

```
In [1]: import os
import PIL
from PIL import Image
from PIL.ImageDraw import Draw
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from keras.models import load_model
from keras.preprocessing import image
```

Using TensorFlow backend.

```
In [3]: ls ../

anaconda3/      env/            mpich-3.2/      'Ubuntu_vm'
(1) '/'         anneal_data.csv fig_code/        Music/          'Ubuntu_vm'
(1).zip'        Desktop/        foo/            Pictures/        Untitled1.ip
ynb              'DG foundation'/ kp.jpg          prati.py         Untitled2.ip
ynb              djangodev/      kpk.jpg         Public/          Untitled3.ip
ynb              Documents/      'ML LAB 18MCB1005'/ Templates/       Untitled4.ip
ynb              Downloads/      mozilla.pdf     ticket.pdf.odt  Untitled.ipynb
```

```
In [4]: MODEL_BASE = '/home/lab2/DG foundation/Whale/models_csv'
DATA = '/home/lab2/DG foundation'
TRAIN_DATA = os.path.join(DATA, 'train')
TEST_DATA = os.path.join(DATA, 'test')
```

```
In [5]: model = load_model(os.path.join(MODEL_BASE, 'cropping.model'))  
        model.summary()
```

Layer (type) ected to	Output Shape	Param #	Conn
=====	=====	=====	=====
input_2 (InputLayer)	(None, 128, 128, 1)	0	
conv2d_18 (Conv2D) t_2[0][0]	(None, 128, 128, 64)	5248	input_2[0][0]
conv2d_19 (Conv2D) 2d_18[0][0]	(None, 128, 128, 64)	36928	conv2d_18[0][0]
batch_normalization_7 (BatchNor 2d_19[0][0]	(None, 128, 128, 64)	256	conv2d_19[0][0]
conv2d_20 (Conv2D) h_normalization_7[0][0]	(None, 64, 64, 64)	16448	batch_normalization_7[0][0]
conv2d_21 (Conv2D) 2d_20[0][0]	(None, 64, 64, 64)	36928	conv2d_20[0][0]
conv2d_22 (Conv2D) 2d_21[0][0]	(None, 64, 64, 64)	36928	conv2d_21[0][0]
batch_normalization_8 (BatchNor 2d_22[0][0]	(None, 64, 64, 64)	256	conv2d_22[0][0]
conv2d_23 (Conv2D) h_normalization_8[0][0]	(None, 32, 32, 64)	16448	batch_normalization_8[0][0]
conv2d_24 (Conv2D) 2d_23[0][0]	(None, 32, 32, 64)	36928	conv2d_23[0][0]
conv2d_25 (Conv2D) 2d_24[0][0]	(None, 32, 32, 64)	36928	conv2d_24[0][0]
batch_normalization_9 (BatchNor 2d_25[0][0]	(None, 32, 32, 64)	256	conv2d_25[0][0]
conv2d_26 (Conv2D) h_normalization_9[0][0]	(None, 16, 16, 64)	16448	batch_normalization_9[0][0]

conv2d_27 (Conv2D) 2d_26[0][0]	(None, 16, 16, 64)	36928	conv
conv2d_28 (Conv2D) 2d_27[0][0]	(None, 16, 16, 64)	36928	conv
batch_normalization_10 (Batch Normalization) 2d_28[0][0]	(None, 16, 16, 64)	256	conv
conv2d_29 (Conv2D) h_normalization_10[0][0]	(None, 8, 8, 64)	16448	batch_normalization_10
conv2d_30 (Conv2D) 2d_29[0][0]	(None, 8, 8, 64)	36928	conv
conv2d_31 (Conv2D) 2d_30[0][0]	(None, 8, 8, 64)	36928	conv
batch_normalization_11 (Batch Normalization) 2d_31[0][0]	(None, 8, 8, 64)	256	conv
conv2d_32 (Conv2D) h_normalization_11[0][0]	(None, 4, 4, 64)	16448	batch_normalization_11
conv2d_33 (Conv2D) 2d_32[0][0]	(None, 4, 4, 64)	36928	conv
conv2d_34 (Conv2D) 2d_33[0][0]	(None, 4, 4, 64)	36928	conv
batch_normalization_12 (Batch Normalization) 2d_34[0][0]	(None, 4, 4, 64)	256	conv
max_pooling2d_3 (MaxPooling2D) h_normalization_12[0][0]	(None, 4, 1, 64)	0	batch_normalization_12
max_pooling2d_4 (MaxPooling2D) h_normalization_12[0][0]	(None, 1, 4, 64)	0	batch_normalization_12
flatten_3 (Flatten) pooling2d_3[0][0]	(None, 256)	0	max_pooling2d_3

flatten_4 (Flatten) pooling2d_4[0][0]	(None, 256)	0	max_
dense_4 (Dense) ten_3[0][0]	(None, 16)	4112	flat
dense_5 (Dense) ten_4[0][0]	(None, 16)	4112	flat
concatenate_2 (Concatenate) e_4[0][0] e_5[0][0]	(None, 32)	0	dens dens
dense_6 (Dense) atenate_2[0][0]	(None, 4)	132	conc
=====			
Total params: 503,588			
Trainable params: 502,820			
Non-trainable params: 768			

```
In [6]: train_paths = [os.path.join(TRAIN_DATA, img) for img in os.listdir(TRAIN_DATA)]
test_paths = [os.path.join(TEST_DATA, img) for img in os.listdir(TEST_DATA)]
```

```
In [7]: train_paths[1]
```

```
Out[7]: '/home/lab2/DG foundation/train/c1b007419.jpg'
```

```
In [31]: img = image.load_img(train_paths[5])
```

In [32]: `img`

Out[32]:

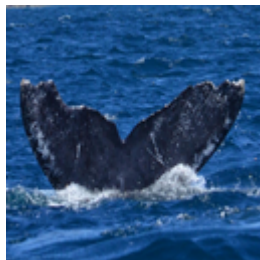


In [33]: `img_arr = image.img_to_array(img)`

In [35]: `rimg = img.resize((128, 128), PIL.Image.ANTIALIAS)`

In [36]: `rimg`

Out[36]:



In [37]: `rimg_arr1 = image.img_to_array(rimg)`

In [39]: `rimg_ = rimg.convert('L')`

In [40]: `rimg_`
`#rimg_arr_ = image.img_to_array(rimg_)`

Out[40]:



```
In [41]: rimg_arr_ = image.img_to_array(rimg_)
```

```
In [42]: bbox = model.predict(np.expand_dims(rimg_arr_, axis=0))
```

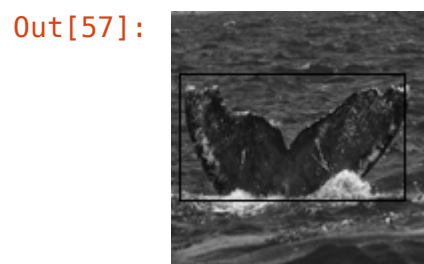
```
In [43]: bbox
```

```
Out[43]: array([[ 4.295079,  31.994919, 116.87786 ,  94.61023 ]], dtype=float32)
```

```
In [53]: draw = Draw(rimg_)
```

```
In [56]: draw.rectangle(bbox, outline="Black")
```

```
In [57]: rimg_
```



```
In [58]: rimg
```



```
In [59]: img_crop = rimg_.crop(tuple(bbox[0]))
```

```
In [60]: img_crop
```



```
In [61]: def make_bbox_image(img_path):  
    main_img = image.load_img(img_path)  
    r_img = main_img.resize((128, 128), PIL.Image.ANTIALIAS)  
    rb_img = r_img.convert('L')  
    rb_img_arr = image.img_to_array(rb_img)  
    bbox = model.predict(np.expand_dims(rb_img_arr, axis=0))  
    img_crop = r_img.crop(tuple(bbox[0]))  
    img_arr = image.img_to_array(img_crop)  
    return img_crop, main_img
```

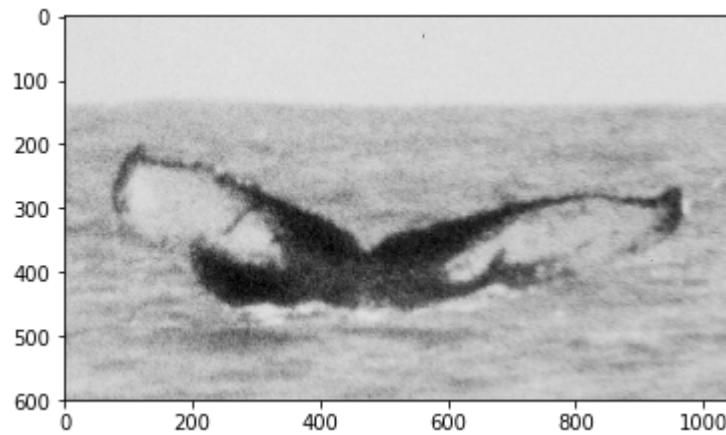
```
In [62]: train_paths[13]
```

```
Out[62]: '/home/lab2/DG foundation/train/5db98037f.jpg'
```

```
In [63]: img_cr, img_or = make_bbox_image(train_paths[13])
```

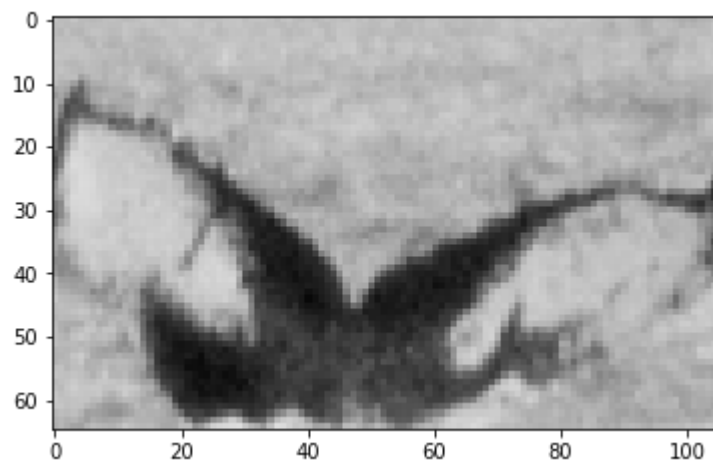
```
In [64]: plt.imshow(img_or)
```

```
Out[64]: <matplotlib.image.AxesImage at 0x7f6a4d4e29b0>
```



```
In [65]: plt.imshow(img_cr)
```

```
Out[65]: <matplotlib.image.AxesImage at 0x7f6a4d086c18>
```



```
In [ ]:
```