TRAIN IMAGE CLASSIFICATION MODEL

Date:17 November 2022

Team ID: PNT2022TMID50335

Project Name:Emerging Method For Early Detection Of Forest Fire

```
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 botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

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)

model.save("model.h5")

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

model.h5

Prediction

```
from ibm_watson_machine_learning import APIClient

API_KEY = "5W65wtnV1kus0WCtJ4HEMzw-lVetPUlY_B2Nje3fDq4p"

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: d9308ab8-179a-48da-974b-d986f1649bd5

```
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'