AI POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS

Team ID: PNT2022TMID49087

Build model in ibm cloud:

pwd

Out[38]:

'/home/wsuser/work'

In [39]:

```
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
```

In [40]:

```
# Image Data Agumentation for training set
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2,
zoom_range=0.2, horizontal_flip=True)
```

In [41]:

```
#Image Data Agumentation for testing set
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [42]:

```
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='_gXd4yyBnkqt_bO8wtCPNJIH-yWSXu35FcmBOLwyPXMZ',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'aipoweredrnutritionanalyzerforfitn-donotdelete-pr-fcjabxcyukagdp'
object_key = 'Dataset.zip'

streaming_body_1 = 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 [43]:

```
from io import BytesIO
import zipfile
```

```
unzip = zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_path = unzip.namelist()
for path in file_path:
    unzip.extract(path)
```

In [44]:

```
pwd
```

Out[44]:

```
'/home/wsuser/work'
```

In [45]:

```
import os
filename = os.listdir('/home/wsuser/work/Dataset/TRAIN_SET')
```

In [46]:

```
x_train =
train_datagen.flow_from_directory('/home/wsuser/work/Dataset/TRAI
N_SET',
                    target_size=(64, 64), batch_size=29, color_mode='rgb',
class_mode='sparse')

x_test =
test_datagen.flow_from_directory('/home/wsuser/work/Dataset/TEST
_SET',
                    target_size=(64, 64), batch_size=29, color_mode='rgb',
class_mode='sparse')
Found 4118 images belonging to 5 classes.
Found 929 images belonging to 5 classes.
```

In [47]:

```
print(x_train.class_indices)
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON'
: 4}
```

In [48]:

```
print(x_test.class_indices)
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON'
: 4}
```

In [49]:

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

Out[49]:

```
Counter({0: 995, 1: 1354, 2: 1019, 3: 275, 4: 475})
```

In [50]:

```
import tensorflow as tf

from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
```

In [51]:

```
(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-pyt
hon.tar.gz
170500096/170498071 [==============================] - 6s 0us/
step
170508288/170498071 [==============================] - 6s 0us/
step
```

In [52]:

```
#Creating the model
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))
```

In [53]:

```
model.summary()
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 max_pooling2d (MaxPooling2D  (None, 15, 15, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 4, 4, 64)          36928

 flatten (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
```

In [54]:

```
#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 [==============================] - 38s 24ms/step - loss
: 1.5220 - accuracy: 0.4453 - val_loss: 1.2448 - val_accuracy: 0.5545
Epoch 2/10
1563/1563 [==============================] - 36s 23ms/step - loss
: 1.1736 - accuracy: 0.5831 - val_loss: 1.0946 - val_accuracy: 0.6091
Epoch 3/10
1563/1563 [==============================] - 37s 24ms/step - loss
: 1.0342 - accuracy: 0.6359 - val_loss: 1.0575 - val_accuracy: 0.6246
Epoch 4/10
1563/1563 [==============================] - 37s 24ms/step - loss
: 0.9432 - accuracy: 0.6676 - val_loss: 1.0259 - val_accuracy: 0.6479
Epoch 5/10
1563/1563 [==============================] - 37s 24ms/step - loss
: 0.8719 - accuracy: 0.6937 - val_loss: 0.9573 - val_accuracy: 0.6669
Epoch 6/10
1563/1563 [==============================] - 39s 25ms/step - loss
: 0.8168 - accuracy: 0.7127 - val_loss: 0.9246 - val_accuracy: 0.6754
Epoch 7/10
1563/1563 [==============================] - 36s 23ms/step - loss
: 0.7700 - accuracy: 0.7305 - val_loss: 0.9097 - val_accuracy: 0.6887
Epoch 8/10
```

```
1563/1563 [==============================] - 36s 23ms/step - loss
: 0.7301 - accuracy: 0.7420 - val_loss: 0.9384 - val_accuracy: 0.6822
Epoch 9/10
1563/1563 [==============================] - 37s 24ms/step - loss
: 0.6877 - accuracy: 0.7582 - val_loss: 0.8420 - val_accuracy: 0.7113
Epoch 10/10
1563/1563 [==============================] - 37s 23ms/step - loss
: 0.6567 - accuracy: 0.7691 - val_loss: 0.9031 - val_accuracy: 0.6980
```

In [55]:

```
#Saving our model
model.save('nutrition.h5')
```

In [56]:

```
!tar -zvcf Image-Classification-Model_new.tgz nutrition.h5
nutrition.h5
```

In [57]:

```
ls -1
Dataset/
Image-Classification-Model_new.tgz
nutrition.h5
```

In [68]:

```
!pip install watson-machine-learning-client --upgrade
Requirement already satisfied: watson-machine-learning-client in /opt/
conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.391)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from watson-machine-learning-client)
(2.11.0)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/li
b/python3.9/site-packages (from watson-machine-learning-client) (1.26
.7)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9
/lib/python3.9/site-packages (from watson-machine-learning-client) (0.
8.9)
```

Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botoc

```
ore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0
)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda
/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->wats
on-machine-learning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/
conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk
->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Pytho
n-3.9/lib/python3.9/site-packages (from requests->watson-machine-lea
rning-client) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda
/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-
machine-learning-client) (2.0.4)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Pytho
n-3.9/lib/python3.9/site-packages (from pandas->watson-machine-lear
ning-client) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Pyt
hon-3.9/lib/python3.9/site-packages (from pandas->watson-machine-le
arning-client) (1.20.3)
```

In [69]:

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
   "url": "https://us-south.ml.cloud.ibm.com",
   "apikey": "IaEUv6fA1ZkCDrHhDbXfY58uLsWZ7iuPdpqlVt263zJw"
}
client = APIClient(wml_credentials)
```

In [70]:

```
client = APIClient(wml_credentials)
```

In [75]:

```
def guid_from_space_name(client, space_name):
   space = client.spaces.get_details()
```

```
    return(next(item for item in space['resources'] if
item['entity']['name']== space_name)['metadata']['id'])
                                                                    In [76]:
space_uid = guid_from_space_name(client,'ImageClassification')
print("Space UID =" +space_uid)
Space UID =fcd374ad-0043-4fc9-ac11-00f1e5e014a6
                                                                    In [77]:
client.set.default_space(space_uid)
                                                                   Out[77]:
'SUCCESS'
                                                                    In [78]:
client.software_specifications.list()
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  bas
e
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a
base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d2
88  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  b
ase
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee
base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471
base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  b
ase
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3
f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  ba
se
```

tensorflow_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base
autoai-kb_rt22.2-py3.10 125b6d9a-5b1f-5e8d-972a-b251688ccf40 base
runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base
scikit-learn_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base
default_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base
pytorch-onnx_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base
kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base
pytorch-onnx_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base
tensorflow_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base
spark-mllib_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base
tensorflow_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base
runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base
do_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base
autoai-ts_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base
tensorflow_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base
kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base
pytorch_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base
spark-mllib_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base

pytorch-onnx_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67
e base
spark-mllib_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 b
ase
spark-mllib_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base
autoai-ts_rt22.2-py3.10 396b2e83-0953-5b86-9a55-7ce1628a406f
base
xgboost_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e b
ase
pytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe1
2 base
pytorch-onnx_rt22.2-py3.10 40e73f55-783a-5535-b3fa-0c8b9429143
1 base
default_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 bas
e
autoai-ts_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 b
ase
autoai-obm_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 bas
e
pmml-3.0_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base
spark-mllib_2.4-r_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 b
ase
xgboost_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 ba
se
pytorch-onnx_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b
base
autoai-ts_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde ba
se
spark-mllib_2.4-scala_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5
base
spark-mllib_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base
autoai-obm_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee bas
e

```
spss-modeler_18.1           5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  b
ase
cuda-py3.8                  5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
runtime-22.2-py3.10-xc      5e8cddff-db4a-5a6a-b8aa-2d4af9864dab
base
autoai-kb_3.1-py3.7         632d4b22-10aa-5180-88f0-f52dfb6444d7  ba
se
--------------------------  --------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' p
arameter.
```

In [79]:

```
software_spec_uid=
client.software_specifications.get_uid_by_name("tensorflow_rt22.1-
py3.9")
software_spec_uid
```

Out[79]:

```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

In [80]:

```
model_details =client.repository.store_model(model="Image-
Classification-Model_new.tgz", meta_props={
   client.repository.ModelMetaNames.NAME:"CNN",
   client.repository.ModelMetaNames.TYPE: "tensorflow_2.7",

client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_sp
ec_uid
})
model_id = client.repository.get_model_id(model_details)
```

In [81]:

```
model_id
```

Out[81]:

```
'ef2254ce-def2-44f7-bbe7-cba71a75716b'
```

In [82]:

```
client.repository.download(model_id,'my_model.tar.gz')
```

Successfully saved model content to file: 'my_model.tar.gz'

'/home/wsuser/work/my_model.tar.gz'