PROJECT DEVELOPMENT PHASE
SPRINT-III

Date	09 November 2022
TeamID	PNT2022TMID49469
Project Name	AI-Powered Nutrition Analyzer For Fitness Enthusiasts
MaximumMarks	8 mark

MODEL BUILDING

Click Here To View The Project (Hyperlink)

Importing The Model Building Libraries

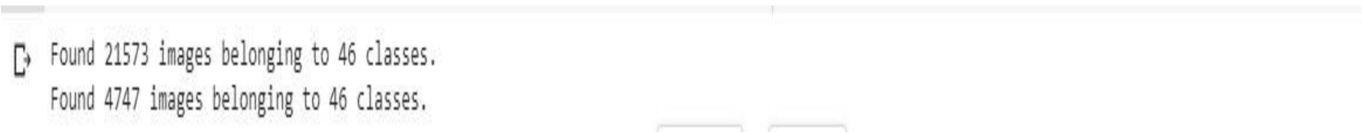
```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from keras.layers import Dense
from keras.layers import Conv2D
from keras.layers import MaxPooling2D,Dropout
from keras.layers import Flatten
```

Configure ImageDataGenerator Class

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

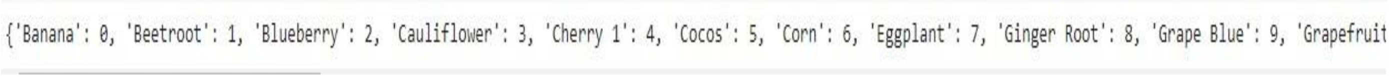
Apply Image DataGenerator Functionality To Trainset And Testset

```
x_train = train_datagen.flow_from_directory(r"/content/Dataset/training",target_size=(
64,64),batch_size=32,color_mode="rgb",class_mode="sparse")
x_test = test_datagen.flow_from_directory(r"/content/Dataset/testing",target_size=
(64,64),batch_size=32,color_mode="rgb",class_mode="sparse")
```



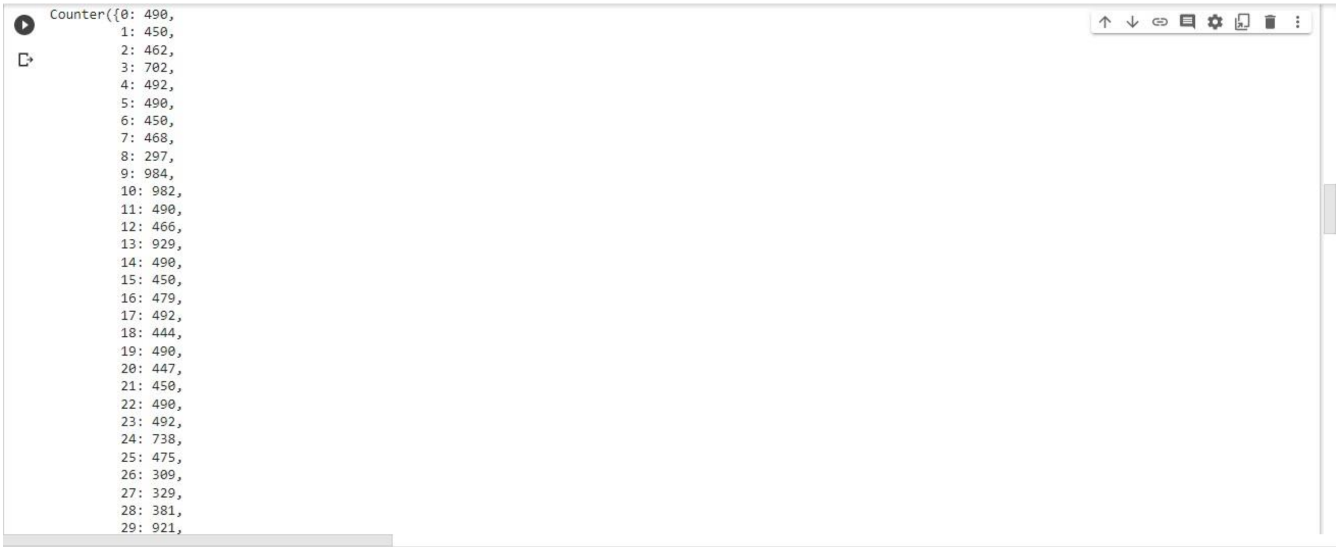
```
Found 21573 images belonging to 46 classes.
Found 4747 images belonging to 46 classes.
```

```
print(x_train.class_indices)
print(x_test.class_indices)
```



```
{'Banana': 0, 'Beetroot': 1, 'Blueberry': 2, 'Cauliflower': 3, 'Cherry 1': 4, 'Cocos': 5, 'Corn': 6, 'Eggplant': 7, 'Ginger Root': 8, 'Grape Blue': 9, 'Grapefruit
```

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



```
Counter({0: 490,
         1: 450,
         2: 462,
         3: 702,
         4: 492,
         5: 490,
         6: 450,
         7: 468,
         8: 297,
         9: 984,
         10: 982,
         11: 490,
         12: 466,
         13: 929,
         14: 490,
         15: 450,
         16: 479,
         17: 492,
         18: 444,
         19: 490,
         20: 447,
         21: 450,
         22: 490,
         23: 492,
         24: 738,
         25: 475,
         26: 309,
         27: 329,
         28: 381,
         29: 921,
```

Initializing The Model

```
model=Sequential()
```

Adding CNN Layers

```
#Initializing the CNN
classifier=Sequential()
```

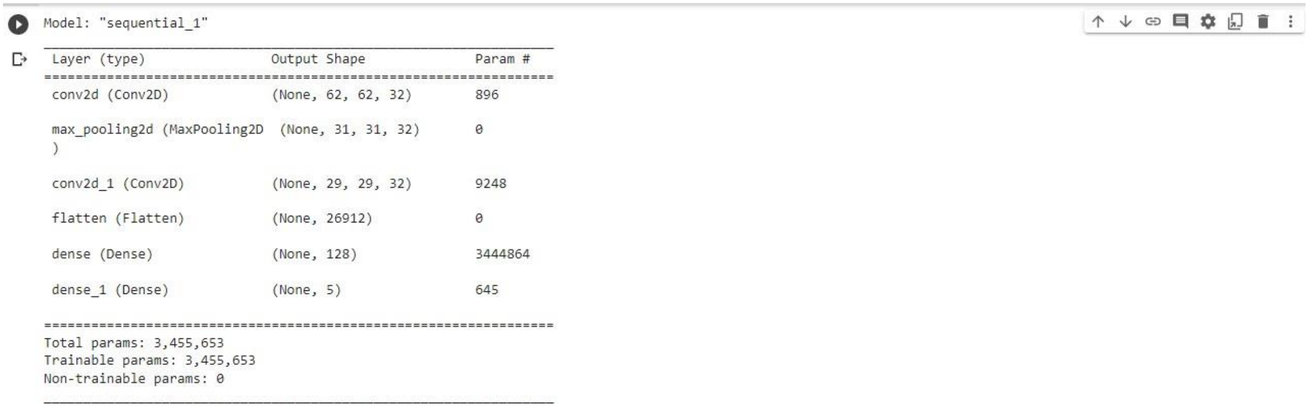
```
# First convolution layer and pooling
classifier.add(Conv2D(32, (3,3), activation = "relu", input_shape = (64,64,3) ))
classifier.add(MaxPooling2D(pool_size=(2,2)))
```

```
# First convolution layer and pooling
classifier.add(Conv2D(32, (3,3), activation = "relu"))
classifier.add(Flatten())
```

Adding Dense Layers

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

 flatten (Flatten)           (None, 26912)             0

 dense (Dense)               (None, 128)               3444864

 dense_1 (Dense)             (None, 5)                 645

=================================================================
Total params: 3,455,653
Trainable params: 3,455,653
Non-trainable params: 0
_________________________________________________________________
```

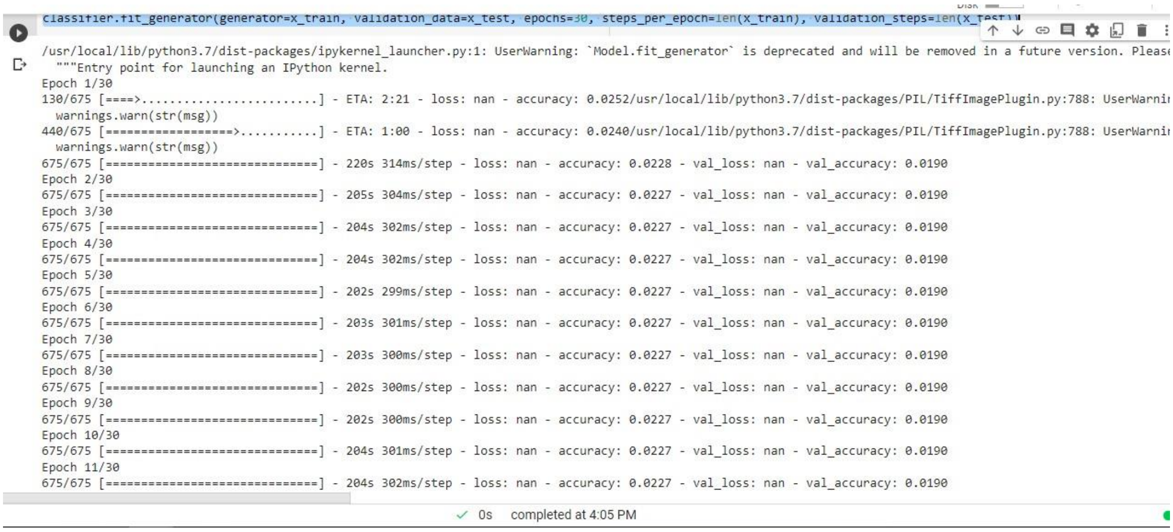
Configure The Learning Process

### Compiling the model

classifier.compile(optimizer='adam',loss="sparse_categorical_crossentropy",metrics=['accuracy'])

Train The Model

classifier.fit_generator(generator=x_train, validation_data=x_test, epochs=30, steps_per_epoch=len(x_train), validation_steps=len(x_test))



```
classifier.fit_generator(generator=x_train, validation_data=x_test, epochs=30, steps_per_epoch=len(x_train), validation_steps=len(x_test))
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please
  """Entry point for launching an IPython kernel.
Epoch 1/30
130/675 [====>.........................] - ETA: 2:21 - loss: nan - accuracy: 0.0252/usr/local/lib/python3.7/dist-packages/PIL/TiffImagePlugin.py:788: UserWarnin
  warnings.warn(str(msg))
440/675 [==================>...........] - ETA: 1:00 - loss: nan - accuracy: 0.0240/usr/local/lib/python3.7/dist-packages/PIL/TiffImagePlugin.py:788: UserWarnin
  warnings.warn(str(msg))
675/675 [==============================] - 220s 314ms/step - loss: nan - accuracy: 0.0228 - val_loss: nan - val_accuracy: 0.0190
Epoch 2/30
675/675 [==============================] - 205s 304ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 3/30
675/675 [==============================] - 204s 302ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 4/30
675/675 [==============================] - 204s 302ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 5/30
675/675 [==============================] - 202s 299ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 6/30
675/675 [==============================] - 203s 301ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 7/30
675/675 [==============================] - 203s 300ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 8/30
675/675 [==============================] - 202s 300ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 9/30
675/675 [==============================] - 202s 300ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 10/30
675/675 [==============================] - 204s 301ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
Epoch 11/30
675/675 [==============================] - 204s 302ms/step - loss: nan - accuracy: 0.0227 - val_loss: nan - val_accuracy: 0.0190
```

Save The Model

classifier.save('scanmynutri.h5')

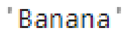
Test The Model

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model("scanmynutri.h5")
```

```
img = image.load_img('/content/Dataset/training/Banana/101_100.jpg',target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
predicted_class=model.predict(x)
```

```
labels = ['Banana', 'Beetroot', 'Blueberry', 'Cauliflower', 'Cherry 1', 'Cocos', 'Eggplan', 'Ginger Root', 'Grape Blue',
'Grapefruit Pink', 'Guava', 'Kiwi', 'Lemon', 'Mango', 'Onion Red', 'Orange', 'Papaya', 'Pepper Green', 'Pineapple',
'Plum', 'Potato Red', 'Raspberry', 'Strawberry', 'Tomato 1', 'Watermelon', 'burger', 'butter_naan', 'chai', 'chapati',
'chole_bhature', 'dal_makhani', 'dhokla', 'fried_rice', 'idli', 'jalebi', 'kaathi_rolls', 'kadai_paneer', 'kulfi',
'masala_dosa', 'momos', 'paani_puri', 'pakode', 'pav_bhaji', 'pizza', 'samosa']

labels[np.argmax(predicted_class)]
```



```
'Banana'
```