Team ID : PNT2022TMID21978
Date : 6 Nov 2022

Sprint - 2

Image Augmentation / PreProcessing :

```
#Import req. Lib.
from tensorflow.keras.preprocessing.image import ImageDataGenerator #Augmentation On Training Variable train_datagen =
ImageDataGenerator(rescale= 1./255,                  zoom_range=0.2,                  horizontal_flip =True)


#Augmentation On Testing Variable test_datagen =
ImageDataGenerator(rescale= 1./255)


#Augmentation On Training Variable ftrain =
train_datagen.flow_from_directory('/content/Dataset_Collection/Train',
target_size=(64,64),
class_mode='categorical',
batch_size=100)

     Found 4111 images belonging to 5 classes.


#Augmentation On Training Variable ftest =
test_datagen.flow_from_directory('/content/Dataset_Collection/Test',
target_size=(64,64),
class_mode='categorical',
batch_size=100)
```

```
Found 429 images belonging to 5 classes.
```

Model Building Adding Layers:

```
#Import req. Lib. from tensorflow.keras.models import Sequential from
tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense


# Build a CNN Block:
model = Sequential() #intializing sequential model
model.add(Convolution2D(32,(3,3),activation='relu', input_shape=(64,64,3))) #convolution layer
model.add(MaxPooling2D(pool_size=(2, 2))) #Maxpooling layer model.add(Flatten()) #Flatten
layer model.add(Dense(400,activation='relu')) #Hidden Layer 1
model.add(Dense(200,activation='relu')) #Hidden Layer 2
model.add(Dense(5,activation='softmax')) #Output Layer
```

Compiling

```
#                    Compiling                    The                    Model...

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

Fit/Train The Model

```
#Train Model: model.fit_generator(ftrain,
steps_per_epoch=len(ftrain),
epochs=10,
validation_data=ftest,
validation_steps=len(ftest))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: UserWarning: `Model.fit_generator` is deprecated and w
```

```
Epoch 1/10
42 /42 [==============================] - 17s 197ms/step - loss: 1.4441 - accuracy: 0.6132 - val_loss: 0.3529 -
val_acc
Epoch 2/10
42 /42 [==============================] - 8s 195ms/step - loss: 0.3129 - accuracy: 0.8964 - val_loss: 0.1108 -
val_accu
Epoch 3/10
42 /42 [==============================] - 8s 194ms/step - loss: 0.2162 - accuracy: 0.9226 - val_loss: 0.1193 -
val_accu
Epoch 4/10
42 /42 [==============================] - 8s 194ms/step - loss: 0.1945 - accuracy: 0.9319 - val_loss: 0.0962 -
val_accu
Epoch 5/10
42 /42 [==============================] - 8s 194ms/step - loss: 0.1539 - accuracy: 0.9438 - val_loss: 0.2203 -
val_accu
Epoch 6/10
42 /42 [==============================] - 8s 193ms/step - loss: 0.1729 - accuracy: 0.9358 - val_loss: 0.1221 -
val_accu
Epoch 7/10
42 /42 [==============================] - 8s 196ms/step - loss: 0.1290 - accuracy: 0.9535 - val_loss: 0.0437 -
val_accu
Epoch 8/10
42 /42 [==============================] - 8s 195ms/step - loss: 0.1076 - accuracy: 0.9604 - val_loss: 0.1042 -
val_accu
Epoch 9/10
42 /42 [==============================] - 8s 195ms/step - loss: 0.1055 - accuracy: 0.9608 - val_loss: 0.0655 -
val_accu
Epoch 10/10
42 /42 [==============================] - 8s 194ms/step - loss: 0.1005 - accuracy: 0.9642 - val_loss: 0.0338 -
val_accu
< keras.callbacks.History at  0x7f31d8214150>
```

Saving The Model :

```
#Save Model
model.save('fruitsmodel.h5') Testing
```

The Model :

```
#Import req. Lib. from
tensorflow.keras.preprocessing import image
import numpy as np


#Testing No 1 :- img =
image.load_img('/content/Dataset_Collection/Test/guava/108_100.jpg',target_size=(64,64)) #Reading image f =
image.img_to_array(img) #Convertinng image to array f = np.expand_dims(f,axis=0) #Expanding dimensions pred =
np.argmax(model.predict(f)) #predicting higher propability index op =
['DATES','GUAVA','ORANGE','PINEAPPLE','WATERMELON'] #Creating List op[pred] #List indexing with output

     1/1 [==============================] - 0s 15ms/step
     'GUAVA'


#Testing No 2 :- img =
image.load_img('/content/Dataset_Collection/Test/pinenapple/img_1191.jpeg',target_size=(64,64)) #Reading image f =
image.img_to_array(img) #Convertinng image to array f = np.expand_dims(f,axis=0) #Expanding dimensions pred =
np.argmax(model.predict(f)) #predicting higher propability index op =
['DATES','GUAVA','ORANGE','PINEAPPLE','WATERMELON'] #Creating List op[pred] #List indexing with output

     1/1 [==============================] - 0s 17ms/step
     'PINEAPPLE'
```

```
#Testing No 3 :- img =
image.load_img('/content/Dataset_Collection/Test/watermelon/img_11.jpeg',target_size=(64,64)) #Reading image f =
image.img_to_array(img) #Convertinng image to array f = np.expand_dims(f,axis=0) #Expanding dimensions pred =
np.argmax(model.predict(f)) #predicting higher propability index op =
['DATES','GUAVA','ORANGE','PINEAPPLE','WATERMELON'] #Creating List op[pred] #List indexing with output
```

```
1/1 [==============================] - 0s 15ms/step
'WATERMELON'
```

Model Tuning: `from tensorflow.keras.callbacks import EarlyStopping,`

`ReduceLROnPlateau`

```
early_stop = EarlyStopping(monitor='val_accuracy',
patience=5)

lr = ReduceLROnPlateau(monitor='val_accuracy',
factor=0.5,
min_lr=0.00001) callback = [early_stop,lr]
```

```
# Train model
model.fit_generator(ftrain,
steps_per_epoch=len(ftrain),
epochs=100,
callbacks=callback,
validation_data=ftest,
validation_steps=len(ftest))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8: UserWarning: `Model.fit_generator` is deprecated and w

Epoch 1/100
```

```
42 /42 [==============================] - 8s 194ms/step - loss: 0.0898 - accuracy: 0.9657 - val_loss: 0.0791 -
val_accu
Epoch 2/100
42 /42 [==============================] - 8s 196ms/step - loss: 0.0746 - accuracy: 0.9747 - val_loss: 0.0165 -
val_accu
Epoch 3/100
42 /42 [==============================] - 8s 191ms/step - loss: 0.0747 - accuracy: 0.9740 - val_loss: 0.0145 -
val_accu
Epoch 4/100
42 /42 [==============================] - 8s 191ms/step - loss: 0.0701 - accuracy: 0.9728 - val_loss: 0.0155 -
val_accu
Epoch 5/100
42 /42 [==============================] - 8s 192ms/step - loss: 0.0703 - accuracy: 0.9737 - val_loss: 0.0155 -
val_accu
Epoch 6/100
42 /42 [==============================] - 8s 190ms/step - loss: 0.0854 - accuracy: 0.9674 - val_loss: 0.0411 -
val_accu
Epoch 7/100
42 /42 [==============================] - 8s 190ms/step - loss: 0.0680 - accuracy: 0.9742 - val_loss: 0.0144 -
val_accu
Epoch 8/100
42 /42 [==============================] - 8s 192ms/step - loss: 0.0490 - accuracy: 0.9830 - val_loss: 0.0216 -
val_accu
< keras.callbacks.History at  0x7f31725a9710>
```

```
#Testing No 4 :- img =
image.load_img('/content/Dataset_Collection/Test/orange/img_1271.jpeg',target_size=(64,64)) #Reading image f =
image.img_to_array(img) #Convertinng image to array f = np.expand_dims(f,axis=0) #Expanding dimensions pred =
np.argmax(model.predict(f)) #predicting higher propability index op =
['DATES','GUAVA','ORANGE','PINEAPPLE','WATERMELON'] #Creating List op[pred] #List indexing with output
```

```
1/1 [==============================] - 0s 15ms/step
'ORANGE'
```

```
#Testing No 5 :
img = image.load_img('/content/Dataset_Collection/Test/dates/104_100.jpg',target_size=(64,64)) #Reading image
f = image.img_to_array(img) #Convertinng image to array
f = np.expand_dims(f,axis=0) #Expanding dimensions
pred = np.argmax(model.predict(f)) #predicting higher propability index
op = ['DATES','GUAVA','ORANGE','PINEAPPLE','WATERMELON'] #Creating List
op[pred] #List indexing with output
```

```
1/1 [==============================] - 0s 15ms/step
'DATES'
```