

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES CLOUD DEPLOYMENT

Date	18 November 2022
Team ID	PNT2022TMID03761
Project Name	Emerging Methods for Early Detection of Forest Fires

CLOUD DEPLOYMENT:

Cloud deployment is the process of deploying an application through one or more hosting models—software as a service (SaaS), platform as a service (PaaS) and/or infrastructure as a service (IaaS)—that leverage the cloud. This includes architecting, planning, implementing and operating workloads on cloud.

CODE:

```
import keras
```

```
import tensorflow
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
import tensorflow
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```

import os, types
import pandas as pd
from boto3.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='3ls6KARqw4K7Icbfhp-X36q5Q5UtIjkPBVHg67294jVf ',
    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 = 'forestfiredetection-donotdelete-pr-bpytmsf9pwiglr'
object_key = 'Dataset.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket,
Key=object_key)['Body']

# Your data file was loaded into a boto3.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_1.read()), 'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)

train_datagen = ImageDataGenerator(rescale=1./255,
                                    shear_range=0.2,
                                    rotation_range=180,
                                    zoom_range=0.2,
                                    horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1./255)

x_train = train_datagen.flow_from_directory(r'./Dataset/train_set/',
                                             target_size=(128, 128),
                                             batch_size=32,
                                             class_mode='binary')

```

Found 436 images belonging to 2 classes.

```

x_test = train_datagen.flow_from_directory(r'./Dataset/test_set/',
                                           target_size=(128, 128),
                                           batch_size=32,
                                           class_mode='binary')

```

Found 121 images belonging to 2 classes.

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Convolution2D, MaxPooling2D,
Flatten

model = Sequential()
model.add(Convolution2D(32, (3,3), input_shape=(128, 128, 3),
activation="relu"))

```

```
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(150,activation="relu"))
model.add(Dense(1, activation="sigmoid"))
model.compile(loss="binary_crossentropy",
              optimizer="adam",
              metrics=["accuracy"])

model.fit(x_train, steps_per_epoch=14, epochs=10, validation_data=x_test,
        validation_steps=4)
```

Epoch 1/10

14/14 [=====] - 24s 2s/step - loss: 2.0141 -
accuracy: 0.7133 - val_loss: 0.1630 - val_accuracy: 0.9421

Epoch 2/10

14/14 [=====] - 22s 2s/step - loss: 0.3240 -
accuracy: 0.8922 - val_loss: 0.1051 - val_accuracy: 0.9835

Epoch 3/10

14/14 [=====] - 22s 2s/step - loss: 0.2306 -
accuracy: 0.9014 - val_loss: 0.1186 - val_accuracy: 0.9421

Epoch 4/10

14/14 [=====] - 21s 2s/step - loss: 0.1938 -
accuracy: 0.9174 - val_loss: 0.0852 - val_accuracy: 0.9752

Epoch 5/10

14/14 [=====] - 22s 1s/step - loss: 0.1953 -
accuracy: 0.9243 - val_loss: 0.1242 - val_accuracy: 0.9339

Epoch 6/10

14/14 [=====] - 21s 2s/step - loss: 0.1797 -
accuracy: 0.9128 - val_loss: 0.0790 - val_accuracy: 0.9835

Epoch 7/10

14/14 [=====] - 21s 1s/step - loss: 0.1688 -
accuracy: 0.9335 - val_loss: 0.0905 - val_accuracy: 0.9421

Epoch 8/10

14/14 [=====] - 21s 1s/step - loss: 0.1727 -
accuracy: 0.9220 - val_loss: 0.1370 - val_accuracy: 0.9256

Epoch 9/10

14/14 [=====] - 22s 2s/step - loss: 0.2078 -
accuracy: 0.9128 - val_loss: 0.0687 - val_accuracy: 0.9917

Epoch 10/10

14/14 [=====] - 22s 2s/step - loss: 0.1622 -
accuracy: 0.9266 - val_loss: 0.0673 - val_accuracy: 0.9835

Save the model

```
model.save("model.h5")
```

```
!tar -zcvf model.tgz model.h5
```

model.h5

Prediction

```
!pip install watson-machine-learning-client
```

Collecting watson-machine-learning-client

Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (53
8 kB)

 538 kB 18.1 MB/s
eta 0:00:01

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: tabulate in /opt/conda/envs/Python-3.9/lib/pytho
n3.9/site-packages (from watson-machine-learning-client) (0.8.9)

Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python
3.9/site-packages (from watson-machine-learning-client) (1.26.7)

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: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)

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

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

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->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)

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: 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: 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: 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: 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)

Installing collected packages: watson-machine-learning-client

Successfully installed watson-machine-learning-client-1.0.391

from ibm_watson_machine_learning **import** APIClient

API_KEY = "3ls6KARqw4K7Icbfhp-X36q5Q5UtIjkPBVHg67294jVf "

```
credentials = {  
    "url": "https://us-south.ml.cloud.ibm.com",  
    "apikey": API_KEY  
}
```

client = APIClient(credentials)

```

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'])

space = client.spaces.get_details()
space
{'resources': []}
space_uid = guid_from_space_name(client, 'Forest fire detection')
print("Space UID: ", space_uid)

Space UID: 34bbs79c-5e6d-556c-9183-3bcb4fe1e0jb
client.set.default_space(space_uid)

'SUCCESS'
software_spec_uid = client.software_specifications.get_uid_by_name("runtime-
22.1-py3.9")
software_spec_uid
'12b83a17-24d8-5082-900f-0ab31fbfd3cb'
model_details = client.repository.store_model(model="model.tgz",
meta_props={
    client.repository.ModelMetaNames.NAME: "CNN",
    client.repository.ModelMetaNames.TYPE: "tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:
software_spec_uid
})

model_id = client.repository.get_model_id(model_details)
client.repository.download(model_id, "model.tar.gz")

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

```


`'/home/wsuser/work/model.tar.gz'`