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

```
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1605.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_2105.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_235.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1742.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1916.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_226.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1808.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_2077.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1754.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1629.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_173.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1849.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1735.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_175.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1623.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_2326.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_2026.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_183.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1812.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1848.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_158.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_165.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1723.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1859.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1787.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1656.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1698.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1803.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1926.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1743.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1668.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1677.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1612.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1734.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_166.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1949.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_15003.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1582.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1647.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1834.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1679.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1978.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1889.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1937.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_14802.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1844.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_242.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_1923.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_15087.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_14869.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_15074.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_14997.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_14939.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_14889.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_15064.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_15047.jpg
  inflating: Dataset/TRAIN_SET/APPLES/n07740461_14886.jpg
```

```
        inflating: Dataset/TRAIN_SET/APPLES/n07740461_14886.jpg
        inflating: Dataset/TRAIN_SET/APPLES/n07740461_15239.jpg
```

```
from keras.preprocessing.image import ImageDataGenerator


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


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')
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 929 images belonging to 5 classes.
```

```
### Importing Neccessary 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 computatio
#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


model=Sequential()


### Creating the model
# Initializing the CNN
classifier = Sequential()
# First convolution Layer and pooling
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Second convolution Layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution Layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Flattening the Layers
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
 )

 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 the CNN # categorical_crossentropy for more than 2
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['acc


##Fitting the model
classifier.fit_generator(
    generator=x_train, steps_per_epoch = len(x_train),
    epochs=20, validation_data=x_test, validation_steps = len(x_test))
```

```
Epoch 1/20
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: UserWarning: `Model.f
  after removing the cwd from sys.path.
824/824 [==============================] - 20s 15ms/step - loss: 0.5772 - accuracy: 0
Epoch 2/20
824/824 [==============================] - 11s 14ms/step - loss: 0.4342 - accuracy: 0
Epoch 3/20
824/824 [==============================] - 11s 14ms/step - loss: 0.3827 - accuracy: 0
Epoch 4/20
824/824 [==============================] - 11s 14ms/step - loss: 0.3494 - accuracy: 0
Epoch 5/20
824/824 [==============================] - 12s 15ms/step - loss: 0.3393 - accuracy: 0
Epoch 6/20
824/824 [==============================] - 11s 14ms/step - loss: 0.3135 - accuracy: 0
Epoch 7/20
824/824 [==============================] - 11s 14ms/step - loss: 0.3015 - accuracy: 0
Epoch 8/20
824/824 [==============================] - 11s 14ms/step - loss: 0.2831 - accuracy: 0
Epoch 9/20
824/824 [==============================] - 11s 14ms/step - loss: 0.2555 - accuracy: 0
Epoch 10/20
824/824 [==============================] - 12s 15ms/step - loss: 0.2486 - accuracy: 0
```

```
Epoch 11/20
824/824 [==============================] - 11s 14ms/step - loss: 0.2170 - accuracy: 0
Epoch 12/20
824/824 [==============================] - 11s 14ms/step - loss: 0.2176 - accuracy: 0
Epoch 13/20
824/824 [==============================] - 11s 14ms/step - loss: 0.1990 - accuracy: 0
Epoch 14/20
824/824 [==============================] - 11s 14ms/step - loss: 0.1913 - accuracy: 0
Epoch 15/20
824/824 [==============================] - 12s 15ms/step - loss: 0.1894 - accuracy: 0
Epoch 16/20
824/824 [==============================] - 11s 14ms/step - loss: 0.1691 - accuracy: 0
Epoch 17/20
824/824 [==============================] - 11s 14ms/step - loss: 0.1582 - accuracy: 0
Epoch 18/20
824/824 [==============================] - 11s 14ms/step - loss: 0.1593 - accuracy: 0
Epoch 19/20
824/824 [==============================] - 11s 14ms/step - loss: 0.1259 - accuracy: 0
Epoch 20/20
824/824 [==============================] - 13s 16ms/step - loss: 0.1199 - accuracy: 0
<keras.callbacks.History at 0x7fe8bc68cc50>
```

```
### Saving our model
# Save the model
classifier.save('nutrition.h5')


### Predicting our results
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model = load_model("nutrition.h5") #Loading the model for testing
```

Colab paid products - Cancel contracts here