

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

In []:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

creating: Dataset/TEST_SET/
creating: Dataset/TEST_SET/APPLES/
inflating: Dataset/TEST_SET/APPLES/151_100.jpg
inflating: Dataset/TEST_SET/APPLES/152_100.jpg
inflating: Dataset/TEST_SET/APPLES/153_100.jpg
inflating: Dataset/TEST_SET/APPLES/154_100.jpg
inflating: Dataset/TEST_SET/APPLES/155_100.jpg
inflating: Dataset/TEST_SET/APPLES/3_100.jpg
inflating: Dataset/TEST_SET/APPLES/32_100.jpg
inflating: Dataset/TEST_SET/APPLES/33_100.jpg
inflating: Dataset/TEST_SET/APPLES/34_100.jpg
inflating: Dataset/TEST_SET/APPLES/35_100.jpg
inflating: Dataset/TEST_SET/APPLES/36_100.jpg
inflating: Dataset/TEST_SET/APPLES/37_100.jpg
inflating: Dataset/TEST_SET/APPLES/38_100.jpg
inflating: Dataset/TEST_SET/APPLES/39_100.jpg
inflating: Dataset/TEST_SET/APPLES/4_100.jpg
inflating: Dataset/TEST_SET/APPLES/40_100.jpg
inflating: Dataset/TEST_SET/APPLES/41_100.jpg
inflating: Dataset/TEST_SET/APPLES/42_100.jpg
inflating: Dataset/TEST_SET/APPLES/43_100.jpg
inflating: Dataset/TEST_SET/APPLES/44_100.jpg
inflating: Dataset/TEST_SET/APPLES/45_100.jpg
inflating: Dataset/TEST_SET/APPLES/46_100.jpg
inflating: Dataset/TEST_SET/APPLES/47_100.jpg
inflating: Dataset/TEST_SET/APPLES/48_100.jpg
inflating: Dataset/TEST_SET/APPLES/49_100.jpg
inflating: Dataset/TEST_SET/APPLES/5_100.jpg
inflating: Dataset/TEST_SET/APPLES/50_100.jpg
inflating: Dataset/TEST_SET/APPLES/51_100.jpg
inflating: Dataset/TEST_SET/APPLES/52_100.jpg
inflating: Dataset/TEST_SET/APPLES/53_100.jpg
inflating: Dataset/TEST_SET/APPLES/54_100.jpg
inflating: Dataset/TEST_SET/APPLES/55_100.jpg
inflating: Dataset/TEST_SET/APPLES/56_100.jpg
inflating: Dataset/TEST_SET/APPLES/57_100.jpg
inflating: Dataset/TEST_SET/APPLES/58_100.jpg
inflating: Dataset/TEST_SET/APPLES/59_100.jpg
inflating: Dataset/TEST_SET/APPLES/6_100.jpg
inflating: Dataset/TEST_SET/APPLES/60_100.jpg
inflating: Dataset/TEST_SET/APPLES/61_100.jpg
inflating: Dataset/TEST_SET/APPLES/62_100.jpg
inflating: Dataset/TEST_SET/APPLES/63_100.jpg
inflating: Dataset/TEST_SET/APPLES/64_100.jpg
inflating: Dataset/TEST_SET/APPLES/65_100.jpg
inflating: Dataset/TEST_SET/APPLES/66_100.jpg
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

from keras.preprocessing.image import ImageDataGenerator

import numpy as np
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential
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

train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=
0.2, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

x_train=train_datagen.flow_from_directory(
    r'/content/drive/MyDrive/ibm_final/TRAIN_SET',
    target_size=(64,64), batch_size=32, class_mode='categorical')
x_test=train_datagen.flow_from_directory(
    r'/content/drive/MyDrive/ibm_final/TEST_SET',
    target_size=(64,64), batch_size=32, class_mode='categorical')

Found 2626 images belonging to 5 classes.
Found 1055 images belonging to 5 classes.

print(x_train.class_indices)
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
print(x_test.class_indices)
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}

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

Counter({0: 606, 1: 445, 2: 479, 3: 621, 4: 475})

```