Team ID	**PNT2022TMID00728**
Project Name	**AI-powered Nutrition Analyzer for Fitness Enthusiasts**

IBM | Service Details - IBM Cloud | Fruit Classifier Model - IBM

jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/46f9fa0f-38c9-49c4-a44a-a1e6f6acf80b

IBM Watson Studio | Search in your workspaces | Buy | Sanjay Bharath's Account | Tokyo

Projects / Fruit Classification / Fruit Classifier Model

File Edit View Insert Cell Kernel Help | Trusted | Python 3.9

Run | Format Code

```
model.fit(x=train_batches, validation_data= valid_batches, epochs=10, verbose=2)
model.save("modelfinal.h5")
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 32)      896

 max_pooling2d (MaxPooling2D  (None, 112, 112, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 112, 112, 64)      18496

 max_pooling2d_1 (MaxPooling  (None, 56, 56, 64)       0
 2D)

 flatten (Flatten)           (None, 200704)            0

 dense (Dense)               (None, 4)                 802820

=================================================================
Total params: 822,212
Trainable params: 822,212
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/10
195/195 - 110s - loss: 7.7628 - accuracy: 0.9487 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 110s/epoch - 562ms/step
Epoch 2/10
195/195 - 108s - loss: 2.8505e-06 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 108s/epoch - 552ms/step
Epoch 3/10
195/195 - 108s - loss: 7.8777e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 108s/epoch - 551ms/step
Epoch 4/10
195/195 - 108s - loss: 6.4710e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 108s/epoch - 555ms/step
Epoch 5/10
195/195 - 108s - loss: 5.2355e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 108s/epoch - 552ms/step
Epoch 6/10
195/195 - 109s - loss: 4.4037e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 109s/epoch - 557ms/step
Epoch 7/10
195/195 - 108s - loss: 3.7371e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 108s/epoch - 555ms/step
Epoch 8/10
195/195 - 107s - loss: 3.1316e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 107s/epoch - 550ms/step
Epoch 9/10
195/195 - 111s - loss: 2.7034e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 111s/epoch - 567ms/step
Epoch 10/10
195/195 - 108s - loss: 2.3181e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 - 108s/epoch - 551ms/step
```

In [ ]:

Data

Files | Connections

Upload one file at a time. All file types accepted. 5 GB max file size.

Drag and drop files here or upload.

Trial.rar
Insert to code

Trial.zip
Insert to code

29°C Partly sunny | ENG | 10:34 19-11-2022

IBM | Service Details - IBM Cloud | Fruit Classifier Model - IBM

jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/46f9fa0f-38c9-49c4-a44a-a1e6f6acf80b

Sites

IBM Watson Studio | Search in your workspaces | Buy | Sanjay Bharath's Account | Tokyo

Projects / Fruit Classification / Fruit Classifier Model

File Edit View Insert Cell Kernel Help | Trusted | Python 3.9

Run | Format: Code

In [3]:
```
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, Dense, Flatten, BatchNormalization, Conv2D, MaxPool2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [7]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='CqO8CnIrUHwELtYhF5-rpY4uurGABI5J86akClkAid7P',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.ap.cloud-object-storage.appdomain.cloud')

bucket = 'fruitclassification-donotdelete-pr-owgell5o83vk7x'
object_key = 'Trial.zip'

streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [8]:
```
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')
file_paths= unzip.namelist()
for path in file_paths:
```

Data

Files | Connections

Upload one file at a time. All file types accepted. 5 GB max file size.

Drag and drop files here or upload.

Trial.rar
Insert to code

Trial.zip
Insert to code

29°C Partly sunny | ENG | 10:34 19-11-2022

IBM | Service Details - IBM Cloud | Fruit Classifier Model - IBM

jp-tok.dataplatform.cloud.ibm.com/analytics/notebooks/v2/46f9fa0f-38c9-49c4-a44a-a1e6f6acf80b

Sites

IBM Watson Studio | Search in your workspaces | Buy | Sanjay Bharath's Account | Tokyo

Projects / Fruit Classification / Fruit Classifier Model

File Edit View Insert Cell Kernel Help | Trusted | Python 3.9

Run | Format Code

```
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [8]:
```
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')
file_paths= unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [10]:
```
train_path= "/home/wsuser/work/Trial Train"
test_path="/home/wsuser/work/Trial Test"
train_batches=ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input) \
    .flow_from_directory(directory=train_path,target_size=(224,224), classes=["Apple","Lychee","Pear","Watermelon"], batch_size=10)
valid_batches=ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input) \
    .flow_from_directory(directory=test_path,target_size=(224,224), classes=["Apple","Lychee","Pear","Watermelon"], batch_size=10)
test_batches=ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input) \
    .flow_from_directory(directory=test_path,target_size=(224,224), classes=["Apple","Lychee","Pear","Watermelon"], batch_size=10, shuffle=False)
```

```
Found 1949 images belonging to 4 classes.
Found 651 images belonging to 4 classes.
Found 651 images belonging to 4 classes.
```

In [11]:
```
model=Sequential([
    Conv2D(filters=32, kernel_size=(3,3), input_shape=(224,224,3), activation="relu", padding="same"),
    MaxPool2D(pool_size=(2,2), strides=2),
    Conv2D(filters=64, kernel_size=(3,3),activation="relu", padding="same"),
    MaxPool2D(pool_size=(2,2), strides=2),
    Flatten(),
    Dense(units=4, activation="softmax")
])
```

In [12]:
```
print(model.summary())
model.compile(optimizer=Adam(learning_rate=0.0001),loss="categorical_crossentropy", metrics=["accuracy"])
model.fit(x=train_batches, validation_data= valid_batches, epochs=10, verbose=2)
model.save("modelfinal.h5")
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

Data

Files | Connections

Upload one file at a time. All file types accepted. 5 GB max file size.

Drag and drop files here or upload.

Trial.rar
Insert to code

Trial.zip
Insert to code

29°C Partly sunny | ENG | 10:34 19-11-2022