EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

CODE LAYOUT, READABILITY, REUSABILITY

Date	17 November 2022
Team ID	PNT2022TMID07050
Project Name	Emerging Methods for Early Detection of Forest Fires

ENTIRE CODE:

```
#Importing Keras libraries
import keras
#Importing ImageDataGenerator from Keras
from matplotlib import pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
#Defining the Parameters
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,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True)
#Applying ImageDataGenerator functionality to train dataset
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/train_set',target_size=(64,64),batch_size=32,class_mode='binary')
Found 442 images belonging to 2 classes.
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
#Applying ImageDataGenerator functionality to test dataset
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/test_set',target_size=(64,64),batch_size=32,class_mode='binary')
Found 121 images belonging to 2 classes.

#to define the linear Initialisation import sequential
from keras.models import Sequential
#to add layers import Dense
from keras.layers import Dense
#to create Convolutional 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()
#Adding CNN Layers
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='r
elu'))
#add maxpooling layers
model.add(MaxPooling2D(pool_size=(2,2)))
#add faltten layer
model.add(Flatten())
#Add Dense layers
#add hidden layers
model.add(Dense(150,activation='relu'))
#add output layer
model.add(Dense(1,activation='sigmoid'))
#configuring the learning process
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["
accuracy"])
#Training the model


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

```
Epoch 1/10
14/14 [==============================] - 143s 10s/step - loss: 0.840
5 - accuracy: 0.6516 - val_loss: 0.3531 - val_accuracy: 0.8430
Epoch 2/10
14/14 [==============================] - 22s 2s/step - loss: 0.2768
- accuracy: 0.8846 - val_loss: 0.1287 - val_accuracy: 0.9752
Epoch 3/10
14/14 [==============================] - 20s 1s/step - loss: 0.2019
- accuracy: 0.9163 - val_loss: 0.1078 - val_accuracy: 0.9669
Epoch 4/10
14/14 [==============================] - 20s 1s/step - loss: 0.2020
- accuracy: 0.9118 - val_loss: 0.0813 - val_accuracy: 0.9917
Epoch 5/10
14/14 [==============================] - 22s 2s/step - loss: 0.1888
- accuracy: 0.9050 - val_loss: 0.0940 - val_accuracy: 0.9752
Epoch 6/10
14/14 [==============================] - 20s 1s/step - loss: 0.1760
- accuracy: 0.9253 - val_loss: 0.0769 - val_accuracy: 0.9835
Epoch 7/10
14/14 [==============================] - 22s 2s/step - loss: 0.1742
- accuracy: 0.9299 - val_loss: 0.0751 - val_accuracy: 0.9917
Epoch 8/10
14/14 [==============================] - 22s 2s/step - loss: 0.1689
- accuracy: 0.9253 - val_loss: 0.0609 - val_accuracy: 0.9917
Epoch 9/10
```

```
14/14 [==============================] - 20s 1s/step - loss: 0.1591
- accuracy: 0.9253 - val_loss: 0.0851 - val_accuracy: 0.9587
Epoch 10/10
14/14 [==============================] - 20s 1s/step - loss: 0.1671
- accuracy: 0.9231 - val_loss: 0.0849 - val_accuracy: 0.9752
#Save the model
model.save("/content/drive/MyDrive/forest1.h5")
#Predictions
#import load model from keras.model
from keras.models import load_model
#import image from keras
from tensorflow.keras.preprocessing import image
import numpy as np
#import cv2
import cv2
#load the saved model
model=load_model("/content/drive/MyDrive/forest1.h5")
img=image.load_img('/content/drive/MyDrive/test_set/with
fire/Forest_fire_MNRF_esize_IMG_6743.jpg')
x=image.img_to_array(img)
res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)
pred=model.predict(x)
pred = int(pred[0][0])
pred
int(pred)
1/1 [==============================] - 0s 102ms/step
pip install twilio
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.d
ev/colab-wheels/public/simple/
Collecting twilio
  Downloading twilio-7.15.3-py2.py3-none-any.whl (1.4 MB)
     |████████████████████████████████| 1.4 MB 5.1 MB/s
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist
-packages (from twilio) (2022.6)
Collecting PyJWT<3.0.0,>=2.0.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/pyt
hon3.7/dist-packages (from twilio) (2.23.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/p
ython3.7/dist-packages (from requests>=2.0.0->twilio) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21
.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->
twilio) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/
python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python
3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.3
from twilio.rest import Client
if pred==0:
  print('Forest fire')
  account_sid='AC4c9a105651d0150d1b85af1bd4cf090c'
  auth_token='d18b90389f18b6069775b89c5c10ca1f'
  client=Client(account_sid,auth_token)
  message=client.messages \
  .create(
      body='forest fire is detected,stay alert',
      #use twilio free number
      from_='+15134660214',
      #to number
      to='+919361632961')
  print(message.sid)
  print("Fire detected")
  print("SMS Sent!")
elif pred==1:
  print('No Fire')
```

```
No Fire
```

```
#Open cv for video processing
```

```
pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.d
ev/colab-wheels/public/simple/
Requirement already satisfied: twilio in /usr/local/lib/python3.7/di
st-packages (7.15.3)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/pyt
hon3.7/dist-packages (from twilio) (2.23.0)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in /usr/local/lib
/python3.7/dist-packages (from twilio) (2.6.0)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist
-packages (from twilio) (2022.6)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/p
ython3.7/dist-packages (from requests>=2.0.0->twilio) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python
3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21
.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->
twilio) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/
python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)
#Creating An Account in Twilio Service
#Sending Alert Message
```

```
from logging import WARNING
#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
```

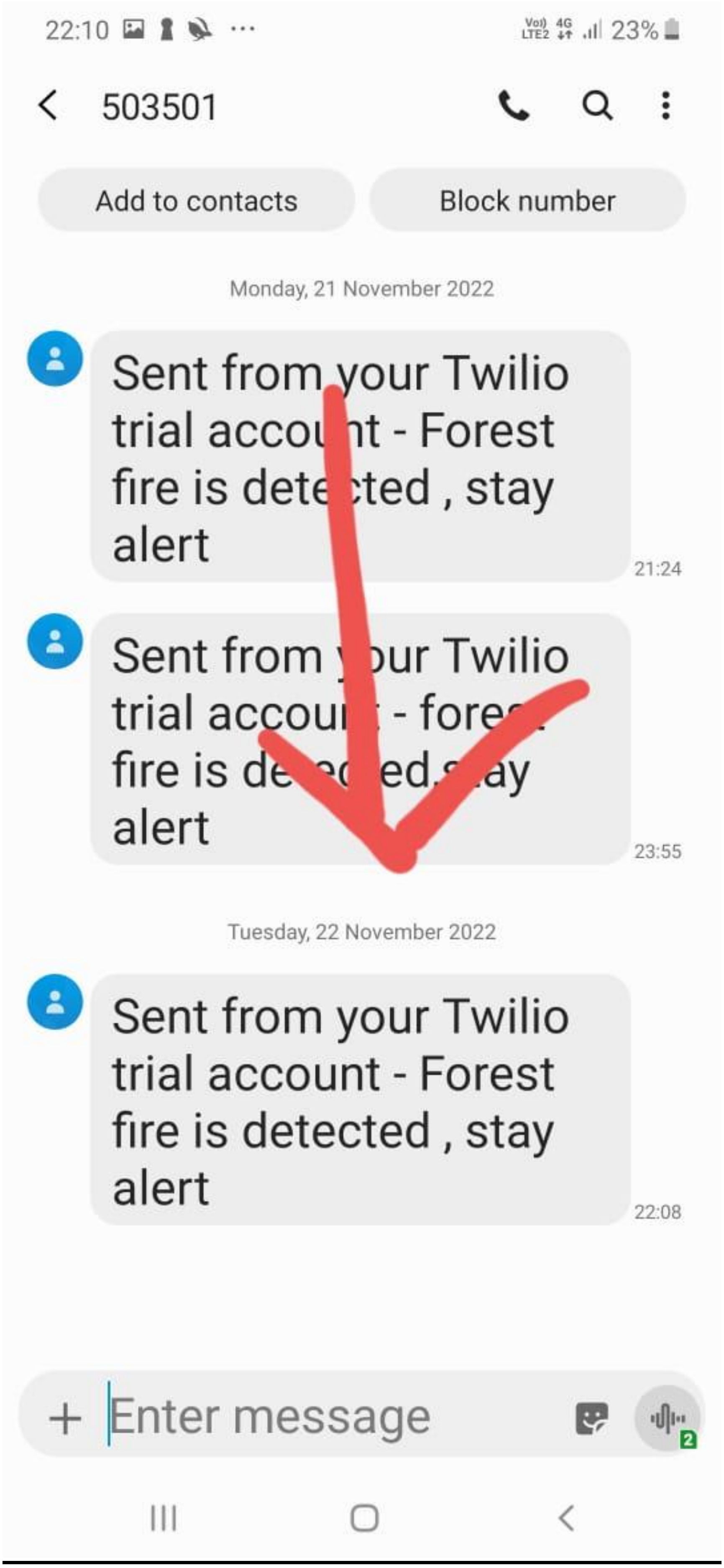
```
import cv2
import numpy as np
```

```
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
import librosa
from tensorflow.keras.preprocessing import image
from keras.models import load_model
# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
cap = cv2.VideoCapture('/FOREST  FIRE.mp4')

# Check if camera opened successfully
if (cap.isOpened()== False):
  print("Error opening video stream or file")

# Read until video is completed
while(cap.isOpened()):
  # Capture frame-by-frame
  ret, frame = cap.read()
  if ret == True:

    cv2_imshow(frame)
    x=image.img_to_array(frame)
    res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
    #expand the image shape
    x=np.expand_dims(res,axis=0)
    model=load_model("/content/drive/MyDrive/forest1.h5")
    pred=model.predict(x)
    pred = int(pred[0][0])
    pred
    int(pred)
    if pred==0:
      print('Forest fire')
      break
    else:
      print("no danger")
      break

# When everything done, release the video capture object
cap.release()

# Closes all the frames
cv2.destroyAllWindows()
```

```
1/1 [==============================] - 0s 67ms/step
Forest fire
from twilio.rest import Client
if pred==0:
  print('Forest fire')
  from twilio.rest import Client
  account_sid='AC4c9a105651d0150d1b85af1bd4cf090c'
  auth_token='d18b90389f18b6069775b89c5c10ca1f'
  client=Client(account_sid,auth_token)
  message=client.messages \
  .create(
      body='forest fire is detected,stay alert',
      #use twilio free number
      from_='+15134660214',
      #to number
      to='+919361632961')
  print(message.sid)
  print("Fire detected")
  print("SMS Sent!")
elif pred==1:
  print('No Fire')
```

```
Forest fire
SM6b18a2fbdd0752525173e0c318202c27
Fire detected
SMS Sent!
```

OUTPUT:



Monday, 21 November 2022

Sent from your Twilio trial account - Forest fire is detected , stay alert 21:24

Sent from your Twilio trial account - forest fire is detected,stay alert 23:55

Tuesday, 22 November 2022

Sent from your Twilio trial account - Forest fire is detected , stay alert 22:08