

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

#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, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, rotation_range=180, zoom_range=0.2, horizontal_flip=True)

#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')

#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')
```

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)

#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/Uttarakhand_forest_f
x=image.img_to_array(img)
res=cv2.resize(x,dsi= (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)
```

VIDEO ANALYSIS

```
pip install twilio
```

```

from twilio.rest import Client

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
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 without 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=(150,150),interpolation=cv2.INTER_CUBIC)
        #expand the image shape
        x=np.expand_dims(res,axis=0)
        model=load_model(r"/content/forestuttar.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()
```

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

