

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m



IMAGE PREPROCESSING

```
#importing keras library
import keras
```

```
#importing the image data generator
from matplotlib import pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
```

```
#Defining the parameter for image generator class
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, rotation_range=180, zoom_range=0.2,
test_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, rotation_range=180, zoom_range=0.2,

```

```
#Applying image data generator functionality to trainset
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/train_set',
target_size=(150,150), batch_size=32, class_mode='binary')
```

Found 436 images belonging to 2 classes.

```
#Applying image data generator functionality to trainset
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/test_set',
target_size=(150,150), batch_size=32, class_mode='binary')
```

Found 121 images belonging to 2 classes.

MODEL BUILDING

```
#To define linear intialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To creat Convolution kernal 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 convolution layer
model.add(Convolution2D(32,(3,3),input_shape=(150,150,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add convolution layer
model.add(Convolution2D(64,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add convolution layer
model.add(Convolution2D(128,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add convolution layer
model.add(Convolution2D(128,(3,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add flatten layer
model.add(Flatten())

model.add(Dense(512,activation='relu'))
model.add(Dense(1,activation='sigmoid'))

#configuring the learning process
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])

```

```

#Training the model
r=model.fit(x_train,epochs=10,validation_data=x_test)

```

```

↳ Epoch 1/10
14/14 [=====] - 20s 1s/step - loss: 0.5653 - accuracy: 0.754
Epoch 2/10
14/14 [=====] - 20s 1s/step - loss: 0.3740 - accuracy: 0.871
Epoch 3/10
14/14 [=====] - 18s 1s/step - loss: 0.1926 - accuracy: 0.916
Epoch 4/10
14/14 [=====] - 18s 1s/step - loss: 0.2402 - accuracy: 0.899
Epoch 5/10
14/14 [=====] - 21s 2s/step - loss: 0.1809 - accuracy: 0.931
Epoch 6/10
14/14 [=====] - 18s 1s/step - loss: 0.1658 - accuracy: 0.946
Epoch 7/10
14/14 [=====] - 19s 1s/step - loss: 0.1380 - accuracy: 0.949
Epoch 8/10
14/14 [=====] - 18s 1s/step - loss: 0.1262 - accuracy: 0.954
Epoch 9/10
14/14 [=====] - 19s 1s/step - loss: 0.1321 - accuracy: 0.949
Epoch 10/10
14/14 [=====] - 18s 1s/step - loss: 0.1256 - accuracy: 0.956

```

```

#save the model
model.save("forestalert.h5")

```

```

#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/forestalert.h5")
img=image.load_img('/content/drive/MyDrive/Dataset/test_set/with fire/FORESTFIRE (1).jpg')
x=image.img_to_array(img)
res=cv2.resize(x,dsize=(150,150),interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)

#predictions
pred=model.predict(x)
pred = int(pred[0][0])
pred
int(pred)

```

```

1/1 [=====] - 0s 221ms/step
1

```

VIDEO ANALYSIS

```
pip install twilio
```

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/r
Collecting twilio
  Downloading twilio-7.15.3-py2.py3-none-any.whl (1.4 MB)
    | 1.4 MB 33.9 MB/s
Collecting PyJWT<3.0.0,>=2.0.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from twilio)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from twilio)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from twilio)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from twilio)
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.3

```

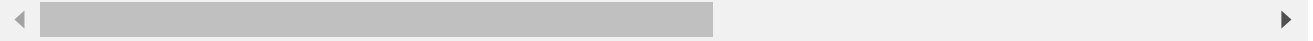
```
pip install playsound
```

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/r
Collecting playsound
  Downloading playsound-1.3.0.tar.gz (7.7 kB)
Building wheels for collected packages: playsound
  Building wheel for playsound (setup.py) ... done

```

```
Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7035 st
Stored in directory: /root/.cache/pip/wheels/ba/f8/bb/ea57c0146b664dca3a0ada4199b0e
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0
```



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

```
pip install pygobject
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: pygobject in /usr/lib/python3/dist-packages (3.26.1)
```



```
pip install pydub
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub
Successfully installed pydub-0.25.1
```



```
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

#OPENCV2
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 google.colab import drive
# 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(r'/content/drive/MyDrive/forest fire.mp4')

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

```
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,dsiz=(150,150),interpolation=cv2.INTER_CUBIC)
        #expand the image shape
        x=np.expand_dims(res,axis=0)
        model=load_model(r"/content/forestalert.h5")
        pred=model.predict(x)
        pred = int(pred[0][0])
        pred
        int(pred)
        if pred==1:
            print('Forest fire')
            #sending alert messages
            from twilio.rest import Client
            from playsound import playsound
            account_sid = 'ACab5b7ac22466b88a9cda7cf5414b750a'
            auth_token = 'c9c95130eade17e5e3d3f936283bef7a'
            client=Client(account_sid,auth_token)
            message=client.messages \
                .create(
                body='Danger!Stay alert,Forest fire is detected',
                #use twilio free number
                from_='+17088477470',
                #to number
                to='+918825826199')
            print(message.sid)
            print("Fire detected")
            print("SMS Sent!")
            # import required module
            break
        elif pred==0:
            print("no danger")
            break

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

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



1/1 [=====] - 0s 121ms/step

Forest fire

SM2e96d5362748c94fa32e942f3246f2bf

Fire detected

SMS Sent!

[Colab paid products](#) - [Cancel contracts here](#)

✓ 12s completed at 3:37 PM

● ✕