

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

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



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=0.1,
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
#Applying image data generator functionality to train set
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/train_set',
target_size=(128,128),batch_size=32,class_mode='binary')
```

Found 436 images belonging to 2 classes.

```
#Applying image data generator functionality to test set
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/test_set',
target_size=(128, 128),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=(128,128,3),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 [=====] - 234s 16s/step - loss: 0.6142 - accuracy: 0.
Epoch 2/10
14/14 [=====] - 17s 1s/step - loss: 0.3169 - accuracy: 0.86
Epoch 3/10
14/14 [=====] - 19s 1s/step - loss: 0.4177 - accuracy: 0.87
Epoch 4/10
14/14 [=====] - 17s 1s/step - loss: 0.3900 - accuracy: 0.86
Epoch 5/10
14/14 [=====] - 17s 1s/step - loss: 0.2594 - accuracy: 0.89
Epoch 6/10
14/14 [=====] - 19s 1s/step - loss: 0.1766 - accuracy: 0.92
Epoch 7/10
14/14 [=====] - 17s 1s/step - loss: 0.1475 - accuracy: 0.94
Epoch 8/10
14/14 [=====] - 19s 1s/step - loss: 0.1356 - accuracy: 0.94
Epoch 9/10
14/14 [=====] - 17s 1s/step - loss: 0.1492 - accuracy: 0.93
Epoch 10/10
14/14 [=====] - 19s 1s/step - loss: 0.1470 - accuracy: 0.94

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

VIDEO ANALYSIS

```
pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Collecting twilio
  Downloading twilio-7.15.3-py2.py3-none-any.whl (1.4 MB)
    |████████████████████████████████████████| 1.4 MB 17.6 MB/s
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)
Collecting PyJWT<3.0.0,>=2.0.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from PyJWT)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from PyJWT)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from PyJWT)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from PyJWT)
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.3
```

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

```
#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/Uttarakhand_forest_fire.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(x,axis=0)
```

```
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
import tensorflow
from tensorflow.keras.preprocessing import image
from keras.models import load_model
from google.colab import drive
from google.colab.patches import cv2_imshow

```

WITH FIRE

```

# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
video = cv2.VideoCapture(r'/content/drive/MyDrive/Wild fire.mp4')
name=['forest','with fire']

while(1):
    success,frame = video.read()
    cv2.imwrite("image.jpg",frame)
    img = tensorflow.keras.utils.load_img("image.jpg",target_size = (128,128))
    x = image.img_to_array(img)
    x = np.expand_dims(x,axis = 0)
    pred = model.predict(x)
    pred = pred[0][0]
    if pred > 0.5:
        pred = 1
    else :
        pred = 0
    print(pred)
    cv2.putText(frame,"predicted class = "+str(name[pred]),(100,100),cv2.FONT_HERSHEY_SIMPLEX
    if pred==1:
        account_sid = 'ACab5b7ac22466b88a9cda7cf5414b750a'
        auth_token = 'c9c95130eade17e5e3d3f936283bef7a'
        client = Client(account_sid, auth_token)
        message = client.messages \
            .create(
                body='Forest Fire is detected,Stay alert',
                from_='+17[REDACTED]',
                to='+918[REDACTED]')
        print(message.sid)
        print("Fire detected")
        print("SMS Sent!")
        cv2_imshow(frame)
        break
    else:
        print("No Danger")
        break
    cv2_imshow(frame)
    if cv2.waitKey(1) & 0xFF == ord('a'):

```

```
break  
video.release()  
cv2.destroyAllWindows()
```

```
1/1 [=====] - 0s 15ms/step
```

```
1
```

```
SM5f311301d5eb273b1cc7685535fde506
```

```
Fire detected
```

```
SMS Sent!
```



WITHOUT FIRE

```

# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
video = cv2.VideoCapture(r'/content/drive/MyDrive/Forest.mp4')

while(1):
    success,frame = video.read()
    cv2.imwrite("image.jpg",frame)
    img = tensorflow.keras.utils.load_img("image.jpg",target_size = (128,128))
    x = image.img_to_array(img)
    x = np.expand_dims(x,axis = 0)
    pred = model.predict(x)
    pred = pred[0][0]
    if pred > 0.5:
        pred = 1
    else :
        pred = 0
    print(pred)
    cv2.putText(frame,"predicted class = "+str(name[pred]),(100,100),cv2.FONT_HERSHEY_SIMPLEX
    if pred==1:
        account_sid = 'ACab5b7ac22466b88a9cda7cf5414b750a'
        auth_token = 'c9c95130eade17e5e3d3f936283bef7a'
        client = Client(account_sid, auth_token)
        message = client.messages \
            .create(
                body='Forest Fire is detected,Stay alert',
                from_='+17[REDACTED]',
                to='+918[REDACTED]')
        print(message.sid)
        print("Fire detected")
        print("SMS Sent!")
        cv2.imshow(frame)
        break
    else:
        print("No Danger")
        cv2.imshow(frame)
        break
    if cv2.waitKey(1) & 0xFF == ord('a'):
        break
    video.release()
cv2.destroyAllWindows()

```



1/1 [=====] - 0s 15ms/step
0
No Danger



[Colab paid products](#) - [Cancel contracts here](#)1m

✓ 2s

completed at 12:08 AM

● ✕