

Data Collection

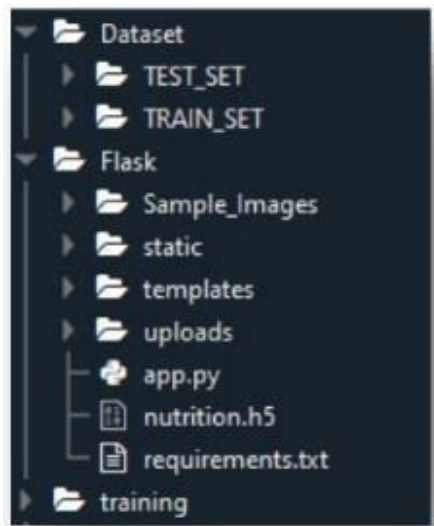
Date	09 November 2022
Team ID	IBM-Project-PNT2022TMID38129
Project Name	AI-powered nutrition analyzer for fitness enthusiasts
Maximum Marks	6

Data Collection

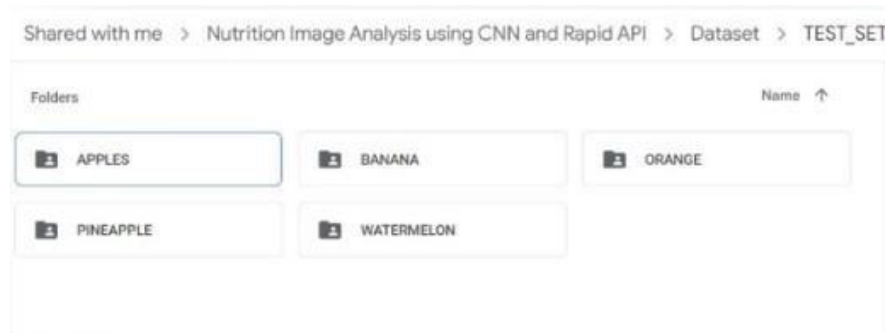
Collect images of different food items organized into subdirectories based on their respective names as shown in the project structure.

Project structure sample:

Create folders of types of food items that need to be recognized.



In this project, we have collected images of 5 types of food items apples, 'banana', 'orange', 'pineapple' and 'watermelon', they are saved in the respective subdirectories with their respective names.

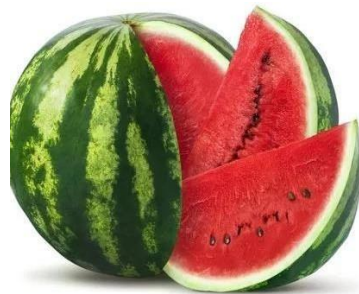


For more accurate results we can collect images of high resolution and feed the model with more images.

Collect images of different food items organized into subdirectories based on their respective names as shown in the project structure.

Create folders of types of food items that need to be recognized.

In this project, we have collected images of 5 types of food items apples, 'banana', 'orange', 'pineapple' and 'watermelon', they are saved in the respective subdirectories with their respective names.



Nutrition Image Analysis using CNN

```
!unzip '/content/Dataset-Fruit.zip'
```

```
inflating: Dataset/TEST_SET/APPLES/n07740461_12841.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12010.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_14600.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9294.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_13931.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12071.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9816.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9813.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9636.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_13800.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9756.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_14211.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9582.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12121.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_958.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12990.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9268.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12930.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9172.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_8842.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12101.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_8689.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_14501.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9944.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_14300.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_14721.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_14450.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9067.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_13950.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_11910.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_8774.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9026.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_13171.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_14570.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_1261.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9157.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_13580.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12300.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_8649.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12350.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_8617.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_904.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_14510.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_8764.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_13030.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_862.jpg
inflating: Dataset/TEST SET/APPLES/n07740461_1191.jpg
```

```

inflating: Dataset/TEST_SET/APPLES/n07740461_13390.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_11871.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9129.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_111.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_8834.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9074.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_8915.jpg
inflating: Dataset/TEST_SET/APPLES/n07740461_12360.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9092.jpg
inflating: Dataset/TRAIN_SET/APPLES/n07740461_9023.jpg

```

▼ Importing Necessary 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
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense, Flatten
#Flatten-used for 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

```

▼ Image Data Augmentation

```

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

```

▼ Loading our data and performing data augmentation

```

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

```

```

Found 4118 images belonging to 5 classes.
Found 1500 images belonging to 5 classes.

```

```

print(x_train.class_indices)#checking the number of classes

```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

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

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

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

```
Counter({0: 995, 1: 1354, 2: 1019, 3: 275, 4: 475})
```

```
from collections import Counter as c  
c(x_test .labels)
```

```
Counter({0: 266, 1: 415, 2: 248, 3: 224, 4: 347})
```