```
from keras.preprocessing.image import ImageDataGenerator

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

)


x_test=test_datagen.flow_from_directory(

r'/content/drive/MyDrive/TRAIN_SET',target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse'

)

Found 1210 images belonging to 5 classes.

Found 1210 images belonging to 5 classes.

print(x_train.class_indices)


print(x_test.class_indices)

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

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

from collections import Counter as c

c(x_train .labels)

Counter({2: 114, 3: 621, 4: 475})

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

import tensorflow as tf
```

```
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()

# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 4s 0us/step
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10))
model.summary()
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 30, 30, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 64)         0
```

2D)

conv2d_4 (Conv2D) (None, 4, 4, 64) 36928

flatten_1 (Flatten) (None, 1024) 0

dense (Dense) (None, 64) 65600

dense_1 (Dense) (None, 10) 650

===

Total params: 122,570

Trainable params: 122,570

Non-trainable params: 0

_____

```
#Compiling the model
model.compile(optimizer='adam',
        loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        metrics=['accuracy'])
#Fitting the model
history = model.fit(train_images, train_labels, epochs=10,
            validation_data=(test_images, test_labels))
```

Epoch 1/10

1563/1563 [==============================] - 85s 54ms/step - loss: 1.5347 - accuracy: 0.4394 - val_loss: 1.2535 - val_accuracy: 0.5445

Epoch 2/10

1563/1563 [==============================] - 85s 55ms/step - loss: 1.1773 - accuracy: 0.5808 - val_loss: 1.1283 - val_accuracy: 0.5983

Epoch 3/10

1563/1563 [==============================] - 81s 52ms/step - loss: 1.0302 - accuracy: 0.6365 - val_loss: 1.0501 - val_accuracy: 0.6269

Epoch 4/10

1563/1563 [==============================] - 84s 54ms/step - loss: 0.9381 - accuracy: 0.6696 - val_loss: 0.9480 - val_accuracy: 0.6668

Epoch 5/10

1563/1563 [==============================] - 85s 55ms/step - loss: 0.8697 - accuracy: 0.6958 - val_loss: 0.9290 - val_accuracy: 0.6762

Epoch 6/10

1563/1563 [==============================] - 83s 53ms/step - loss: 0.8083 - accuracy: 0.7173 - val_loss: 0.8973 - val_accuracy: 0.6915

Epoch 7/10

1563/1563 [==============================] - 82s 52ms/step - loss: 0.7693 - accuracy: 0.7280 - val_loss: 0.8785 - val_accuracy: 0.6967

Epoch 8/10

1563/1563 [==============================] - 84s 54ms/step - loss: 0.7229 - accuracy: 0.7467 - val_loss: 0.8668 - val_accuracy: 0.6999

Epoch 9/10

1563/1563 [==============================] - 80s 51ms/step - loss: 0.6890 - accuracy: 0.7580 - val_loss: 0.8592 - val_accuracy: 0.7107

Epoch 10/10

1563/1563 [==============================] - 79s 51ms/step - loss: 0.6501 - accuracy: 0.7698 - val_loss: 0.8901 - val_accuracy: 0.7108

#Saving our model

model.save('nutrition.h5')

#Prediciting our results

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

model=load_model('nutrition.h5')

img=image.load_img('/content/drive/MyDrive/1_100.jpg',target_size=(70,70))

img

x= image.img_to_array(img)

x = np.expand_dims(x, axis=0)

index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']

result=str(index[0])

```
result
'APPLES'
img=load_img(r"/content/16_100.jpg",grayscale=False,target_size=(64,64))
x=img_to_array(img)
x
x=np.expand_dims(x,axis=0)
x



#pred_x=model.predict(x)
#pred=np.argmax(pred_x,axis=1)


pred = (model.predict(x) > 0.5).astype("int32")


type(pred[0])
1/1 [==============================] - 0s 21ms/step
numpy.ndarray
from google.colab import drive
drive.mount('/content/drive')
index=['APPLE','BANANA','ORANGE','PINEAPPLE','WATERMELON']
out_images = np.array(x)[index.astype(int)]
out_images
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
 in
      1 index=['APPLE','BANANA','ORANGE','PINEAPPLE','WATERMELON']
----> 2 out_images = np.array(x)[index.astype(int)]
      3 out_images

AttributeError: 'list' object has no attribute 'astype'
index=['APPLE','BANANA','ORANGE','PINEAPPLE','WATERMELON']
```

```
result=str(index[0])
result
'APPLE'
```