EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

Date	16 November 2022
Team ID	PNT2022TMID10687
Project Name	Emerging Methods for Early Detection of Forest Fires

Importing The ImageDataGenerator Library

pwd

!pip install keras

!pip install tensorflow==1.14.0

import tensorflow

import keras

from keras.preprocessing.image import ImageDataGenerator

Define the parameters/arguments for ImageDataGenerator class

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)

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='Hu0crJth4iJIgd922IJK46d06bVFaEwYc-4rmxAF7-sm',
    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 = 'emergingmethodsforforestfiredetec-donotdelete-pr-meznojcru6qftr'
object_key = 'train_set.zip'

streaming_body_5 = 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/



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='Hu0crJth4iJIgd922IJK46d06bVFaEwYc-4rmxAF7-sm',
    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 = 'emergingmethodsforforestfiredetec-donotdelete-pr-meznojcru6qftr'
object_key = 'test_set.zip'

streaming_body_6 = 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
train_unzip =zipfile.ZipFile(BytesIO(streaming_body_5.read()),'r')
train_file_paths=train_unzip.namelist()
for path in train_file_paths:
    train_unzip.extract(path)
from io import BytesIO
import zipfile
test_unzip =zipfile.ZipFile(BytesIO(streaming_body_6.read()),'r')
test_file_paths=test_unzip.namelist()
for path in test_file_paths:
    test_unzip.extract(path)
pwd
import os
file_path=os.listdir('/home/wsuser/work/train_set')
import os
test_file_path=os.listdir('/home/wsuser/work/test_set')
```

Applying ImageDataGenerator functionality to trainset

```
x_train=train_datagen.flow_from_directory('/home/wsuser/work/train_set',target_size=(128,128),batch_size=32, class_mode='binary')
```

Applying ImageDataGenerator functionality to testset

```
x_test=test_datagen.flow_from_directory('/home/wsuser/work/test_set',target_size=(128,128),batch_size=32, class_mode='binary')
```

Import model building libraries

```
#To define Linear initialisation import Sequential

from keras.models import Sequential

#To add layers import Dense

from keras.layers import Dense

#To create Convolution kernel import Convolution2D

from keras.layers import Convolution2D

#import Maxpooling layer

from keras.layers import MaxPooling2D

#import flatten layer

from keras.layers import Flatten

import warnings

warnings.filterwarnings('ignore')
```

Initializing the model

```
model=Sequential()
```

Add CNN Layer

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

#add maxpooling layer

model.add(MaxPooling2D(pool_size=(2,2)))

#add flatten layer

model.add(Flatten())
```

Add Hidden Layer

```
#add hidden layer

model.add(Dense(150,activation='relu'))

#add output layer

model.add(Dense(1,activation='sigmoid'))
```

Configure the learning process

```
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])
```

Train the model

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

Save The Model

```
model.save("forest1.h5")
```

```
!tar -zcvf forest_fire_detection-model_new.tgz forest1.h5
```

```
ls -1
```

Predictions

```
#import load_model from keras.model
```

```
from keras.models import load_model
```

```
#import image class from keras
```

```
from tensorflow.keras.preprocessing import image #import numpy
```

```
import numpy as np
```

```
#import cv2
```

```
import cv2
```

```
#load the saved model
```

```
model = load_model("forest1.h5")
```

```
img=image.load_img('/home/wsuser/work/test_set/forest/0.48007200_1530881924_final_forest.jpg')
```

```
x=image.img_to_array(img)
```

```
res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)
```

```
#expand the image shape
```

```
x=np.expand_dims(res,axis=0)
```

```
pred= model.predict(x)
```

```
pred
```

```
!pip install twilio
```

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

```
from ibm_watson_machine_learning import APIClient
```

```
uml_credientials = {
```

```
   "url":"https://us-south.ml.cloud.ibm.com",
   "apikey":"JwrUkG_NWgWEoonXcz4EIJSJzXWbH97kolVXGvjt9Apr"
   }
client=APIClient(uml_credientials)
client=APIClient(uml_credientials)
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_uid=guid_from_space_name(client, 'CNN_algorithm')
print("Space-UID="+space_uid)
client.set.default_space(space_uid)
client.software_specifications.list()
software_spec_uid=client.software_specifications.get_uid_by_name("tensorflow_2.4-py3.7-
horovod")
software_spec_uid
model_details=client.repository.store_model(model='forest_fire_detection-
model_new.tgz',meta_props={
   client.repository.ModelMetaNames.NAME:"CNN",
   client.repository.ModelMetaNames.TYPE:"keras_2.2.4",
   client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid}
                                )
model_id=client.repository.get_model_uid(model_details)
model_id
```

OpenCV For Video Processing

```
#import opencv library
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
```

```
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#import playsound package
#from playsound import playsound
#load the saved model
model=load_model("forest1.h5")
video=cv2.VideoCapture(0)
name=['forest','with fire']
Creating An Account In Twilio Service
account_sid='AC7fbd9e1b65a166f13459d8eca7b664cf'
auth_token='8e7e8e6672a8fb0a908ab3137560022d'
client=Client(account_sid,auth_token)
message=client.messages \
.create(
 body='Forest Fire is detected, stay alert',
 from_='+18434385489',
 to='+91 95666 05556'
)
print(message.sid)
Sending Alert Message
from tensorflow.keras.utils import load_img,img_to_array
while(1):
    success, frame= video.read()
    cv2.imwrite("image.jpg",frame)
    img=load_img("image.jpg",target_size=(128,128))
    x=img_to_array(img)
    x=np.expand_dims(x,axis=0)
    predict_x=model.predict(x)
    #classes_x=np.argmax(qqqpredict_x,axis=1)
    #pred=model.predict_classes(x)
```

```
    p=predict_x[0]
    print(predict_x)
    #cv2.putText(frame,"predicted class="+str(name[p]),(100,100),cv2.FONT_HERSHEY_SIMPLEX,1,
(0,0,0), 1)
    pred=model.predict(x)
    if pred[0]==1:
        account_sid='AC7fbd9e1b65a166f13459d8eca7b664cf'
        auth_token='8e7e8e6672a8fb0a908ab3137560022d'
        client=Client(account_sid,auth_token)
        message=client.messages \
        .create(
        body='Forest Fire is detected, stay alert', from_='+18434385489',to='+91 95666 05556')
        print(message.sid)
        print('Fire Detected')
        print('SMS sent!')
    else:
        print('No Danger')
        cv2.imshow("image",frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
video.release()
cv2.destroyAllWindows()
```