Creating and training the CNN model

pwd

'/home/wsuser/work'

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='EHmhit2MD64AQnqYijN7mrXyaEYoh02jLsiuzU5mzGbt',

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 = 'ffdcnnmodelbook-donotdelete-pr-giva0vdmx0opfa'

object\_key = 'forestfiredataset.zip'

streaming\_body\_3 = 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/

from io import BytesIO

import zipfile

unzip=zipfile.ZipFile(BytesIO(streaming\_body\_3.read()),'r')

file\_paths=unzip.namelist()

for path in file\_paths:

unzip.extract(path)

ls

Dataset/ fire-classification-model.tgz forest1.h5

fie-classification-model.tgz fire.h5

Import the libraries

import keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from matplotlib import pyplot as plt

Importing ImageDataGenerator from Keras

# image preprocessing (or) image augmentation

from tensorflow.keras.preprocessing.image import ImageDataGenerator

#import the cnn layers

Defining the Parameters

train\_datagen = ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_flip=True,vertical\_flip=True)

#rescale => rescaling pixel value from 0 to 255 to 0 to 1

#shear\_range=> counter clock wise rotation(anti clock)

test\_datagen = ImageDataGenerator(rescale=1./255)

Applying ImageDataGenerator functionality to train dataset

x\_train = train\_datagen.flow\_from\_directory(r"/home/wsuser/work/Dataset/Dataset/train\_set",

target\_size=(256,256),

batch\_size=32,

class\_mode="binary")

Found 436 images belonging to 2 classes.

Applying ImageDataGenerator functionality to test dataset

x\_test = test\_datagen.flow\_from\_directory(r"/home/wsuser/work/Dataset/Dataset/test\_set",

target\_size=(256,256),

batch\_size=32,

class\_mode="binary")

Found 121 images belonging to 2 classes.

Importing Model Building Libraries

from tensorflow.keras.layers import Convolution2D

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.layers import Flatten

from tensorflow.keras.optimizers import Adam , SGD, RMSprop

x\_train.class\_indices

{'forest': 0, 'with fire': 1}

Intializing the model

model = Sequential()

Adding CNN layers

# add convolution layer

model.add(Convolution2D(32,(3,3),input\_shape=(256,256,3),activation="relu"))

# 32 indicates => no of feature detectors

#(3,3)=> kernel size (feature detector size)

#add max pooling layer

model.add(MaxPooling2D(pool\_size=(2,2)))

#add flatten layer => input to your ANN

model.add(Flatten())

Add Dense layers

#hidden layer

model.add(Dense(units=300,kernel\_initializer="random\_uniform",activation="relu"))

model.add(Dense(units=200,kernel\_initializer="random\_uniform",activation="relu"))

#output layer

model.add(Dense(units=1,kernel\_initializer="random\_uniform",activation="sigmoid"))

Configuring the learning process

#compile the model

model.compile(loss=keras.losses.binary\_crossentropy,optimizer="adam",metrics=['accuracy'])

Summarize the model

model.summary()

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 254, 254, 32) 896

max\_pooling2d (MaxPooling2D (None, 127, 127, 32) 0

)

flatten (Flatten) (None, 516128) 0

dense (Dense) (None, 300) 154838700

dense\_1 (Dense) (None, 200) 60200

dense\_2 (Dense) (None, 1) 201

=================================================================

Total params: 154,899,997

Trainable params: 154,899,997

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Training the model

model.fit(x\_train,steps\_per\_epoch=13,epochs=30,validation\_data=x\_test,validation\_steps=3)

#steps\_per\_epoch = no of training images/batch size

#validation\_steps = no of testing images/batch size

Epoch 1/30

13/13 [==============================] - 49s 4s/step - loss: 1.8251 - accuracy: 0.6485 - val\_loss: 0.2524 - val\_accuracy: 0.8958

Epoch 2/30

13/13 [==============================] - 52s 4s/step - loss: 0.2757 - accuracy: 0.8726 - val\_loss: 0.1387 - val\_accuracy: 0.9479

Epoch 3/30

13/13 [==============================] - 52s 4s/step - loss: 0.3054 - accuracy: 0.8663 - val\_loss: 0.0653 - val\_accuracy: 0.9792

Epoch 4/30

13/13 [==============================] - 52s 4s/step - loss: 0.2152 - accuracy: 0.9084 - val\_loss: 0.0805 - val\_accuracy: 0.9896

Epoch 5/30

13/13 [==============================] - 47s 4s/step - loss: 0.1913 - accuracy: 0.9233 - val\_loss: 0.1705 - val\_accuracy: 0.9375

Epoch 6/30

13/13 [==============================] - 51s 4s/step - loss: 0.2007 - accuracy: 0.9158 - val\_loss: 0.0850 - val\_accuracy: 0.9688

Epoch 7/30

13/13 [==============================] - 51s 4s/step - loss: 0.1476 - accuracy: 0.9455 - val\_loss: 0.0729 - val\_accuracy: 0.9792

Epoch 8/30

13/13 [==============================] - 50s 4s/step - loss: 0.1483 - accuracy: 0.9356 - val\_loss: 0.0579 - val\_accuracy: 0.9792

Epoch 9/30

13/13 [==============================] - 50s 4s/step - loss: 0.1606 - accuracy: 0.9282 - val\_loss: 0.1238 - val\_accuracy: 0.9688

Epoch 10/30

13/13 [==============================] - 50s 4s/step - loss: 0.1764 - accuracy: 0.9158 - val\_loss: 0.1050 - val\_accuracy: 0.9688

Epoch 11/30

13/13 [==============================] - 52s 4s/step - loss: 0.1448 - accuracy: 0.9406 - val\_loss: 0.0601 - val\_accuracy: 0.9792

Epoch 12/30

13/13 [==============================] - 50s 4s/step - loss: 0.1229 - accuracy: 0.9554 - val\_loss: 0.0309 - val\_accuracy: 0.9896

Epoch 13/30

13/13 [==============================] - 53s 4s/step - loss: 0.1220 - accuracy: 0.9579 - val\_loss: 0.0533 - val\_accuracy: 0.9896

Epoch 14/30

13/13 [==============================] - 52s 4s/step - loss: 0.1291 - accuracy: 0.9455 - val\_loss: 0.0525 - val\_accuracy: 0.9792

Epoch 15/30

13/13 [==============================] - 50s 4s/step - loss: 0.1065 - accuracy: 0.9554 - val\_loss: 0.0221 - val\_accuracy: 0.9896

Epoch 16/30

13/13 [==============================] - 50s 4s/step - loss: 0.1161 - accuracy: 0.9554 - val\_loss: 0.0206 - val\_accuracy: 1.0000

Epoch 17/30

13/13 [==============================] - 52s 4s/step - loss: 0.1607 - accuracy: 0.9356 - val\_loss: 0.0258 - val\_accuracy: 0.9896

Epoch 18/30

13/13 [==============================] - 48s 4s/step - loss: 0.1090 - accuracy: 0.9629 - val\_loss: 0.0293 - val\_accuracy: 0.9896

Epoch 19/30

13/13 [==============================] - 51s 4s/step - loss: 0.1500 - accuracy: 0.9332 - val\_loss: 0.0269 - val\_accuracy: 0.9896

Epoch 20/30

13/13 [==============================] - 50s 4s/step - loss: 0.1445 - accuracy: 0.9431 - val\_loss: 0.0187 - val\_accuracy: 1.0000

Epoch 21/30

13/13 [==============================] - 51s 4s/step - loss: 0.1292 - accuracy: 0.9530 - val\_loss: 0.0313 - val\_accuracy: 0.9792

Epoch 22/30

13/13 [==============================] - 50s 4s/step - loss: 0.1079 - accuracy: 0.9554 - val\_loss: 0.0496 - val\_accuracy: 0.9792

Epoch 23/30

13/13 [==============================] - 51s 4s/step - loss: 0.1115 - accuracy: 0.9554 - val\_loss: 0.0274 - val\_accuracy: 0.9792

Epoch 24/30

13/13 [==============================] - 51s 4s/step - loss: 0.0999 - accuracy: 0.9579 - val\_loss: 0.0221 - val\_accuracy: 0.9896

Epoch 25/30

13/13 [==============================] - 52s 4s/step - loss: 0.0801 - accuracy: 0.9752 - val\_loss: 0.0125 - val\_accuracy: 0.9896

Epoch 26/30

13/13 [==============================] - 51s 4s/step - loss: 0.0761 - accuracy: 0.9736 - val\_loss: 0.0234 - val\_accuracy: 0.9792

Epoch 27/30

13/13 [==============================] - 51s 4s/step - loss: 0.0825 - accuracy: 0.9752 - val\_loss: 0.0092 - val\_accuracy: 1.0000

Epoch 28/30

13/13 [==============================] - 52s 4s/step - loss: 0.0738 - accuracy: 0.9703 - val\_loss: 0.0167 - val\_accuracy: 0.9896

Epoch 29/30

13/13 [==============================] - 51s 4s/step - loss: 0.0780 - accuracy: 0.9663 - val\_loss: 0.0024 - val\_accuracy: 1.0000

Epoch 30/30

13/13 [==============================] - 51s 4s/step - loss: 0.1051 - accuracy: 0.9455 - val\_loss: 0.0151 - val\_accuracy: 1.0000

Saving the model

model.save("fire.h5")

IBM Deployment

!pip install watson-machine-learning-client

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: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)

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: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)

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: 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: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)

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: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)

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: 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: 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->botocore<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->watson-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/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-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/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)

Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)

from ibm\_watson\_machine\_learning import APIClient

wml\_credentials={

"url":"https://us-south.ml.cloud.ibm.com",

"apikey":"1AfypwQwqeHikzD7u4LIKT6DMnD-RPDTyYLRBofzNBPp"

}

client=APIClient(wml\_credentials)

client

def guid\_space\_name(client,fire\_deploy):

space=client.spaces.get\_details()

return(next(item for item in space['resources'] if item['entity']['name']==fire\_deploy)['metadata']['id'])

space\_uid=guid\_space\_name(client,'cnn\_fire')

print("Space UID "+space\_uid)

Space UID def3a2d0-3dd4-4f16-9ba5-cb9feb7700a1

client.set.default\_space(space\_uid)

'SUCCESS'

client.software\_specifications.list(200)

------------------------------- ------------------------------------ ----

NAME ASSET\_ID TYPE

default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base

kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base

pytorch-onnx\_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base

scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base

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 base

shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base

tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base

pytorch\_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base

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-dde874a8d67e base

spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base

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 base

pytorch-onnx\_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base

pytorch-onnx\_rt22.2-py3.10 40e73f55-783a-5535-b3fa-0c8b94291431 base

default\_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base

autoai-ts\_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base

autoai-obm\_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base

pmml-3.0\_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base

spark-mllib\_2.4-r\_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base

xgboost\_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base

pytorch-onnx\_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base

autoai-ts\_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base

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 base

spss-modeler\_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base

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 base

pytorch-onnx\_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base

spark-mllib\_2.3-r\_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base

tensorflow\_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base

spss-modeler\_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base

pytorch-onnx\_1.2-py3.6 692a6a4d-2c4d-45ff-a1ed-b167ee55469a base

spark-mllib\_2.3-scala\_2.11 7963efe5-bbec-417e-92cf-0574e21b4e8d base

spark-mllib\_2.4-py37 7abc992b-b685-532b-a122-a396a3cdbaab base

caffe\_1.0-py3.6 7bb3dbe2-da6e-4145-918d-b6d84aa93b6b base

pytorch-onnx\_1.7-py3.7 812c6631-42b7-5613-982b-02098e6c909c base

cuda-py3.6 82c79ece-4d12-40e6-8787-a7b9e0f62770 base

tensorflow\_1.15-py3.6-horovod 8964680e-d5e4-5bb8-919b-8342c6c0dfd8 base

hybrid\_0.1 8c1a58c6-62b5-4dc4-987a-df751c2756b6 base

pytorch-onnx\_1.3-py3.7 8d5d8a87-a912-54cf-81ec-3914adaa988d base

caffe-ibm\_1.0-py3.6 8d863266-7927-4d1e-97d7-56a7f4c0a19b base

runtime-22.2-py3.10-cuda 8ef391e4-ef58-5d46-b078-a82c211c1058 base

spss-modeler\_17.1 902d0051-84bd-4af6-ab6b-8f6aa6fdeabb base

do\_12.10 9100fd72-8159-4eb9-8a0b-a87e12eefa36 base

do\_py3.7 9447fa8b-2051-4d24-9eef-5acb0e3c59f8 base

spark-mllib\_3.0-r\_3.6 94bb6052-c837-589d-83f1-f4142f219e32 base

cuda-py3.7-opence 94e9652b-7f2d-59d5-ba5a-23a414ea488f base

nlp-py3.8 96e60351-99d4-5a1c-9cc0-473ac1b5a864 base

cuda-py3.7 9a44990c-1aa1-4c7d-baf8-c4099011741c base

hybrid\_0.2 9b3f9040-9cee-4ead-8d7a-780600f542f7 base

spark-mllib\_3.0-py38 9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418 base

autoai-kb\_3.3-py3.7 a545cca3-02df-5c61-9e88-998b09dc79af base

spark-mllib\_3.0-py39 a6082a27-5acc-5163-b02c-6b96916eb5e0 base

runtime-22.1-py3.9-do a7e7dbf1-1d03-5544-994d-e5ec845ce99a base

default\_py3.8 ab9e1b80-f2ce-592c-a7d2-4f2344f77194 base

tensorflow\_rt22.1-py3.9 acd9c798-6974-5d2f-a657-ce06e986df4d base

kernel-spark3.2-py3.9 ad7033ee-794e-58cf-812e-a95f4b64b207 base

autoai-obm\_2.0 with Spark 3.0 af10f35f-69fa-5d66-9bf5-acb58434263a base

runtime-22.2-py3.10 b56101f1-309d-549b-a849-eaa63f77b2fb base

default\_py3.7\_opence c2057dd4-f42c-5f77-a02f-72bdbd3282c9 base

tensorflow\_2.1-py3.7 c4032338-2a40-500a-beef-b01ab2667e27 base

do\_py3.7\_opence cc8f8976-b74a-551a-bb66-6377f8d865b4 base

spark-mllib\_3.3 d11f2434-4fc7-58b7-8a62-755da64fdaf8 base

autoai-kb\_3.0-py3.6 d139f196-e04b-5d8b-9140-9a10ca1fa91a base

spark-mllib\_3.0-py36 d82546d5-dd78-5fbb-9131-2ec309bc56ed base

autoai-kb\_3.4-py3.8 da9b39c3-758c-5a4f-9cfd-457dd4d8c395 base

kernel-spark3.2-r3.6 db2fe4d6-d641-5d05-9972-73c654c60e0a base

autoai-kb\_rt22.1-py3.9 db6afe93-665f-5910-b117-d879897404d9 base

tensorflow\_rt22.1-py3.9-horovod dda170cc-ca67-5da7-9b7a-cf84c6987fae base

autoai-ts\_1.0-py3.7 deef04f0-0c42-5147-9711-89f9904299db base

tensorflow\_2.1-py3.7-horovod e384fce5-fdd1-53f8-bc71-11326c9c635f base

default\_py3.7 e4429883-c883-42b6-87a8-f419d64088cd base

do\_22.1 e51999ba-6452-5f1f-8287-17228b88b652 base

autoai-obm\_3.2 eae86aab-da30-5229-a6a6-1d0d4e368983 base

runtime-22.2-r4.2 ec0a3d28-08f7-556c-9674-ca7c2dba30bd base

tensorflow\_rt22.2-py3.10 f65bd165-f057-55de-b5cb-f97cf2c0f393 base

do\_20.1 f686cdd9-7904-5f9d-a732-01b0d6b10dc5 base

pytorch-onnx\_rt22.2-py3.10-edt f8a05d07-e7cd-57bb-a10b-23f1d4b837ac base

scikit-learn\_0.19-py3.6 f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9 base

tensorflow\_2.4-py3.8 fe185c44-9a99-5425-986b-59bd1d2eda46 base

------------------------------- ------------------------------------ ----

software\_space\_uid=client.software\_specifications.get\_uid\_by\_name('tensorflow\_rt22.1-py3.9')

software\_space\_uid

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

ls

Dataset/ fire-classification-model.tgz forest1.h5

fie-classification-model.tgz fire.h5

!tar -zcvf fire-classification-model.tgz fire.h5

fire.h5

model\_details=client.repository.store\_model(model='fire-classification-model.tgz',meta\_props={

client.repository.ModelMetaNames.NAME:"CNN Model Building",

client.repository.ModelMetaNames.TYPE:'tensorflow\_2.7',

client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID:software\_space\_uid

})

model\_id=client.repository.get\_model\_id(model\_details)

model\_id

'babd0250-5274-4923-850c-7fe9ce7e2409'

client.repository.download(model\_id,'fire.tar.gb')

Successfully saved model content to file: 'fire.tar.gb'

'/home/wsuser/work/fire.tar.gb'

ls

Dataset/ fire-classification-model.tgz fire.tar.gb

fie-classification-model.tgz fire.h5 forest1.h5