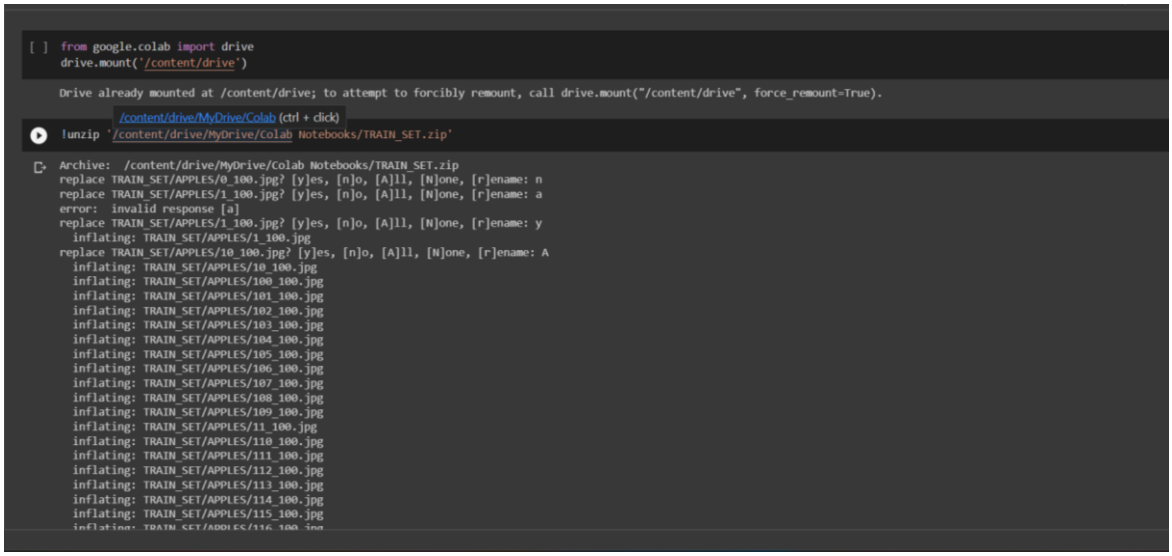
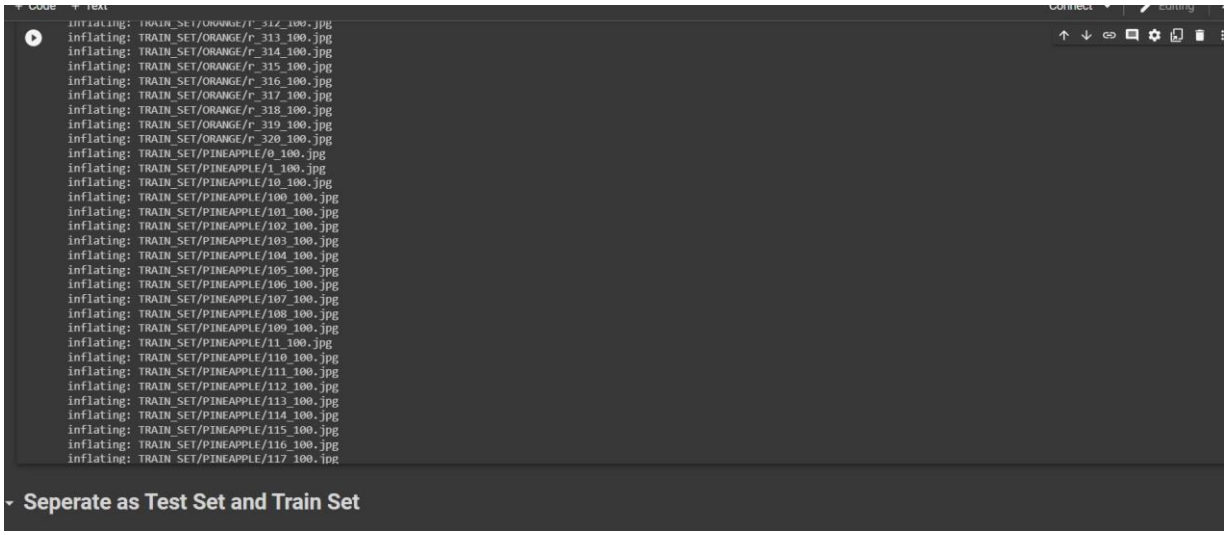
UNZIP THE FILE



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

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

    /content/drive/MyDrive/Colab (ctrl + click)
    !unzip '/content/drive/MyDrive/Colab Notebooks/TRAIN_SET.zip'

    Archive:  /content/drive/MyDrive/Colab Notebooks/TRAIN_SET.zip
    replace TRAIN_SET/APPLES/0_100.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
    replace TRAIN_SET/APPLES/1_100.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: a
    error:  invalid response [a]
    replace TRAIN_SET/APPLES/1_100.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
      inflating: TRAIN_SET/APPLES/1_100.jpg
    replace TRAIN_SET/APPLES/10_100.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
      inflating: TRAIN_SET/APPLES/10_100.jpg
      inflating: TRAIN_SET/APPLES/100_100.jpg
      inflating: TRAIN_SET/APPLES/101_100.jpg
      inflating: TRAIN_SET/APPLES/102_100.jpg
      inflating: TRAIN_SET/APPLES/103_100.jpg
      inflating: TRAIN_SET/APPLES/104_100.jpg
      inflating: TRAIN_SET/APPLES/105_100.jpg
      inflating: TRAIN_SET/APPLES/106_100.jpg
      inflating: TRAIN_SET/APPLES/107_100.jpg
      inflating: TRAIN_SET/APPLES/108_100.jpg
      inflating: TRAIN_SET/APPLES/109_100.jpg
      inflating: TRAIN_SET/APPLES/11_100.jpg
      inflating: TRAIN_SET/APPLES/110_100.jpg
      inflating: TRAIN_SET/APPLES/111_100.jpg
      inflating: TRAIN_SET/APPLES/112_100.jpg
      inflating: TRAIN_SET/APPLES/113_100.jpg
      inflating: TRAIN_SET/APPLES/114_100.jpg
      inflating: TRAIN_SET/APPLES/115_100.jpg
      inflating: TRAIN_SET/APPLES/116_100.jpg
```



```
      inflating: TRAIN_SET/ORANGE/r_313_100.jpg
      inflating: TRAIN_SET/ORANGE/r_314_100.jpg
      inflating: TRAIN_SET/ORANGE/r_315_100.jpg
      inflating: TRAIN_SET/ORANGE/r_316_100.jpg
      inflating: TRAIN_SET/ORANGE/r_317_100.jpg
      inflating: TRAIN_SET/ORANGE/r_318_100.jpg
      inflating: TRAIN_SET/ORANGE/r_319_100.jpg
      inflating: TRAIN_SET/ORANGE/r_320_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/0_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/1_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/10_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/100_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/101_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/102_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/103_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/104_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/105_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/106_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/107_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/108_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/109_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/11_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/110_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/111_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/112_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/113_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/114_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/115_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/116_100.jpg
      inflating: TRAIN_SET/PINEAPPLE/117_100.jpg

## Seperate as Test Set and Train Set
```

SEPARATE THE TEST AND TRAIN SET

Seperate as Test Set and Train Set

```
[ ] from keras.preprocessing.image import ImageDataGenerator
    import tensorflow as tf
```

```
[ ] image_generator =tf.keras.preprocessing.image.ImageDataGenerator(validation_split=0.2)
    train_data_gen=image_generator.flow_from_directory(directory='TRAIN_SET',subset='training')
    val_data_gen = image_generator.flow_from_directory(directory='TRAIN_SET',subset='validation')
```

```
Found 2102 images belonging to 5 classes.
Found 524 images belonging to 5 classes.
```

```
[ ] 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/TRAIN_SET',
    target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse')

x_test=test_datagen.flow_from_directory(
    r'/content/TRAIN_SET',
     target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

```
Found 2626 images belonging to 5 classes.
Found 2626 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})
```

Importing Neccesarry Libraries

```
[ ] import numpy as np
    import tensorflow
    from tensorflow.keras.models import Sequential
    from tensorflow.keras import layers
    from tensorflow.keras.layers import Dense,Flatten
    from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
    from keras.preprocessing.image import ImageDataGenerator
```

INITIALIZING & CREATING THE MODEL

Initializing The Model

```
[ ] model=Sequential()
```

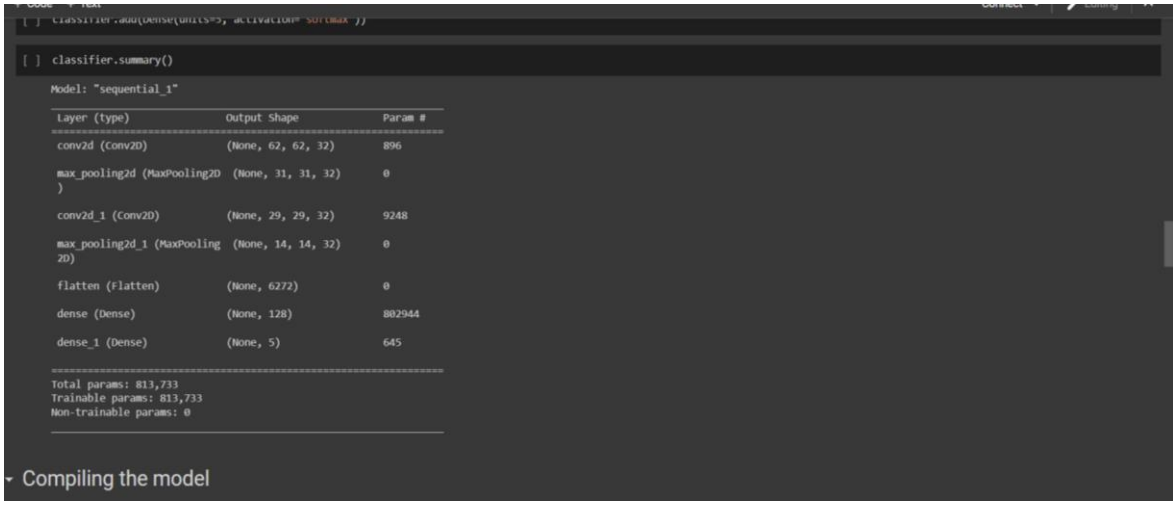
Creating the model

```
[ ] classifier=Sequential()
    classifier.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
    classifier.add(MaxPooling2D(pool_size=(2,2)))
    classifier.add(Conv2D(32,(3,3),activation='relu'))
    classifier.add(MaxPooling2D(pool_size=(2,2)))
    classifier.add(Flatten())
```

```
[ ] classifier.add(Dense(units=128, activation='relu'))
    classifier.add(Dense(units=5, activation='softmax'))
```

```
classifier.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )
```

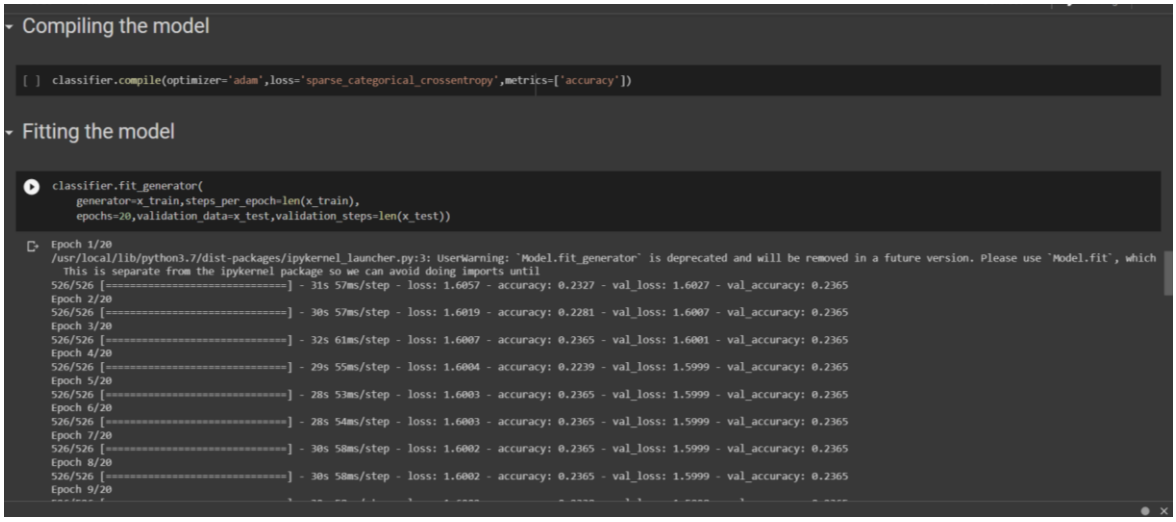


```
[ ] classifier.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 5)                 645

=================================================================
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0
_________________________________________________________________
```

Compiling the model

COMPILING & FITTING THE MODEL



SAVING THE MODEL & PREDICTING THE RESULTS