In [ ]:

In [ ]:

In [ ]:

In [12]:

```
from keras.models import load_model
from keras.preprocessing import image
import numpy as np
import cv2
```

```
Using TensorFlow backend.
```

In [13]:

```
model = load_model("animal.h5")
```

```
WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\keras\backen
d\tensorflow_backend.py:517: The name tf.placeholder is deprecated. Please use
tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\keras\backen
d\tensorflow_backend.py:4138: The name tf.random_uniform is deprecated. Please
use tf.random.uniform instead.

WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\keras\backen
d\tensorflow_backend.py:3976: The name tf.nn.max_pool is deprecated. Please use
tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\keras\backen
d\tensorflow_backend.py:174: The name tf.get_default_session is deprecated. Ple
ase use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\keras\backen
d\tensorflow_backend.py:181: The name tf.ConfigProto is deprecated. Please use
tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\keras\backen
d\tensorflow_backend.py:186: The name tf.Session is deprecated. Please use tf.c
ompat.v1.Session instead.

WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\keras\optimi
zers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v
1.train.Optimizer instead.

WARNING:tensorflow:From C:\Users\Admin\Anaconda3\lib\site-packages\tensorflow\p
ython\ops\math_grad.py:1250: add_dispatch_support.<locals>.wrapper (from tensor
flow.python.ops.array_ops) is deprecated and will be removed in a future versio
n.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

In [16]:

```
img = image.load_img(r'x_testing\bears\b100.jpg',target_size = (64,64))
```

In [17]:

```
x = image.img_to_array(img)#it has 3 dimenstion but 2D convolution will 2 diment
x= np.expand_dims(x,axis = 0)
```

In [18]:

```
x.shape
```

Out[18]:

```
(1, 64, 64, 3)
```

In [19]:

```
pred = model.predict_classes(x)
```

In [20]:

```
pred
```

Out[20]:

```
array([2], dtype=int64)
```

In [23]:

```
#in the above the answer is two so the bear it correct prediction.
```

In [ ]:

```

```