

DATE	17 November 2022
TEAM ID	PNT2022TMID08799
PROJECT NAME	AI-POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS

The screenshot shows the IBM Watson Studio interface. The browser address bar indicates the URL: `dataplatfrom.cloud.ibm.com/analytics/notebooks/v2/7e4d4297-5e12-412e-92b4-c84cd04a9777?projectid=1020ea09-671a-44a3-baf7-886a2f746a56&context...`. The interface includes a top navigation bar with 'IBM Watson Studio' and a search bar. Below this, the project path is shown: 'Projects / AI powered nutritional analyzer f... / Model\_Training\_File'. The main area displays a Jupyter Notebook with the following code:

```
In [ ]: IMPORTING LIBRARIES

In [32]: 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 sklearn.metrics import confusion_matrix
from tensorflow.keras.preprocessing.image import ImageDataGenerator

DATA PREPROCESSING

In [33]: train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip= True)
test_datagen=ImageDataGenerator(rescale=1./255)

In [34]: x_train=train_datagen.flow_from_directory(r'D:\BATCH-IBMPROJECT\AI-Powered Nutrition Analyser for Fitness Enthusiasts\Dataset\TRAIN_SET'
, target_size=(64,64),batch_size=32,class_mode='categorical'
)

x_test=test_datagen.flow_from_directory(r'D:\BATCH-IBMPROJECT\AI-Powered Nutrition Analyser for Fitness Enthusiasts\Dataset\TEST_SET'
, target_size=(64,64),batch_size=32,class_mode='categorical'
)

Found 2626 images belonging to 5 classes.
Found 1055 images belonging to 5 classes.
```

IBM | IBM-EPBL/IBM-Project | Service Details - IBM | train the model - IBM | Model\_Training\_File | New Tab | dataplatform.cloud.ibm.com/analytics/notebooks/v2/7e4d4297-5e12-412e-92b4-c84cd04a9777?projectid=1020ea09-671a-44a3-baf7-886a2f746a56&context... | IBM Watson Studio | Search in your workspaces | Buy | Nivya avanthika Senthil's A... | Dallas | NS

Projects / AI powered nutritional analyzer f... / Model\_Training\_File

File Edit View Insert Cell Kernel Help | Not Trusted | Python 3.9

```
Found 2626 images belonging to 5 classes.
Found 1855 images belonging to 5 classes.

In [35]: x_train.class_indices

Out[35]: {'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}

MODEL BUILDING

In [36]: model=Sequential()

In [37]: model=Sequential()

In [38]: classifier=Sequential()
classifier.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation="relu"))
classifier.add(MaxPool2D(pool_size=(2,2)))
classifier.add(Conv2D(32,(3,3),activation="relu"))
classifier.add(MaxPool2D(pool_size=(2,2)))
classifier.add(Flatten())

In [39]: classifier.add(Dense(units=128,activation="relu"))
classifier.add(Dense(units=5,activation="softmax"))

In [40]: classifier.summary()

Model: "sequential_6"
```

IBM | IBM-EPBL/IBM-Project | Service Details - IBM | train the model - IBM | Model\_Training\_File | New Tab | dataplatform.cloud.ibm.com/analytics/notebooks/v2/7e4d4297-5e12-412e-92b4-c84cd04a9777?projectid=1020ea09-671a-44a3-baf7-886a2f746a56&context... | IBM Watson Studio | Search in your workspaces | Buy | Nivya avanthika Senthil's A... | Dallas | NS

Projects / AI powered nutritional analyzer f... / Model\_Training\_File

File Edit View Insert Cell Kernel Help | Not Trusted | Python 3.9

```
In [40]: classifier.summary()

Model: "sequential_6"

Layer (type)                 Output Shape              Param #
-----
conv2d_5 (Conv2D)            (None, 62, 62, 32)       896
max_pooling2d_5 (MaxPooling2 (None, 31, 31, 32)       0
conv2d_6 (Conv2D)            (None, 29, 29, 32)       9248
max_pooling2d_6 (MaxPooling2 (None, 14, 14, 32)       0
flatten_3 (Flatten)          (None, 6272)              0
dense_6 (Dense)              (None, 128)               802944
dense_7 (Dense)              (None, 5)                  645
-----
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0

In [41]: classifier.compile(optimizer="rmsprop",loss="categorical_crossentropy",metrics=["accuracy"])

In [42]: classifier.fit(x_train,steps_per_epoch=82,epochs=20,validation_data=x_test,validation_steps=28)
```

IBM | IBM-EPBL/IBM-Project | Service Details - IBM | train the model - IBM | Model\_Training\_File | New Tab | dataplatform.cloud.ibm.com/... | IBM Watson Studio | Buy | Niviya avanthika Senthil's A... | Dallas | NS

Projects / AI powered nutritional analyzer f... / Model\_Training\_File

File Edit View Insert Cell Kernel Help Not Trusted | Python 3.9

```
In [41]: classifier.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])

In [42]: classifier.fit(x_train, steps_per_epoch=82, epochs=20, validation_data=x_test, validation_steps=28)

Epoch 1/20
82/82 [=====] - 7s 77ms/step - loss: 0.5474 - accuracy: 0.8150 - val_loss: 0.0931 - val_accuracy: 0.9632
Epoch 2/20
82/82 [=====] - 6s 73ms/step - loss: 0.0608 - accuracy: 0.9869 - val_loss: 0.0498 - val_accuracy: 0.9688
Epoch 3/20
82/82 [=====] - 9s 105ms/step - loss: 0.1224 - accuracy: 0.9815 - val_loss: 0.0258 - val_accuracy: 0.9810
Epoch 4/20
82/82 [=====] - 8s 103ms/step - loss: 0.1547 - accuracy: 0.9865 - val_loss: 8.6944e-04 - val_accuracy: 1.0000
Epoch 5/20
82/82 [=====] - 8s 98ms/step - loss: 6.5788e-04 - accuracy: 1.0000 - val_loss: 0.0190 - val_accuracy: 0.9900
Epoch 6/20
82/82 [=====] - 8s 102ms/step - loss: 0.1110 - accuracy: 0.9907 - val_loss: 0.0077 - val_accuracy: 1.0000
Epoch 7/20
82/82 [=====] - 8s 99ms/step - loss: 0.0903 - accuracy: 0.9800 - val_loss: 0.5883 - val_accuracy: 0.8694
Epoch 8/20
82/82 [=====] - 8s 95ms/step - loss: 0.0040 - accuracy: 0.9988 - val_loss: 0.0128 - val_accuracy: 0.9933
Epoch 9/20
82/82 [=====] - 8s 100ms/step - loss: 0.1629 - accuracy: 0.9877 - val_loss: 0.1352 - val_accuracy: 0.9710
Epoch 10/20
82/82 [=====] - 8s 97ms/step - loss: 7.3137e-05 - accuracy: 1.0000 - val_loss: 0.0095 - val_accuracy: 0.9989
Epoch 11/20
82/82 [=====] - 8s 96ms/step - loss: 0.0934 - accuracy: 0.9907 - val_loss: 5.1888e-04 - val_accuracy: 1.0000
Epoch 12/20
82/82 [=====] - 8s 101ms/step - loss: 1.2171e-05 - accuracy: 1.0000 - val_loss: 0.0213 - val_accuracy: 0.9877
Epoch 13/20
```

IBM | IBM-EPBL/IBM-Project | Service Details - IBM | train the model - IBM | Model\_Training\_File | New Tab | dataplatform.cloud.ibm.com/... | IBM Watson Studio | Buy | Niviya avanthika Senthil's A... | Dallas | NS

Projects / AI powered nutritional analyzer f... / Model\_Training\_File

File Edit View Insert Cell Kernel Help Not Trusted | Python 3.9

```
SAVING THE MODEL

In [43]: classifier.save("model.h5")

MODEL PREDICTION

In [44]: from tensorflow.keras.models import load_model
model = load_model("model.h5")
model

Out[44]: <tensorflow.python.keras.engine.sequential.Sequential at 0x1ff9a7cac40>

In [45]: from tensorflow.keras.preprocessing import image

In [46]: img = image.load_img(r"C:\Users\91720\Downloads\Nutrition_Image_Analysis\Dataset\TEST_SET\BANANA\13_100.jpg", target_size = (64, 64))
x = image.img_to_array(img)
x = np.expand_dims(x, axis = 0)

In [47]: prediction = model.predict(x)
index = ['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']
prediction

Out[47]: array([[0., 1., 0., 0., 0.]], dtype=float32)
```

IBM EPBL/IBM-Project Service Details - IBM train the model - IBM Model\_Training\_File New Tab

dataplatfom.cloud.ibm.com/analytics/notebooks/v2/7e4d4297-5e12-412e-92b4-c84cd04a9777?projectid=1020ea09-671a-44a3-baf7-886a2f746a56&context...

IBM Watson Studio Search in your workspaces Buy Nivya avanthika Senthil's A... Dallas NS

Projects / AI powered nutritional analyzer f... / Model\_Training\_File

File Edit View Insert Cell Kernel Help Not Trusted | Python 3.9

In [46]: `img = image.load_img(r"C:\Users\91720\Downloads\Nutrition_Image_Analysis\Dataset\TEST_SET\BANANA\13_100.jpg", target_size = (64, 64))  
x = image.img_to_array(img)  
x = np.expand_dims(x, axis = 0)`

In [47]: `prediction = model.predict(x)  
index = ['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']  
prediction`

Out[47]: `array([[0., 1., 0., 0., 0.]], dtype=float32)`

In [48]: `ind = np.argmax(prediction)`

In [49]: `index[ind]`

Out[49]: `'BANANA'`

In [ ]: