In [8]:
```
import keras
from keras.preprocessing.image import ImageDataGenerator
```

In [14]:
```
from keras.models import load_model
from keras.layers import Lambda
import tensorflow as tf
```

In [22]:
```
tf.keras.preprocessing.image_dataset_from_directory(
    directory="C:\\Users\\Akash\\Downloads\\Dataset",
    labels="inferred",
    label_mode="int",
    class_names=None,
    color_mode="rgb",
    batch_size=32,
    image_size=(256, 256),
    shuffle=True,
    seed=None,
    validation_split=None,
    subset=None,
    interpolation="bilinear",
    follow_links=False,
    crop_to_aspect_ratio=False,
)
```

```
Found 558 files belonging to 1 classes.
```

Out[22]:
```
<BatchDataset element_spec=(TensorSpec(shape=(None, 256, 256, 3), dtype=tf.fl
oat32, name=None), TensorSpec(shape=(None,), dtype=tf.int32, name=None))>
```

In [58]:
```
tf.keras.preprocessing.image.load_img(
    path="C:\\Users\\Akash\\Downloads\\Dataset\\Dataset\\train_set\\forest\\wi
)
```

Out[58]:

In [65]:
```
from numpy import *
image = tf.keras.preprocessing.image.load_img("C:\\Users\\Akash\\Downloads\\Da
input_arr = tf.keras.preprocessing.image.img_to_array(image)
input_arr = np.array([input_arr])  # Convert single image to a batch.
predictions = image.predict(input_arr)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_64712\1459244112.py in <module>
      1 from numpy import *
----> 2 image = tf.keras.preprocessing.image.load_img("C:\\Users\\Akash\\Down
loads\\Dataset\\Dataset\\train_set\\forest\\with_fire (1).gif")
      3 input_arr = tf.keras.preprocessing.image.img_to_array(image)
      4 input_arr = np.array([input_arr])  # Convert single image to a batch.
      5 predictions = image.predict(input_arr)

C:\ProgramData\Anaconda3\lib\site-packages\PIL\Image.py in __getattr__(self,
name)
    515             deprecate("Image categories", 10, "is_animated", plural=T
rue)
    516             return self._category
--> 517         raise AttributeError(name)
    518 
    519     @property

AttributeError: keras
```

In [10]:
```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [21]:
```
#: Applying ImageDataGenerator functionality to trainset.
x_train = train_datagen.flow_from_directory(r'C:\Users\Akash\Downloads\Dataset
                                            target_size = (128,128),
                                            batch_size = 32,
                                            class_mode= 'binary')
```

```
Found 436 images belonging to 2 classes.
```

In [22]:
```
x_test = test_datagen.flow_from_directory(r'C:\Users\Akash\Downloads\Dataset\D
                                          target_size = (128,128),
                                          batch_size = 32,
                                          class_mode= 'binary')
```

```
Found 121 images belonging to 2 classes.
```

In [23]:
```
from keras.models import Sequential
from keras.layers import Convolution2D,MaxPooling2D,Dense,Flatten
import warnings
warnings.filterwarnings('ignore')
```

In [24]:
```
model = Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=256,activation='relu'))
model.add(Dense(units=1,activation='sigmoid'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0         
 )                                                               
                                                                 
 flatten (Flatten)           (None, 127008)            0         
                                                                 
 dense (Dense)               (None, 256)               32514304  
                                                                 
 dense_1 (Dense)             (None, 1)                 257       
                                                                 
=================================================================
Total params: 32,515,457
Trainable params: 32,515,457
Non-trainable params: 0
_________________________________________________________________
```

In [28]:
```
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy','mse'])
```

In [31]:
```
model.fit_generator (x_train, steps_per_epoch=14,
                     epochs=10, validation_data=x_test,
                     validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 60s 4s/step - loss: 3.7004 - accurac
y: 0.6674 - mse: 0.2822 - val_loss: 0.4052 - val_accuracy: 0.9174 - val_mse:
0.0760
Epoch 2/10
14/14 [==============================] - 42s 3s/step - loss: 0.3510 - accurac
y: 0.8739 - mse: 0.0887 - val_loss: 0.2228 - val_accuracy: 0.9587 - val_mse:
0.0375
Epoch 3/10
14/14 [==============================] - 46s 3s/step - loss: 0.2168 - accurac
y: 0.9243 - mse: 0.0582 - val_loss: 0.1112 - val_accuracy: 0.9587 - val_mse:
0.0278
Epoch 4/10
14/14 [==============================] - 35s 3s/step - loss: 0.1760 - accurac
y: 0.9358 - mse: 0.0494 - val_loss: 0.0607 - val_accuracy: 0.9587 - val_mse:
0.0197
Epoch 5/10
14/14 [==============================] - 37s 3s/step - loss: 0.1988 - accurac
y: 0.9128 - mse: 0.0621 - val_loss: 0.0753 - val_accuracy: 0.9752 - val_mse:
0.0229
Epoch 6/10
14/14 [==============================] - 37s 3s/step - loss: 0.1705 - accurac
y: 0.9197 - mse: 0.0540 - val_loss: 0.0659 - val_accuracy: 0.9752 - val_mse:
0.0203
Epoch 7/10
14/14 [==============================] - 36s 3s/step - loss: 0.1686 - accurac
y: 0.9220 - mse: 0.0526 - val_loss: 0.0701 - val_accuracy: 0.9752 - val_mse:
0.0214
Epoch 8/10
14/14 [==============================] - 37s 3s/step - loss: 0.1564 - accurac
y: 0.9381 - mse: 0.0493 - val_loss: 0.0773 - val_accuracy: 0.9752 - val_mse:
0.0238
Epoch 9/10
14/14 [==============================] - 47s 3s/step - loss: 0.1739 - accurac
y: 0.9358 - mse: 0.0507 - val_loss: 0.0990 - val_accuracy: 0.9752 - val_mse:
0.0273
Epoch 10/10
14/14 [==============================] - 48s 3s/step - loss: 0.1718 - accurac
y: 0.9266 - mse: 0.0523 - val_loss: 0.0545 - val_accuracy: 0.9835 - val_mse:
0.0162
```

Out[31]:
```
<keras.callbacks.History at 0x161007b09d0>
```

In [32]:
```
model.save("forest1.h5")
```

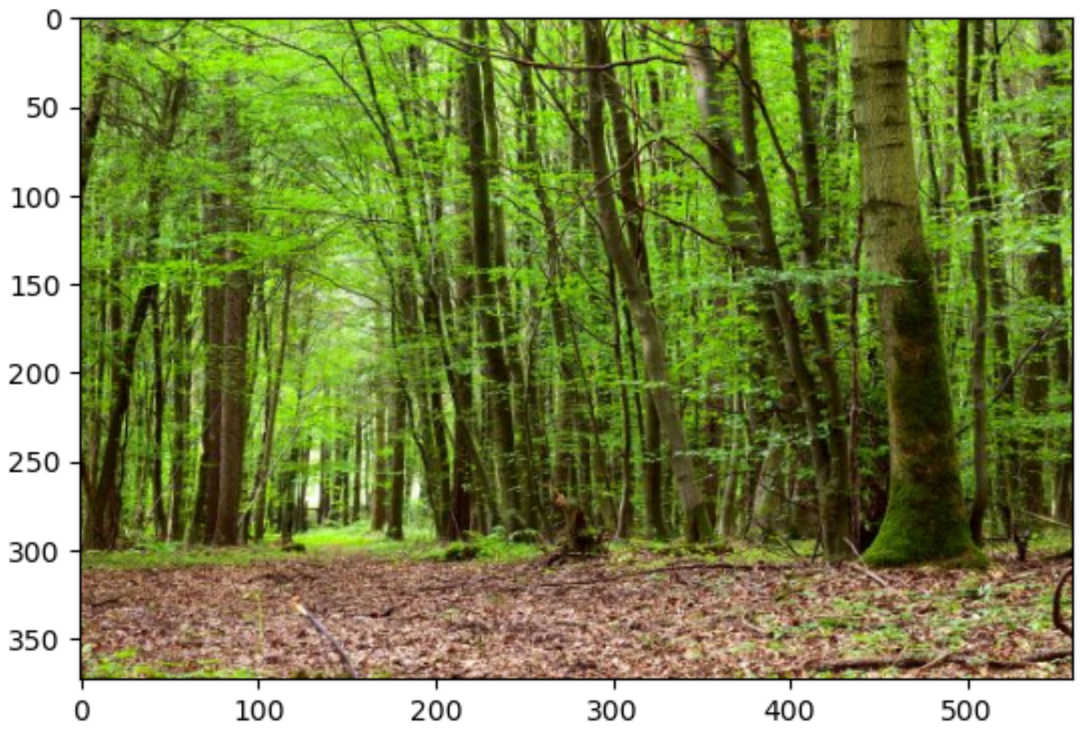
In [81]:
```
#import load_model from keras.model
import matplotlib.pyplot as plt
from keras.models import load_model
#import image class from keras
from keras.preprocessing import image
#import numpy
import numpy as np
from PIL import Image
#import cv2
import cv2
from PIL import Image
from keras.utils import img_to_array
```

In [82]:
```
model = load_model("forest1.h5")
```

In [83]:
```
def prediction(img_path):
    i = cv2.imread(img_path)
    i = cv2.cvtColor(i, cv2.COLOR_BGR2RGB)
    img = Image.open(img_path)
    img = img.resize((128,128))
    x = img_to_array(img)
    x = np.expand_dims(x,axis=0)
    pred = model.predict(x)
    plt.imshow(i)
    print("%s"%("FOREST FIRE DETECTED! SMS SENT!" if pred==[[1.]] else "NO FOR
```

In [84]:
```
prediction(r'C:\Users\Akash\Downloads\Dataset\Dataset\test_set\forest\beech_oa
```

```
1/1 [==============================] - 0s 215ms/step
NO FOREST FIRE DETECTED
```

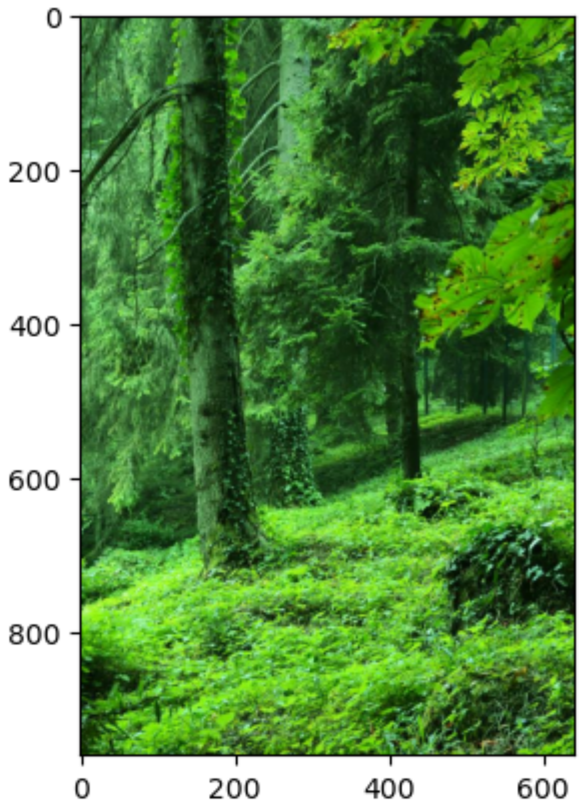


```
In [85]: prediction(r'C:\Users\Akash\Downloads\Dataset\Dataset\test_set\forest\europesl
```

```
1/1 [==============================] - 0s 112ms/step
NO FOREST FIRE DETECTED
```

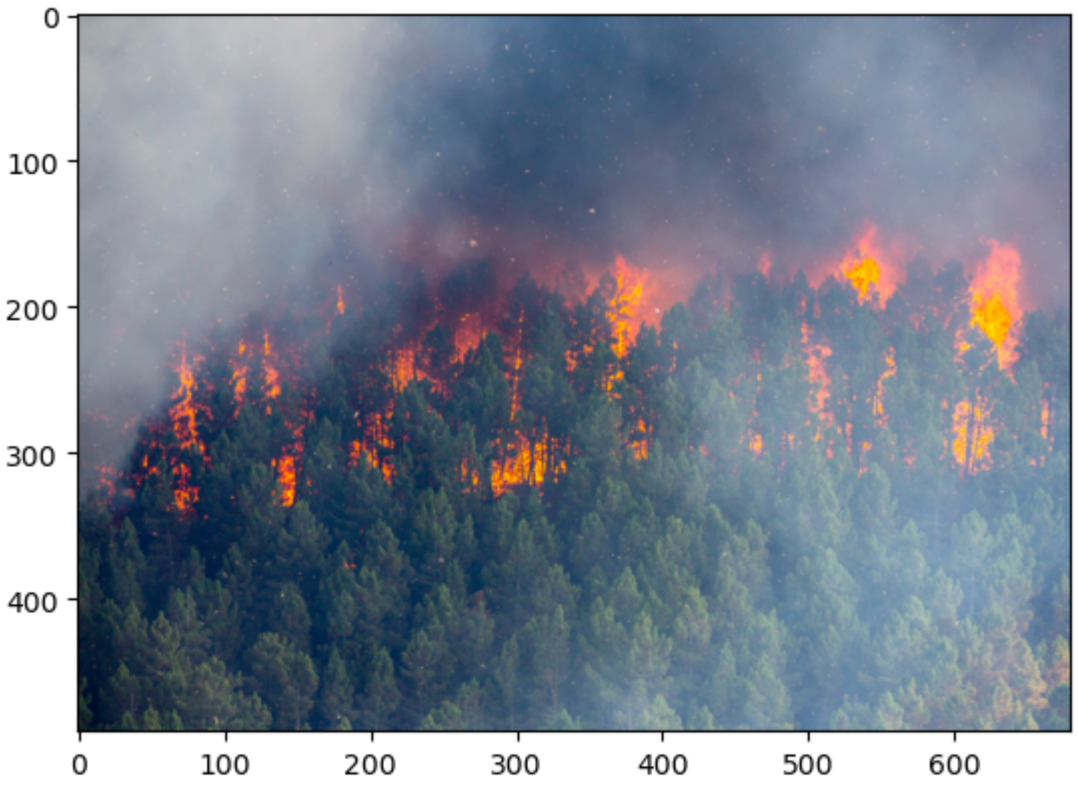
```
In [86]: prediction(r'C:\Users\Akash\Downloads\Dataset\Dataset\test_set\forest\55967210
```

```
1/1 [==============================] - 0s 92ms/step
NO FOREST FIRE DETECTED
```



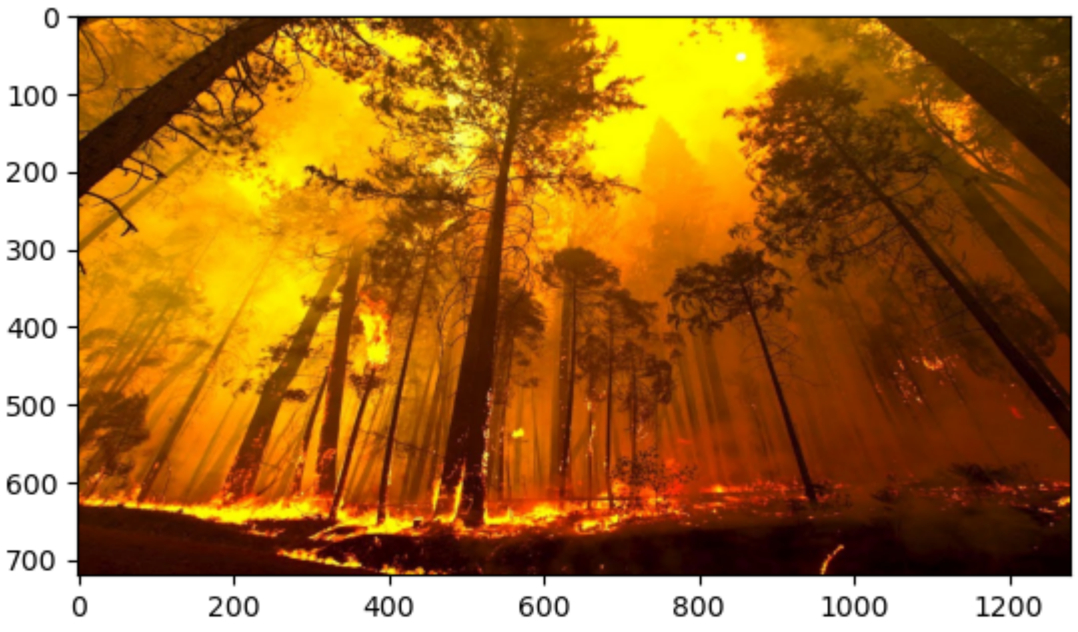
In [87]:
```
prediction(r'C:\Users\Akash\Downloads\Dataset\Dataset\test_set\with fire\in_fo
```

```
1/1 [==============================] - 0s 112ms/step
FOREST FIRE DETECTED! SMS SENT!
```



In [88]:
```
prediction(r'C:\Users\Akash\Downloads\Dataset\Dataset\test_set\with fire\maxre
```

```
1/1 [==============================] - 0s 115ms/step
FOREST FIRE DETECTED! SMS SENT!
```



In [ ]:
```
import cv2
import os
import numpy as np
from tensorflow.keras.utils import load_img,img_to_array
from tensorflow.keras.models import load_model
from twilio.rest import Client
import getpass
from playsound import playsound
```

In [ ]:
```
msg_sent = False
model = load_model(r'forest1.h5')
#define video
video = cv2.VideoCapture("C:\\Users\\Akash\\Downloads\\forest fire vedio.mp4")
#define the featues
name = ['forest', 'with fire']
```

In [ ]:
```
while(1):
success, frame = video.read()
cv2.imwrite("C:\\Users\\Akash\\Downloads\\Dataset\\Dataset\\test_set\\with fir
img = image.load_img("C:\\Users\\Akash\\Downloads\\Dataset\\Dataset\\test_set\
x = image.img_to_array(img)
x = np.expand_dims(x,axis= 0)
pred = model.predict_classes(x)
p = pred[0]
print(pred)
cv2.putText(frame, "predicted class = "+str(name[p]), (100,100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)
```

In [ ]:
```
pred = model.predict_classes(x)
if pred[0]==1:
#twilio account ssid
account_sid = 'AC17385ec6719b077cd7b11729f97ffae0'
#twilio account authentication token
auth_token= '4ef4b6bc05abacd88b778518cd1aaba8'
client = Client (account_sid, auth_token)
message = client.messages \
.create(
body='Forest Fire is detected, stay alert',
#use twilio free number
from_=' +16802196438',
to='+919025764607')
print(message.sid)
print('Fire Detected')
print ('SMS sent!')
playsound(I'C:\\Users\\Akash\\Downloads\\Fire alarm (Message Tone).mp3')
else:
print("No Danger") #break
cv2.imshow("C:\\Users\\Akash\\Downloads\\Dataset1\\Dataset\\test_set\\with fir
if cv2.waitKey(1) & 0xFF== ord('a'):
break
video.release()
cv2.destroyAllWindows()
```

```
In [ ]:
```