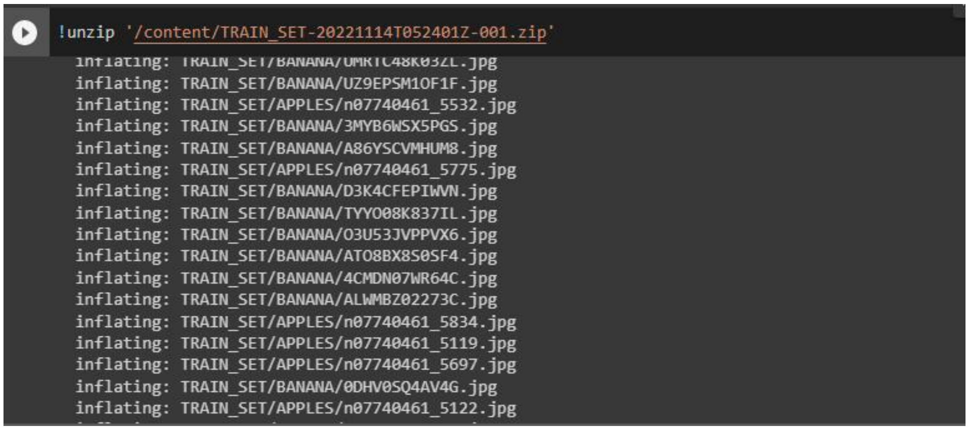
COLLECT THE DATASET

In this project, we have collected images of 3 types of food items: apples, 'banana' and 'orange'. They are saved in the respective sub directories with their respective names. For more accurate results, we can collect images at high resolution and feed the model with more images



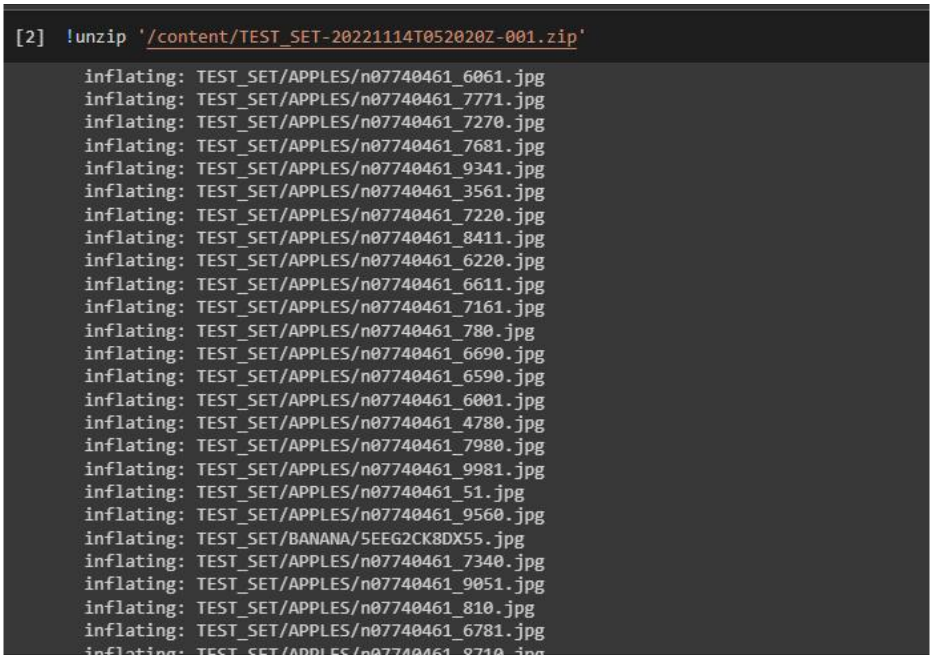
```
..
TEST_SET
  APPLES
  BANANA
  ORANGE
TRAIN_SET
  APPLES
  BANANA
  ORANGE
sample_data
TEST_SET-20221114T052020Z-00...
TRAIN_SET-20221114T052401Z-0...
app.h5
```

>> !unzip '/content/TRAIN_SET-20221114T052401Z-001.zip



```
!unzip '/content/TRAIN_SET-20221114T052401Z-001.zip'
  inflating: TRAIN_SET/BANANA/UMRTC48K03ZL.jpg
  inflating: TRAIN_SET/BANANA/UZ9EPSM1OF1F.jpg
  inflating: TRAIN_SET/APPLES/n07740461_5532.jpg
  inflating: TRAIN_SET/BANANA/3MYB6WSX5PGS.jpg
  inflating: TRAIN_SET/BANANA/A86YSCVMHUM8.jpg
  inflating: TRAIN_SET/APPLES/n07740461_5775.jpg
  inflating: TRAIN_SET/BANANA/D3K4CFEPIWVN.jpg
  inflating: TRAIN_SET/BANANA/TYYO08K837IL.jpg
  inflating: TRAIN_SET/BANANA/O3U53JVPPVX6.jpg
  inflating: TRAIN_SET/BANANA/ATO8BX8S0SF4.jpg
  inflating: TRAIN_SET/BANANA/4CMDN07WR64C.jpg
  inflating: TRAIN_SET/BANANA/ALWMBZ02273C.jpg
  inflating: TRAIN_SET/APPLES/n07740461_5834.jpg
  inflating: TRAIN_SET/APPLES/n07740461_5119.jpg
  inflating: TRAIN_SET/APPLES/n07740461_5697.jpg
  inflating: TRAIN_SET/BANANA/0DHV0SQ4AV4G.jpg
  inflating: TRAIN_SET/APPLES/n07740461_5122.jpg
```

>>!unzip '/content/TEST_SET-20221114T052020Z-001.zip'



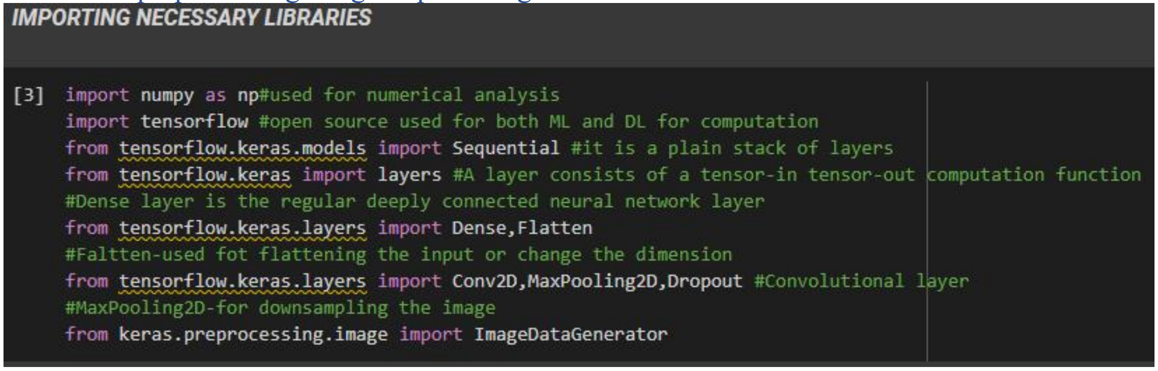
```
[2] !unzip '/content/TEST_SET-20221114T052020Z-001.zip'
      inflating: TEST_SET/APPLES/n07740461_6061.jpg
      inflating: TEST_SET/APPLES/n07740461_7771.jpg
      inflating: TEST_SET/APPLES/n07740461_7270.jpg
      inflating: TEST_SET/APPLES/n07740461_7681.jpg
      inflating: TEST_SET/APPLES/n07740461_9341.jpg
      inflating: TEST_SET/APPLES/n07740461_3561.jpg
      inflating: TEST_SET/APPLES/n07740461_7220.jpg
      inflating: TEST_SET/APPLES/n07740461_8411.jpg
      inflating: TEST_SET/APPLES/n07740461_6220.jpg
      inflating: TEST_SET/APPLES/n07740461_6611.jpg
      inflating: TEST_SET/APPLES/n07740461_7161.jpg
      inflating: TEST_SET/APPLES/n07740461_780.jpg
      inflating: TEST_SET/APPLES/n07740461_6690.jpg
      inflating: TEST_SET/APPLES/n07740461_6590.jpg
      inflating: TEST_SET/APPLES/n07740461_6001.jpg
      inflating: TEST_SET/APPLES/n07740461_4780.jpg
      inflating: TEST_SET/APPLES/n07740461_7980.jpg
      inflating: TEST_SET/APPLES/n07740461_9981.jpg
      inflating: TEST_SET/APPLES/n07740461_51.jpg
      inflating: TEST_SET/APPLES/n07740461_9560.jpg
      inflating: TEST_SET/BANANA/5EEG2CK8DX55.jpg
      inflating: TEST_SET/APPLES/n07740461_7340.jpg
      inflating: TEST_SET/APPLES/n07740461_9051.jpg
      inflating: TEST_SET/APPLES/n07740461_810.jpg
      inflating: TEST_SET/APPLES/n07740461_6781.jpg
      inflating: TEST_SET/APPLES/n07740461_8710.jpg
```

IMPORT THE DATA GENERATOR LIBRARY ,CONFIGURE IMAGE DATA GENERATOR CLASS AND APPLY TO THE TRAIN AND TEST DATA SET

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class. Let us import the ImageDataGenerator class from Keras

i. IMPORT THE REQUIRED LIBRARIES

```
 import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

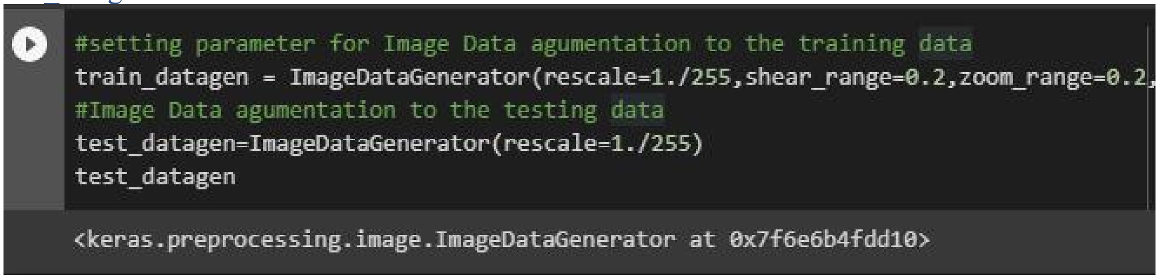


IMPORTING NECESSARY LIBRARIES

```
[3] import numpy as np#used for numerical analysis
    import tensorflow #open source used for both ML and DL for computation
    from tensorflow.keras.models import Sequential #it is a plain stack of layers
    from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
    #Dense layer is the regular deeply connected neural network layer
    from tensorflow.keras.layers import Dense,Flatten
    #Faltten-used fot flattening the input or change the dimension
    from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer
    #MaxPooling2D-for downsampling the image
    from keras.preprocessing.image import ImageDataGenerator
```

ii. IMAGE AUGUMENTATION

```
#setting parameter for Image Data agumentation to the training data
train_datagen =
ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
test_datagen
```

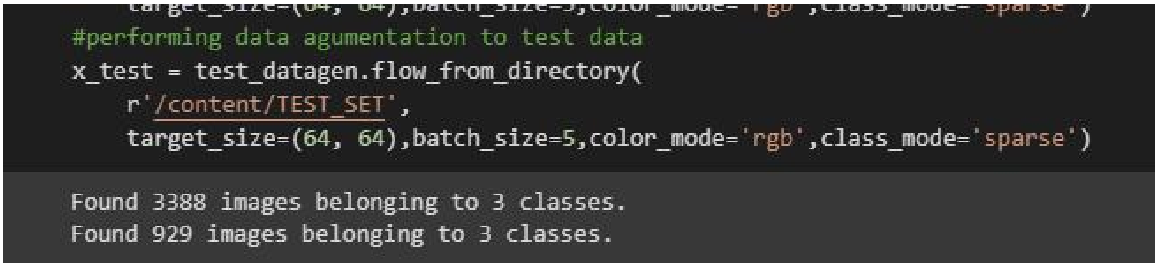


```
#setting parameter for Image Data agumentation to the training data
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
test_datagen
```

```
<keras.preprocessing.image.ImageDataGenerator at 0x7f6e6b4fdd10>
```

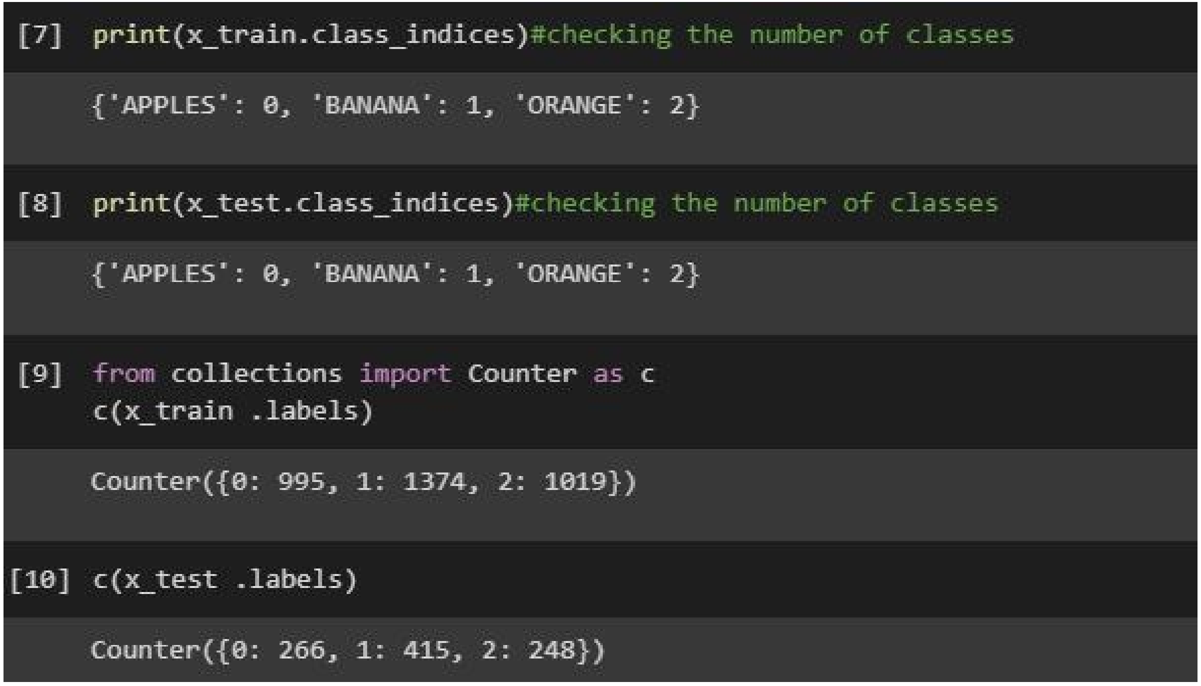
iii. LOADING DATA

```
#performing data agumentation to train data
x_train = train_datagen.flow_from_directory(
   r'/content/TRAIN_SET',
   target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(
   r'/content/TEST_SET',
   target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```



```
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(
    r'/content/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

```
Found 3388 images belonging to 3 classes.
Found 929 images belonging to 3 classes.
```



```
[7]  print(x_train.class_indices)#checking the number of classes
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}
```

```
[8]  print(x_test.class_indices)#checking the number of classes
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}
```

```
[9]  from collections import Counter as c
     c(x_train .labels)
```

```
Counter({0: 995, 1: 1374, 2: 1019})
```

```
[10] c(x_test .labels)
```

```
Counter({0: 266, 1: 415, 2: 248})
```