

video analysis using opencv

#AFTER ANALYSING THE VIDEO THE ALERT MESSAGE IS SENT TO THE CORRESPONDING NUMBER WHEN THE FIRE IS DETECTED

after training and saving the model ..here we used the saved model directly .

the model is loaded and then tested with video having fire and video having no fire

```
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
from twilio.rest import Client
from playsound import playsound
```

WARNING:playsound:playsound is relying on another python subprocess.
Please use `pip install pygobject` if you want playsound to run more efficiently.

```
cap = cv2.VideoCapture('/content/forest with fire.mp4')
if (cap.isOpened()==False):
    print('video streaming or file can not be opened.error occurred ')
while(cap.isOpened()):
    ret,frame=cap.read()
    if ret ==True:
        x=image.img_to_array(frame)
        res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
        x=np.expand_dims(res,axis=0)
        model=load_model('/content/forest1.h5')
        cv2_imshow(frame)
        pred=model.predict(x)
        pred=int(pred[0][0])
        pred
        int(pred)
        if pred==0:
            print('no forest fire')
            break
        else:
            print('forest fire')
            break
cap.release()
cv2.destroyAllWindows()
```



1/1 [=====] - 0s 68ms/step
forest fire

```
cap = cv2.VideoCapture('/content/forest without fire.mp4')
if (cap.isOpened()==False):
    print('video streaming or file can not be opened.error occured ')
while(cap.isOpened()):
    ret,frame=cap.read()
    if ret ==True:
        x=image.img_to_array(frame)
        res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
        x=np.expand_dims(res,axis=0)
        model=load_model('/content/forest1.h5')
        cv2.imshow(frame)
        pred=model.predict(x)
        pred=int(pred[0][0])
        pred
        int(pred)
        if pred==0:
            print('no forest fire')
            break
        else:
            print('forest fire')
            break
cap.release()
cv2.destroyAllWindows()
```



WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7f66fc5b0b90> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

1/1 [=====] - 0s 71ms/step
no forest fire

#SENDING ALERT MESSAGES

```
cap = cv2.VideoCapture('/content/forest with fire.mp4')
if (cap.isOpened()==False):
    print('video streaming or file can not be opened.error occurred ')
while(cap.isOpened()):
    ret,frame=cap.read()
    if ret ==True:
        x=image.img_to_array(frame)
        res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
        x=np.expand_dims(res,axis=0)
        model=load_model('/content/forest1.h5')
        cv2_imshow(frame)
        pred=model.predict(x)
        pred=int(pred[0][0])
        pred
        int(pred)
        if pred==0:
            print('no forest fire')
            break
        else:
            #sending alert message when fire is detected
            account_sid='AC1bde03c3f8ea537b598ad65f2ce62d5d'
            auth_token='7954ac5c2458f8d96d*****'
            client=Client(account_sid,auth_token)
            message=client.messages \
                .create(
                    body='forest fire occurrence is detected,stay safe and
alert',
                    from_='+1606517****',
                    to='+9188381*****'
                )
            print(message.sid)
```

```

        print('forest fire')
        break
cap.release()
cv2.destroyAllWindows()

```



1/1 [=====] - 0s 58ms/step

SM05a0642886e5af91d23a52a779d2fade
forest fire

```

cap = cv2.VideoCapture('/content/forest without fire.mp4')
if (cap.isOpened()==False):
    print('video streaming or file can not be opened.error occurred ')
while(cap.isOpened()):
    ret,frame=cap.read()
    if ret ==True:
        x=image.img_to_array(frame)
        res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
        x=np.expand_dims(res,axis=0)
        model=load_model('/content/forest1.h5')
        cv2_imshow(frame)
        pred=model.predict(x)
        pred=int(pred[0][0])
        pred
        int(pred)
        if pred==0:
            print('no forest fire')
            break
        else:
            #sending alert message when fire is detected
            account_sid='AC1bde03c3f8ea537b598ad65f2ce62d5d'
            auth_token='7954ac5c2458f8d96d*****'
            client=Client(account_sid,auth_token)

```

```
message=client.messages \
.create(
    body='forest fire occurrence is detected, stay safe and
alert',
    from_='+1606517****',
    to='+9188381*****'
)
print(message.sid)
print('forest fire')
break
cap.release()
cv2.destroyAllWindows()
```




1/1 [=====] - 0s 72ms/step
no forest fire