In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
```

In [3]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dense,Flatten
```

Loading Dataset

In [4]:
```
X_traincsv=pd.loadtxt('input.csv',delimiter=',')
Y_traincsv=pd.loadtxt('labels.csv',delimiter=',')

X_testcsv=pd.loadtxt('input_test.csv',delimiter=',')
Y_testcsv=pd.loadtxt('labels_test.csv',delimiter=',')
```

In [5]:
```
X_train.shape
```

Out[5]:
```
(2000, 30000)
```

In [6]:
```
Y_train.shape
```

Out[6]:
```
(2000,)
```

In [7]:
```
X_test.shape
```

Out[7]:
```
(400, 30000)
```

In [8]:
```
Y_test.shape
```

Out[8]:
```
(400,)
```

In [9]:
```
X_train=X_train.reshape(len(X_train),100,100,3)
Y_train=Y_train.reshape(len(Y_train),1)

X_test=X_test.reshape(len(X_test),100,100,3)
Y_test=Y_test.reshape(len(Y_test),1)
```

In [10]:
```
# now as our data size is very high our aim is to reduce the size.
```

In [11]:
```
X_train=X_train/255.0
X_test=X_test/255.0
```

In [12]:
```
X_train
```

Out[12]:
```
array([[[0.14509804, 0.15294118, 0.08089922],
        [0.10196078, 0.09411765, 0.03529412],
        [0.11372233, 0.09803922, 0.03529412],
        ...,
        [0.22352941, 0.17254902, 0.13725490 ],
        [0.21176471, 0.16470588, 0.12549882],
        [0.25490196, 0.2 , 0.16470588]],

       [[0.17647059, 0.16862745, 0.10980392],
        [0.10980392, 0.09803922, 0.03137255],
        [0.20392157, 0.15686275, 0.09411765],
        ...,
        [0.21176471, 0.16470588, 0.12549882],
        [0.21176471, 0.16470588, 0.12549882],
        [0.23529412, 0.18431373, 0.14901961],
        ...,
        [0.20392157, 0.15686275, 0.10196078],
        [0.12549882, 0.09411765, 0.03529412],
        [0.27058824, 0.21176471, 0.17254902 ],

        [0.21176471, 0.15686275, 0.11372549],
        [0.21960784, 0.16470588, 0.12156863],
        [0.21372549, 0.17647059, 0.13333333],
        ...,
        [0.07843137, 0.15294118, 0 ],
        [0.07843137, 0.15294118, 0 ],
        [0.08235294, 0.15686275, 0.01176471],
        [0.08235294, 0.15686275, 0.01176471],
        [0.23529412, 0.23529412, 0.17254902 ],
        [0.20411765, 0.17843137, 0.10980392],
        [0.20235294, 0.20666667, 0.16862745]],

       [[0.21764706, 0.20196078, 0.17254902],
        [0.49411765, 0.5882353, 0.36470431],
        [0.57254902, 0.68235308, 0.44705882],
        ...,
        [0.262745 , 0.24705882, 0.14901961],
        [0.30588235, 0.29019608, 0.15114686],
        [0.22745098, 0.21176471, 0.11372549]],

       [[0.51372549, 0.56196078, 0.52941176],
        [0.51372549, 0.56196078, 0.52941176],
        [0.7647059, 0.75294118, 0.8 ],
        ...,
        [0.00392216, 0.07843059, 0 ],
        [0.00392216, 0.07843059, 0 ],
        [0.54901961, 0.537254 , 0.56470588],
        [0.54901961, 0.537254 , 0.56470588],
        [0.47058824, 0.44705882, 0.48627451],
        ...,
        [0.06413373, 0.06215686, 0.08823529],
        [0.06667843, 1. , 0.06215686],
        [0.06667843, 1. , 0.06413373]],

       [[0.8 , 0.79215686, 0.81176471],
        [0.72333333, 0.72549002, 0.74509804],
        [0.57647059, 0.59686275, 0.58039216],
        ...,
        [0.6764706 , 1. , 1. ],
        [0.63333333, 0.96862745, 0.94901961],
        [0.99980392, 0.94509804, 0.9254902 ]],

       ...,
       [[0.68235294, 0.71372549, 0.76470588],
        [0.67450588, 0.70588235, 0.75686275],
        [0.69803922, 0.72941176, 0.77254902],
        ...,
        [0.34117647, 0.44705882, 0.38039216],
        [0.29411765, 0.3882529, 0.32549882 ],
        [0.31372549, 0.41176471, 0.3372549 ]],

       [[0.65098039, 0.67843137, 0.74117647],
        [0.64313725, 0.67450588, 0.73549882],
        [0.674509 , 0.70588235, 0.75686275],
        ...,
        [0.36588235, 0.41568627, 0.32941176],
        [0.28235294, 0.36039216, 0.28039922],
        [0.30196078, 0.4 , 0.31764706],
        ...,
        [0.67843137, 0.76862745, 0.76862745],
        [0.674509 , 0.76862745, 0.76470588],
        [0.68235294, 0.77372549, 0.76470588],
        ...,
        [0.24705882, 0.35686275, 0.27568824],
        [0.24313725, 0.34117647, 0.25490196],
        [0.27843137, 0.37647059, 0.29019608]],

       [[0.31372549, 0.30078431, 0.34509804],
        [0.32549882, 0.37647059, 0.34901961],
        [0.29803922, 0.36078431, 0.32156863],
        ...,
        [0.46666667, 0.45882353, 0.45882353],
        [0.45882353, 0.45098039, 0.45098039],
        [0.45098039, 0.44313725, 0.44313725],
        ...,
        [0.25098039, 0.25882353, 0.24705882],
        [0.31372549, 0.33333333, 0.30980392],
        [0.36470588, 0.39215686, 0.36078431],
        ...,
        [0.15686275, 0.18039216, 0.12549882 ],
        [0.15686275, 0.18823529, 0.12941176],
        [0.09490196, 0.09627451, 0.07450980]],

       ...,
       [[0.39215686, 0.47450588, 0.36078431],
        [0.36470588, 0.41568627, 0.32714706],
        [0.35294118, 0.41176471, 0.32714706],
        ...,
        [0.47450588, 0.45490196, 0.50039216],
        [0.45882353, 0.44313725, 0.47450588],
        [0.50980392, 0.49803922, 0.52549882],
        ...,
        [0.52156863, 0.56196078, 0.42745098],
        [0.48666667, 0.44705882, 0.38823529],
        [0.48627451, 0.46666667, 0.38823529]],

       [[0.06666667, 0.06666667, 0.05882353],
        [0.06666667, 0.06666667, 0.05882353],
        [0.06666667, 0.06666667, 0.05882353],
        ...,
        [0.52156863, 0.56196078, 0.42745098],
        [0.48666667, 0.44705882, 0.38823529],
        [0.48627451, 0.46666667, 0.38823529]]],


       [[[0.06588235, 0.08627451, 0.06196078],
        [0.06588235, 0.08627451, 0.06196078],
        [0.06294118, 0.0372549 , 0.03333333],
        ...,
        [0.62941176, 0.69411765, 0.83529412],
        [0.76470588, 0.7254902 , 0.87843137],
        [0.83921569, 0.8 , 0.75294118]],

       [[0.06666667, 0.09333333, 0.04666667],
        [0.06666667, 0.09333333, 0.04666667],
        [0.07254902, 0.04313725, 0.04117647],
        ...,
        [0.61764706, 0.67058824, 0.80784314],
        [0.72156863, 0.6745098 , 0.83686274],
        [0.76078431, 0.71372549, 0.85882353]],

       [[0.07137549, 0.07843137, 0.09019608],
        [0.05098039, 0.02745098, 0.02745098],
        [0.07254902, 0.04705882, 0.04313725],
        ...,
        [0.64509804, 0.69411765, 0.81568627],
        [0.76078431, 0.73725490, 0.86666667],
        [0.82352941, 0.75686275, 0.89411765]],

       ...,
       [[0.03137255, 0.00784314, 0.00784314],
        [0.03921569, 0.02352941, 0.01960784],
        [0.05098039, 0.03529412, 0.02352941],
        ...,
        [0.20392157, 0.30980392, 0.32172549 ],
        [0.27058824, 0.36235294, 0.38980392],
        [0.29803922, 0.37568824, 0.39803922]],

       [[0.10823529, 0.17254902, 0.18862745],
        [0.10215686, 0.16862745, 0.18039216],
        [0.10078431, 0.15215686, 0.16431373],
        ...,
        [0.24411765, 0.30980392, 0.33333333],
        [0.27568824, 0.36235294, 0.38980392],
        [0.29490196, 0.39686667, 0.29411765]],

       [[0.25882353, 0.25490196, 0.24705882],
        [0.27058824, 0.26666667, 0.25882353],
        [0.2745098 , 0.2745098 , 0.26666667],
        ...,
        [0.29019608, 0.35196078, 0.32941176],
        [0.26666667, 0.27843137, 0.30588235],
        [0.24313725, 0.25490196, 0.25235294]],


       [[0.23921569, 0.23921569, 0.24705882],
        [0.29411765, 0.30196078, 0.13960392],
        [0.09803922, 0.09803922, 0.12941176],
        ...,
        [0.18431373, 0.2 , 0.20392157],
        [0.15294118, 0.15215686, 0.16862745],
        [0.17647059, 0.15215686, 0.19607843],
        ...,
        [0.04313725, 0.04705882, 0.05490196],
        [0.0627451 , 0.07843137, 0.08235294],
        [0.07058824, 0.08235294, 0.10980392],
        ...,
        [0.2588235, 0.2745098 , 0.27843137],
        [0.23921569, 0.24490196, 0.26666667],
        [0.23137255, 0.24705882, 0.25882353],
        ...,
        [0.01176471, 0.04705882, 0.03529412],
        [0.09803922, 0.10196078, 0.09019608],
        [0.13333333, 0.16470588, 0.17647059],
        ...,
        [0.12941176, 0.34901961, 0.30470423],
        [0.29019608, 0.38039392, 0.3254902 ],
        [0.2745098 , 0.29411765, 0.30588235]],

       ...,
       [[0.57647059, 0.56862745, 0.52156863],
        [0.53333333, 0.52156863, 0.50823529],
        [0.51764706, 0.50588235, 0.57254902],
        ...,
        [0.40627451, 0.43137255, 0.41960784],
        [0.45098039, 0.39607843, 0.38431373],
        [0.50980392, 0.45490196, 0.44313725]],

       [[0.54509804, 0.53333333, 0.6 ],
        [0.50588235, 0.49411765, 0.56470431],
        [0.50196078, 0.49019608, 0.56470588],
        ...,
        [0.45882353, 0.40392157, 0.4 ],
        [0.38039216, 0.31372549, 0.30980392],
        [0.43529412, 0.39215686, 0.38431373]],

       [[0.52941176, 0.51764706, 0.58431373],
        [0.47843137, 0.46666667, 0.53333333],
        [0.50470431, 0.54901961, 0.51372549],
        ...,
        [0.58431373, 0.52941176, 0.52941176],
        [0.48235294, 0.42352942, 0.42352942],
        [0.52941176, 0.46235294, 0.46235294]]],


       [[[0.25882353, 0.12156863, 0.04705882],
        [0.40274431, 0.24705882, 0.16470588],
        [0.74470588, 0.58470588, 0.27843137],
        ...,
        [0.3372549 , 0.17647059, 0.09803922],
        [0.32156863, 0.16862745, 0.09803922],
        [0.33333333, 0.18431373, 0.10196078]],

       [[0.21960627, 0.11372549, 0.04705882],
        [0.34509804, 0.19686275, 0.07843137],
        [0.40000392, 0.31372549, 0.21960627],
        ...,
        [0.35686275, 0.19215686, 0.09803922],
        [0.34117647, 0.18431373, 0.08627451],
        [0.33333333, 0.17647059, 0.07843137]],

       [[0.18627451, 0.12156863, 0.04705882],
        [0.40274431, 0.24705882, 0.16470588],
        [0.74470588, 0.58470588, 0.27843137],
        ...,
        [0.3372549 , 0.17647059, 0.09803922],
        [0.32156863, 0.16862745, 0.09803922],
        [0.33333333, 0.18431373, 0.10196078]],

       [[0.2588235, 0.1529412, 0.0745098],
        [0.40274431, 0.24705882, 0.16470588],
        [0.74470588, 0.58470588, 0.27843137],
        ...,
        [0.3372549 , 0.17647059, 0.09803922],
        [0.32156863, 0.16862745, 0.09803922],
        [0.33333333, 0.18431373, 0.10196078]],

       ...,
       [[0.45098039, 0.47450588, 0.29411765],
        [0.45882353, 0.48235294, 0.29411765],
        [0.47843137, 0.49803137, 0.29627451],
        ...,
        [0.43529412, 0.36431373, 0.31764706],
        [0.2745098 , 0.22745098, 0.17254902],
        [0.23921569, 0.19215686, 0.12725490 ]]]])
```

In [13]:
```
print("Shape of X_train: ", X_train.shape)
print("Shape of Y_train: ", Y_train.shape)
print("Shape of X_test: ", X_test.shape)
print("Shape of Y_test: ", Y_test.shape)
```

```
Shape of X_train:  (2000, 100, 100, 3)
Shape of Y_train:  (2000, 1)
Shape of X_test:  (400, 100, 100, 3)
Shape of Y_test:  (400, 1)
```

In [14]:
```
idx=random.randint(0,len(X_train))
plt.imshow(X_train[idx, :])
plt.show()
```

Model

In [17]:
```
model=Sequential()
    Conv2D(32,(3,3),activation='relu',input_shape=(100,100,3),
    MaxPooling2D(2,2),

    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dense(64,activation='relu'),
    Dense(1,activation='sigmoid')
)
```

In [18]:
```
## another method

model=Sequential()

model.add(Conv2D(32,(3,3),activation='relu',input_shape=(100,100,3)))
model.add(MaxPooling2D(2,2))

model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))

model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
```

In [19]:
```
## compiling model
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

In [22]:
```
## now fitting our model

model.fit(X_train,Y_train,epochs=5,batch_size=64)
```

```
Epoch 1/5
32/32 [==============================] - 9s 273ms/step - loss: 0.4956 - accuracy: 0.7605
Epoch 2/5
32/32 [==============================] - 9s 272ms/step - loss: 0.4579 - accuracy: 0.7840
Epoch 3/5
32/32 [==============================] - 9s 270ms/step - loss: 0.3742 - accuracy: 0.8260
Epoch 4/5
32/32 [==============================] - 9s 267ms/step - loss: 0.3071 - accuracy: 0.8715
Epoch 5/5
32/32 [==============================] - 9s 267ms/step - loss: 0.2566 - accuracy: 0.8915
```

Out[22]:
```
<keras.callbacks.History at 0x20640af5040>
```

In [23]:
```
model.evaluate(X_test,Y_test)
```

```
13/13 [==============================] - 8s 28ms/step - loss: 0.7668 - accuracy: 0.6725
```

Out[23]:
```
[0.7668189918255664, 0.6725000143051147]
```

Making Prediction

In [24]:
```
idx2=random.randint(0,len(Y_test))
plt.imshow(X_test[idx2, :])
plt.show()

y_pred=model.predict(X_test[idx2, ].reshape(1,100,100,3))
y_pred=y_pred >0.5

if(y_pred==0):
    pred='dog'
else:
    pred='cat'

print("this is: ",pred)
```


```
1/1 [==============================] - 0s 18ms/step
this is:  dog
```

In [ ]:

In [ ]: