In [1]:

```
# This Python 3 environment comes with many helpful analytics
 libraries installed
# It is defined by the kaggle/python docker image: https://git
hub.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.r
ead_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift
+Enter) will list the files in the input directory

import os
print(os.listdir("../input"))
import keras
print(keras.__version__)
# Any results you write to the current directory are saved as
 output.
```

```
['humpback-whale-identification', 'whales-cropped']


Using TensorFlow backend.


2.2.4
```

In [2]:

```
HW = 'humpback-whale-identification'
# TRAIN = '../input/humpback-whale-identification/train/'
TRAIN_CROPPED = "whales-cropped/cropped_train/cropped_train/"
TRAIN_CROPPED_IN = '../input/' + TRAIN_CROPPED

# TEST = '../input/humpback-whale-identification/test/'
TEST_CROPPED = "whales-cropped/cropped_test/cropped_test/"
TEST_CROPPED_IN = '../input/' + TEST_CROPPED

LABELS = '../input/humpback-whale-identification/train.csv'
SAMPLE_SUB = '../input/humpback-whale-identification/sample_s
ubmission.csv'

train = pd.read_csv(LABELS)
print("With new_whale:")
train.head()
```

```
With new_whale:
```

Out[2]:

	Image	Id
0	0000e88ab.jpg	w_f48451c
1	0001f9222.jpg	w_c3d896a
2	00029d126.jpg	w_20df2c5
3	00050a15a.jpg	new_whale
4	0005c1ef8.jpg	new_whale

In [3]:

```
MODEL_F = 'Model_ResNet50.h5'
WEIGHTS_F = 'Weights_ResNet50.h5'
```

```
MODEL = '../input/ResNet50/' + MODEL_F
WEIGHTS = '../input/ResNet50/'+ WEIGHTS_F
```

In [4]:

```
train.describe()
```

Out[4]:

	Image	Id
count	25361	25361
unique	25361	5005
top	ca07480bd.jpg	new_whale
freq	1	9664

In [5]:

```
import random
from IPython.display import Image
print("Example whale image")

#show sample image
name = random.choice(train['Image'])
print(name)
Image(filename = TRAIN_CROPPED_IN + name)
```

```
Example whale image
804331825.jpg
```

Out[5]:

In [6]:

```
criteria = train['Id'] != 'new_whale'
whales_train = train[criteria]

print("Without new_whale:")
whales_train.head()
```

```
Without new_whale:
```

Out[6]:

	Image	Id
0	0000e88ab.jpg	w_f48451c
1	0001f9222.jpg	w_c3d896a
2	00029d126.jpg	w_20df2c5
6	000a6daec.jpg	w_dd88965
8	0016b897a.jpg	w_64404ac

In [7]:

```
unique_labels = np.unique(whales_train.Id.values)
labels_list = unique_labels
```

In [8]:

```
whales_train.describe()
```

Out[8]:

	Image	Id
count	15697	15697
unique	15697	5004
top	6a28f4d47.jpg	w_23a388d
freq	1	73

In [9]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mplimg
from matplotlib.pyplot import imshow

from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder

from keras import layers
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator

# from keras.applications.imagenet_utils import preprocess_inp
ut
from keras.applications.resnet50 import ResNet50, preprocess_
input
# from keras.applications.xception import Xception, preprocess
_input
# from keras.applications.inception_resnet_v2 import Inception
ResNetV2, preprocess_input

from keras.losses import binary_crossentropy
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Input, Dense, Activation, BatchNorma
lization, Flatten, Conv2D, GlobalAveragePooling2D
from keras.layers import AveragePooling2D, MaxPooling2D, Drop
out
from keras.models import Model
from keras.metrics import top_k_categorical_accuracy

import keras.backend as K
from keras.models import Sequential
from PIL import Image
import gc
import warnings
warnings.simplefilter("ignore", category=DeprecationWarning)

%matplotlib inline
```

In [10]:

```
IMAGE_HEIGHT = 128
IMAGE_WIDTH = 128
IMAGE_SHAPE = (IMAGE_HEIGHT, IMAGE_WIDTH, 3)
```

In [11]:

```
CLASSES = 5004
EPOCHS = 5
BATCH_SIZE = 32

def top_5_accuracy(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=5)

# setup model
base_model = ResNet50(weights='imagenet', include_top=False,
```

```
input_shape = IMAGE_SHAPE)

x = base_model.output
x = GlobalAveragePooling2D(name='avg_pool')(x)
x = Dropout(0.4)(x)
predictions = Dense(CLASSES, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)

# transfer learning
for layer in base_model.layers:
  layer.trainable = True

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy', 'mae', top_5_accuracy])

model.summary()
```

```
/opt/conda/lib/python3.6/site-packages/keras_applications/res
net50.py:265: UserWarning: The output shape of `ResNet50(incl
ude_top=False)` has been changed since Keras 2.2.0.
  warnings.warn('The output shape of `ResNet50(include_top=Fa
lse)` '


Downloading data from https://github.com/fchollet/deep-learni
ng-models/releases/download/v0.2/resnet50_weights_tf_dim_orde
ring_tf_kernels_notop.h5
94658560/94653016 [==============================] - 7s 0us/s
tep
__________________________________________________________
_____________________________________
Layer (type)                 Output Shape         Param #
    Connected to
==========================================================
=====================================
input_1 (InputLayer)         (None, 128, 128, 3)  0


__________________________________________________________
_____________________________________
conv1_pad (ZeroPadding2D)    (None, 134, 134, 3)  0
    input_1[0][0]
__________________________________________________________
_____________________________________
conv1 (Conv2D)               (None, 64, 64, 64)   9472
    conv1_pad[0][0]
__________________________________________________________
_____________________________________
bn_conv1 (BatchNormalization)   (None, 64, 64, 64)   256
    conv1[0][0]
__________________________________________________________
_____________________________________
activation_1 (Activation)    (None, 64, 64, 64)   0
    bn_conv1[0][0]
__________________________________________________________
_____________________________________
pool1_pad (ZeroPadding2D)    (None, 66, 66, 64)   0
    activation_1[0][0]
__________________________________________________________
_____________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 32, 32, 64)   0
    pool1_pad[0][0]
__________________________________________________________
_____________________________________
res2a_branch2a (Conv2D)      (None, 32, 32, 64)   4160
    max_pooling2d_1[0][0]
```

```
____________________________________________________________________
____________________________________
bn2a_branch2a (BatchNormalizati (None, 32, 32, 64)   256
    res2a_branch2a[0][0]
____________________________________________________________________
____________________________________
activation_2 (Activation)       (None, 32, 32, 64)   0
    bn2a_branch2a[0][0]
____________________________________________________________________
____________________________________
res2a_branch2b (Conv2D)         (None, 32, 32, 64)   36928
    activation_2[0][0]
____________________________________________________________________
____________________________________
bn2a_branch2b (BatchNormalizati (None, 32, 32, 64)   256
    res2a_branch2b[0][0]
____________________________________________________________________
____________________________________
activation_3 (Activation)       (None, 32, 32, 64)   0
    bn2a_branch2b[0][0]
____________________________________________________________________
____________________________________
res2a_branch2c (Conv2D)         (None, 32, 32, 256)  16640
    activation_3[0][0]
____________________________________________________________________
____________________________________
res2a_branch1 (Conv2D)          (None, 32, 32, 256)  16640
    max_pooling2d_1[0][0]
____________________________________________________________________
____________________________________
bn2a_branch2c (BatchNormalizati (None, 32, 32, 256)  1024
    res2a_branch2c[0][0]
____________________________________________________________________
____________________________________
bn2a_branch1 (BatchNormalizatio (None, 32, 32, 256)  1024
    res2a_branch1[0][0]
____________________________________________________________________
____________________________________
add_1 (Add)                     (None, 32, 32, 256)  0
    bn2a_branch2c[0][0]

    bn2a_branch1[0][0]
____________________________________________________________________
____________________________________
activation_4 (Activation)       (None, 32, 32, 256)  0
    add_1[0][0]
____________________________________________________________________
____________________________________
res2b_branch2a (Conv2D)         (None, 32, 32, 64)   16448
    activation_4[0][0]
____________________________________________________________________
____________________________________
bn2b_branch2a (BatchNormalizati (None, 32, 32, 64)   256
    res2b_branch2a[0][0]
____________________________________________________________________
____________________________________
activation_5 (Activation)       (None, 32, 32, 64)   0
    bn2b_branch2a[0][0]
____________________________________________________________________
____________________________________
res2b_branch2b (Conv2D)         (None, 32, 32, 64)   36928
    activation_5[0][0]
____________________________________________________________________
____________________________________
bn2b_branch2b (BatchNormalizati (None, 32, 32, 64)   256
    res2b_branch2b[0][0]
```

```
activation_6 (Activation)       (None, 32, 32, 64)   0
    bn2b_branch2b[0][0]
__________________________________________________________________
__________________________________
res2b_branch2c (Conv2D)         (None, 32, 32, 256)  16640
    activation_6[0][0]
__________________________________________________________________
__________________________________
bn2b_branch2c (BatchNormalizati (None, 32, 32, 256)  1024
    res2b_branch2c[0][0]
__________________________________________________________________
__________________________________
add_2 (Add)                     (None, 32, 32, 256)  0
    bn2b_branch2c[0][0]

    activation_4[0][0]
__________________________________________________________________
__________________________________
activation_7 (Activation)       (None, 32, 32, 256)  0
    add_2[0][0]
__________________________________________________________________
__________________________________
res2c_branch2a (Conv2D)         (None, 32, 32, 64)   16448
    activation_7[0][0]
__________________________________________________________________
__________________________________
bn2c_branch2a (BatchNormalizati (None, 32, 32, 64)   256
    res2c_branch2a[0][0]
__________________________________________________________________
__________________________________
activation_8 (Activation)       (None, 32, 32, 64)   0
    bn2c_branch2a[0][0]
__________________________________________________________________
__________________________________
res2c_branch2b (Conv2D)         (None, 32, 32, 64)   36928
    activation_8[0][0]
__________________________________________________________________
__________________________________
bn2c_branch2b (BatchNormalizati (None, 32, 32, 64)   256
    res2c_branch2b[0][0]
__________________________________________________________________
__________________________________
activation_9 (Activation)       (None, 32, 32, 64)   0
    bn2c_branch2b[0][0]
__________________________________________________________________
__________________________________
res2c_branch2c (Conv2D)         (None, 32, 32, 256)  16640
    activation_9[0][0]
__________________________________________________________________
__________________________________
bn2c_branch2c (BatchNormalizati (None, 32, 32, 256)  1024
    res2c_branch2c[0][0]
__________________________________________________________________
__________________________________
add_3 (Add)                     (None, 32, 32, 256)  0
    bn2c_branch2c[0][0]

    activation_7[0][0]
__________________________________________________________________
__________________________________
activation_10 (Activation)      (None, 32, 32, 256)  0
    add_3[0][0]
__________________________________________________________________
__________________________________
res3a_branch2a (Conv2D)         (None, 16, 16, 128)  32896
```

```
    activation_10[0][0]
==================================================================
==================================
bn3a_branch2a (BatchNormalizati (None, 16, 16, 128)  512
    res3a_branch2a[0][0]
------------------------------------------------------------------
----------------------------------
activation_11 (Activation)      (None, 16, 16, 128)  0
    bn3a_branch2a[0][0]
------------------------------------------------------------------
----------------------------------
res3a_branch2b (Conv2D)         (None, 16, 16, 128)  147584
    activation_11[0][0]
------------------------------------------------------------------
----------------------------------
bn3a_branch2b (BatchNormalizati (None, 16, 16, 128)  512
    res3a_branch2b[0][0]
------------------------------------------------------------------
----------------------------------
activation_12 (Activation)      (None, 16, 16, 128)  0
    bn3a_branch2b[0][0]
------------------------------------------------------------------
----------------------------------
res3a_branch2c (Conv2D)         (None, 16, 16, 512)  66048
    activation_12[0][0]
------------------------------------------------------------------
----------------------------------
res3a_branch1 (Conv2D)          (None, 16, 16, 512)  131584
    activation_10[0][0]
------------------------------------------------------------------
----------------------------------
bn3a_branch2c (BatchNormalizati (None, 16, 16, 512)  2048
    res3a_branch2c[0][0]
------------------------------------------------------------------
----------------------------------
bn3a_branch1 (BatchNormalizatio (None, 16, 16, 512)  2048
    res3a_branch1[0][0]
------------------------------------------------------------------
----------------------------------
add_4 (Add)                     (None, 16, 16, 512)  0
    bn3a_branch2c[0][0]

    bn3a_branch1[0][0]
------------------------------------------------------------------
----------------------------------
activation_13 (Activation)      (None, 16, 16, 512)  0
    add_4[0][0]
------------------------------------------------------------------
----------------------------------
res3b_branch2a (Conv2D)         (None, 16, 16, 128)  65664
    activation_13[0][0]
------------------------------------------------------------------
----------------------------------
bn3b_branch2a (BatchNormalizati (None, 16, 16, 128)  512
    res3b_branch2a[0][0]
------------------------------------------------------------------
----------------------------------
activation_14 (Activation)      (None, 16, 16, 128)  0
    bn3b_branch2a[0][0]
------------------------------------------------------------------
----------------------------------
res3b_branch2b (Conv2D)         (None, 16, 16, 128)  147584
    activation_14[0][0]
------------------------------------------------------------------
----------------------------------
bn3b_branch2b (BatchNormalizati (None, 16, 16, 128)  512
```

```
    res3b_branch2b[0][0]
==================================================================================================
activation_15 (Activation)      (None, 16, 16, 128)  0
    bn3b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res3b_branch2c (Conv2D)         (None, 16, 16, 512)  66048
    activation_15[0][0]
--------------------------------------------------------------------------------------------------
bn3b_branch2c (BatchNormalizati (None, 16, 16, 512)  2048
    res3b_branch2c[0][0]
--------------------------------------------------------------------------------------------------
add_5 (Add)                     (None, 16, 16, 512)  0
    bn3b_branch2c[0][0]

    activation_13[0][0]
--------------------------------------------------------------------------------------------------
activation_16 (Activation)      (None, 16, 16, 512)  0
    add_5[0][0]
--------------------------------------------------------------------------------------------------
res3c_branch2a (Conv2D)         (None, 16, 16, 128)  65664
    activation_16[0][0]
--------------------------------------------------------------------------------------------------
bn3c_branch2a (BatchNormalizati (None, 16, 16, 128)  512
    res3c_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_17 (Activation)      (None, 16, 16, 128)  0
    bn3c_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res3c_branch2b (Conv2D)         (None, 16, 16, 128)  147584
    activation_17[0][0]
--------------------------------------------------------------------------------------------------
bn3c_branch2b (BatchNormalizati (None, 16, 16, 128)  512
    res3c_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_18 (Activation)      (None, 16, 16, 128)  0
    bn3c_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res3c_branch2c (Conv2D)         (None, 16, 16, 512)  66048
    activation_18[0][0]
--------------------------------------------------------------------------------------------------
bn3c_branch2c (BatchNormalizati (None, 16, 16, 512)  2048
    res3c_branch2c[0][0]
--------------------------------------------------------------------------------------------------
add_6 (Add)                     (None, 16, 16, 512)  0
    bn3c_branch2c[0][0]

    activation_16[0][0]
--------------------------------------------------------------------------------------------------
activation_19 (Activation)      (None, 16, 16, 512)  0
    add_6[0][0]
--------------------------------------------------------------------------------------------------
```

```
res3d_branch2a (Conv2D)         (None, 16, 16, 128)  65664
    activation_19[0][0]
____________________________________________________________
____________________________________
bn3d_branch2a (BatchNormalizati (None, 16, 16, 128)  512
    res3d_branch2a[0][0]
____________________________________________________________
____________________________________
activation_20 (Activation)      (None, 16, 16, 128)  0
    bn3d_branch2a[0][0]
____________________________________________________________
____________________________________
res3d_branch2b (Conv2D)         (None, 16, 16, 128)  147584
    activation_20[0][0]
____________________________________________________________
____________________________________
bn3d_branch2b (BatchNormalizati (None, 16, 16, 128)  512
    res3d_branch2b[0][0]
____________________________________________________________
____________________________________
activation_21 (Activation)      (None, 16, 16, 128)  0
    bn3d_branch2b[0][0]
____________________________________________________________
____________________________________
res3d_branch2c (Conv2D)         (None, 16, 16, 512)  66048
    activation_21[0][0]
____________________________________________________________
____________________________________
bn3d_branch2c (BatchNormalizati (None, 16, 16, 512)  2048
    res3d_branch2c[0][0]
____________________________________________________________
____________________________________
add_7 (Add)                     (None, 16, 16, 512)  0
    bn3d_branch2c[0][0]

    activation_19[0][0]
____________________________________________________________
____________________________________
activation_22 (Activation)      (None, 16, 16, 512)  0
    add_7[0][0]
____________________________________________________________
____________________________________
res4a_branch2a (Conv2D)         (None, 8, 8, 256)    131328
    activation_22[0][0]
____________________________________________________________
____________________________________
bn4a_branch2a (BatchNormalizati (None, 8, 8, 256)    1024
    res4a_branch2a[0][0]
____________________________________________________________
____________________________________
activation_23 (Activation)      (None, 8, 8, 256)    0
    bn4a_branch2a[0][0]
____________________________________________________________
____________________________________
res4a_branch2b (Conv2D)         (None, 8, 8, 256)    590080
    activation_23[0][0]
____________________________________________________________
____________________________________
bn4a_branch2b (BatchNormalizati (None, 8, 8, 256)    1024
    res4a_branch2b[0][0]
____________________________________________________________
____________________________________
activation_24 (Activation)      (None, 8, 8, 256)    0
    bn4a_branch2b[0][0]
____________________________________________________________
____________________________________
```

```
res4a_branch2c (Conv2D)         (None, 8, 8, 1024)   263168
    activation_24[0][0]
__________________________________________________________________
_____________________________________
res4a_branch1 (Conv2D)          (None, 8, 8, 1024)   525312
    activation_22[0][0]
__________________________________________________________________
_____________________________________
bn4a_branch2c (BatchNormalizati (None, 8, 8, 1024)   4096
    res4a_branch2c[0][0]
__________________________________________________________________
_____________________________________
bn4a_branch1 (BatchNormalizatio (None, 8, 8, 1024)   4096
    res4a_branch1[0][0]
__________________________________________________________________
_____________________________________
add_8 (Add)                     (None, 8, 8, 1024)   0
    bn4a_branch2c[0][0]

    bn4a_branch1[0][0]
__________________________________________________________________
_____________________________________
activation_25 (Activation)      (None, 8, 8, 1024)   0
    add_8[0][0]
__________________________________________________________________
_____________________________________
res4b_branch2a (Conv2D)         (None, 8, 8, 256)    262400
    activation_25[0][0]
__________________________________________________________________
_____________________________________
bn4b_branch2a (BatchNormalizati (None, 8, 8, 256)    1024
    res4b_branch2a[0][0]
__________________________________________________________________
_____________________________________
activation_26 (Activation)      (None, 8, 8, 256)    0
    bn4b_branch2a[0][0]
__________________________________________________________________
_____________________________________
res4b_branch2b (Conv2D)         (None, 8, 8, 256)    590080
    activation_26[0][0]
__________________________________________________________________
_____________________________________
bn4b_branch2b (BatchNormalizati (None, 8, 8, 256)    1024
    res4b_branch2b[0][0]
__________________________________________________________________
_____________________________________
activation_27 (Activation)      (None, 8, 8, 256)    0
    bn4b_branch2b[0][0]
__________________________________________________________________
_____________________________________
res4b_branch2c (Conv2D)         (None, 8, 8, 1024)   263168
    activation_27[0][0]
__________________________________________________________________
_____________________________________
bn4b_branch2c (BatchNormalizati (None, 8, 8, 1024)   4096
    res4b_branch2c[0][0]
__________________________________________________________________
_____________________________________
add_9 (Add)                     (None, 8, 8, 1024)   0
    bn4b_branch2c[0][0]

    activation_25[0][0]
__________________________________________________________________
_____________________________________
activation_28 (Activation)      (None, 8, 8, 1024)   0
    add_9[0][0]
```

```
____________________________________________________________________
____________________________________
res4c_branch2a (Conv2D)         (None, 8, 8, 256)    262400
    activation_28[0][0]
____________________________________________________________________
____________________________________
bn4c_branch2a (BatchNormalizati (None, 8, 8, 256)    1024
    res4c_branch2a[0][0]
____________________________________________________________________
____________________________________
activation_29 (Activation)      (None, 8, 8, 256)    0
    bn4c_branch2a[0][0]
____________________________________________________________________
____________________________________
res4c_branch2b (Conv2D)         (None, 8, 8, 256)    590080
    activation_29[0][0]
____________________________________________________________________
____________________________________
bn4c_branch2b (BatchNormalizati (None, 8, 8, 256)    1024
    res4c_branch2b[0][0]
____________________________________________________________________
____________________________________
activation_30 (Activation)      (None, 8, 8, 256)    0
    bn4c_branch2b[0][0]
____________________________________________________________________
____________________________________
res4c_branch2c (Conv2D)         (None, 8, 8, 1024)   263168
    activation_30[0][0]
____________________________________________________________________
____________________________________
bn4c_branch2c (BatchNormalizati (None, 8, 8, 1024)   4096
    res4c_branch2c[0][0]
____________________________________________________________________
____________________________________
add_10 (Add)                    (None, 8, 8, 1024)   0
    bn4c_branch2c[0][0]

    activation_28[0][0]
____________________________________________________________________
____________________________________
activation_31 (Activation)      (None, 8, 8, 1024)   0
    add_10[0][0]
____________________________________________________________________
____________________________________
res4d_branch2a (Conv2D)         (None, 8, 8, 256)    262400
    activation_31[0][0]
____________________________________________________________________
____________________________________
bn4d_branch2a (BatchNormalizati (None, 8, 8, 256)    1024
    res4d_branch2a[0][0]
____________________________________________________________________
____________________________________
activation_32 (Activation)      (None, 8, 8, 256)    0
    bn4d_branch2a[0][0]
____________________________________________________________________
____________________________________
res4d_branch2b (Conv2D)         (None, 8, 8, 256)    590080
    activation_32[0][0]
____________________________________________________________________
____________________________________
bn4d_branch2b (BatchNormalizati (None, 8, 8, 256)    1024
    res4d_branch2b[0][0]
____________________________________________________________________
____________________________________
activation_33 (Activation)      (None, 8, 8, 256)    0
    bn4d_branch2b[0][0]
____________________________________________________________________
```

```
____________________________________
res4d_branch2c (Conv2D)         (None, 8, 8, 1024)   263168
    activation_33[0][0]
____________________________________________________________
____________________________________
bn4d_branch2c (BatchNormalizati (None, 8, 8, 1024)   4096
    res4d_branch2c[0][0]
____________________________________________________________
____________________________________
add_11 (Add)                    (None, 8, 8, 1024)   0
    bn4d_branch2c[0][0]

    activation_31[0][0]
____________________________________________________________
____________________________________
activation_34 (Activation)      (None, 8, 8, 1024)   0
    add_11[0][0]
____________________________________________________________
____________________________________
res4e_branch2a (Conv2D)         (None, 8, 8, 256)    262400
    activation_34[0][0]
____________________________________________________________
____________________________________
bn4e_branch2a (BatchNormalizati (None, 8, 8, 256)    1024
    res4e_branch2a[0][0]
____________________________________________________________
____________________________________
activation_35 (Activation)      (None, 8, 8, 256)    0
    bn4e_branch2a[0][0]
____________________________________________________________
____________________________________
res4e_branch2b (Conv2D)         (None, 8, 8, 256)    590080
    activation_35[0][0]
____________________________________________________________
____________________________________
bn4e_branch2b (BatchNormalizati (None, 8, 8, 256)    1024
    res4e_branch2b[0][0]
____________________________________________________________
____________________________________
activation_36 (Activation)      (None, 8, 8, 256)    0
    bn4e_branch2b[0][0]
____________________________________________________________
____________________________________
res4e_branch2c (Conv2D)         (None, 8, 8, 1024)   263168
    activation_36[0][0]
____________________________________________________________
____________________________________
bn4e_branch2c (BatchNormalizati (None, 8, 8, 1024)   4096
    res4e_branch2c[0][0]
____________________________________________________________
____________________________________
add_12 (Add)                    (None, 8, 8, 1024)   0
    bn4e_branch2c[0][0]

    activation_34[0][0]
____________________________________________________________
____________________________________
activation_37 (Activation)      (None, 8, 8, 1024)   0
    add_12[0][0]
____________________________________________________________
____________________________________
res4f_branch2a (Conv2D)         (None, 8, 8, 256)    262400
    activation_37[0][0]
____________________________________________________________
____________________________________
bn4f_branch2a (BatchNormalizati (None, 8, 8, 256)    1024
```

```
    res4f_branch2a[0][0]
==================================================================
==================================
activation_38 (Activation)      (None, 8, 8, 256)    0
    bn4f_branch2a[0][0]
------------------------------------------------------------------
----------------------------------
res4f_branch2b (Conv2D)         (None, 8, 8, 256)    590080
    activation_38[0][0]
------------------------------------------------------------------
----------------------------------
bn4f_branch2b (BatchNormalizati (None, 8, 8, 256)    1024
    res4f_branch2b[0][0]
------------------------------------------------------------------
----------------------------------
activation_39 (Activation)      (None, 8, 8, 256)    0
    bn4f_branch2b[0][0]
------------------------------------------------------------------
----------------------------------
res4f_branch2c (Conv2D)         (None, 8, 8, 1024)   263168
    activation_39[0][0]
------------------------------------------------------------------
----------------------------------
bn4f_branch2c (BatchNormalizati (None, 8, 8, 1024)   4096
    res4f_branch2c[0][0]
------------------------------------------------------------------
----------------------------------
add_13 (Add)                    (None, 8, 8, 1024)   0
    bn4f_branch2c[0][0]

    activation_37[0][0]
------------------------------------------------------------------
----------------------------------
activation_40 (Activation)      (None, 8, 8, 1024)   0
    add_13[0][0]
------------------------------------------------------------------
----------------------------------
res5a_branch2a (Conv2D)         (None, 4, 4, 512)    524800
    activation_40[0][0]
------------------------------------------------------------------
----------------------------------
bn5a_branch2a (BatchNormalizati (None, 4, 4, 512)    2048
    res5a_branch2a[0][0]
------------------------------------------------------------------
----------------------------------
activation_41 (Activation)      (None, 4, 4, 512)    0
    bn5a_branch2a[0][0]
------------------------------------------------------------------
----------------------------------
res5a_branch2b (Conv2D)         (None, 4, 4, 512)    2359808
    activation_41[0][0]
------------------------------------------------------------------
----------------------------------
bn5a_branch2b (BatchNormalizati (None, 4, 4, 512)    2048
    res5a_branch2b[0][0]
------------------------------------------------------------------
----------------------------------
activation_42 (Activation)      (None, 4, 4, 512)    0
    bn5a_branch2b[0][0]
------------------------------------------------------------------
----------------------------------
res5a_branch2c (Conv2D)         (None, 4, 4, 2048)   1050624
    activation_42[0][0]
------------------------------------------------------------------
----------------------------------
res5a_branch1 (Conv2D)          (None, 4, 4, 2048)   2099200
    activation_40[0][0]
```

```
__________________________________________________________________
__________________________________
bn5a_branch2c (BatchNormalizati (None, 4, 4, 2048)   8192
    res5a_branch2c[0][0]
__________________________________________________________________
__________________________________
bn5a_branch1 (BatchNormalizatio (None, 4, 4, 2048)   8192
    res5a_branch1[0][0]
__________________________________________________________________
__________________________________
add_14 (Add)                    (None, 4, 4, 2048)   0
    bn5a_branch2c[0][0]

    bn5a_branch1[0][0]
__________________________________________________________________
__________________________________
activation_43 (Activation)      (None, 4, 4, 2048)   0
    add_14[0][0]
__________________________________________________________________
__________________________________
res5b_branch2a (Conv2D)         (None, 4, 4, 512)    1049088
    activation_43[0][0]
__________________________________________________________________
__________________________________
bn5b_branch2a (BatchNormalizati (None, 4, 4, 512)    2048
    res5b_branch2a[0][0]
__________________________________________________________________
__________________________________
activation_44 (Activation)      (None, 4, 4, 512)    0
    bn5b_branch2a[0][0]
__________________________________________________________________
__________________________________
res5b_branch2b (Conv2D)         (None, 4, 4, 512)    2359808
    activation_44[0][0]
__________________________________________________________________
__________________________________
bn5b_branch2b (BatchNormalizati (None, 4, 4, 512)    2048
    res5b_branch2b[0][0]
__________________________________________________________________
__________________________________
activation_45 (Activation)      (None, 4, 4, 512)    0
    bn5b_branch2b[0][0]
__________________________________________________________________
__________________________________
res5b_branch2c (Conv2D)         (None, 4, 4, 2048)   1050624
    activation_45[0][0]
__________________________________________________________________
__________________________________
bn5b_branch2c (BatchNormalizati (None, 4, 4, 2048)   8192
    res5b_branch2c[0][0]
__________________________________________________________________
__________________________________
add_15 (Add)                    (None, 4, 4, 2048)   0
    bn5b_branch2c[0][0]

    activation_43[0][0]
__________________________________________________________________
__________________________________
activation_46 (Activation)      (None, 4, 4, 2048)   0
    add_15[0][0]
__________________________________________________________________
__________________________________
res5c_branch2a (Conv2D)         (None, 4, 4, 512)    1049088
    activation_46[0][0]
__________________________________________________________________
__________________________________
```

```
bn5c_branch2a (BatchNormalizati (None, 4, 4, 512)    2048
    res5c_branch2a[0][0]
__________________________________________________________________
__________________________________
activation_47 (Activation)      (None, 4, 4, 512)    0
    bn5c_branch2a[0][0]
__________________________________________________________________
__________________________________
res5c_branch2b (Conv2D)         (None, 4, 4, 512)    2359808
    activation_47[0][0]
__________________________________________________________________
__________________________________
bn5c_branch2b (BatchNormalizati (None, 4, 4, 512)    2048
    res5c_branch2b[0][0]
__________________________________________________________________
__________________________________
activation_48 (Activation)      (None, 4, 4, 512)    0
    bn5c_branch2b[0][0]
__________________________________________________________________
__________________________________
res5c_branch2c (Conv2D)         (None, 4, 4, 2048)   1050624
    activation_48[0][0]
__________________________________________________________________
__________________________________
bn5c_branch2c (BatchNormalizati (None, 4, 4, 2048)   8192
    res5c_branch2c[0][0]
__________________________________________________________________
__________________________________
add_16 (Add)                    (None, 4, 4, 2048)   0
    bn5c_branch2c[0][0]

    activation_46[0][0]
__________________________________________________________________
__________________________________
activation_49 (Activation)      (None, 4, 4, 2048)   0
    add_16[0][0]
__________________________________________________________________
__________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)         0
    activation_49[0][0]
__________________________________________________________________
__________________________________
dropout_1 (Dropout)             (None, 2048)         0
    avg_pool[0][0]
__________________________________________________________________
__________________________________
dense_1 (Dense)                 (None, 5004)         10253196
    dropout_1[0][0]
==================================================================
==================================
Total params: 33,840,908
Trainable params: 33,787,788
Non-trainable params: 53,120
__________________________________________________________________
__________________________________
```

In [12]:

```
gc.collect()
```

Out[12]:

```
0
```

Load model and weights from disc

In [13]:
```
# from keras.models import load_model

# # returns a compiled model
# # identical to the previous cell
# model = load_model(MODEL)
# print("Loaded model architecture from disk")
# gc.collect()

# model.load_weights(WEIGHTS)
# print("Loaded model weights from disk")
# model.summary()

# gc.collect()
```

Train with ImageDataGenerator and flow_from_dataframe

In [14]:
```
from keras.callbacks import LambdaCallback, ModelCheckpoint

ROTATE = 20
SEED = 28
BATCH_SIZE = 100

gc.collect()

batch_gc_callback = LambdaCallback(
    on_epoch_begin=lambda epoch,logs: gc.collect())

checkpointer = ModelCheckpoint(filepath='weights.hdf5',
                               verbose=1, save_best_only=True
)

train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    rescale=1./255,
    fill_mode='nearest'
)

train_generator = train_datagen.flow_from_dataframe(
    dataframe=whales_train,
    subset = "training",
    directory=TRAIN_CROPPED_IN,
    x_col="Image",
    y_col="Id",
    has_ext=True,
    seed = SEED,
    color_mode= "rgb",
    target_size=(IMAGE_HEIGHT, IMAGE_WIDTH),
    batch_size=BATCH_SIZE,
    class_mode='categorical')

gc.collect()
```

```
Found 15697 images belonging to 5004 classes.
```

Out[14]:
```
56
```

Visualize augmented data

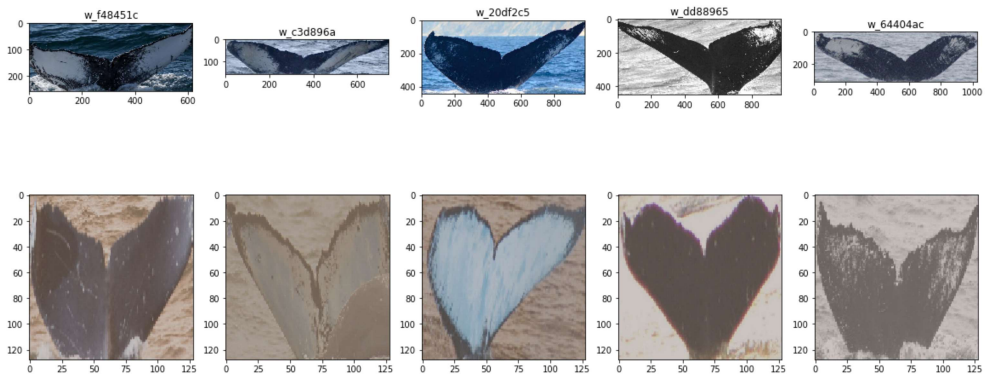
In [15]:
```
from skimage.io import imread
import PIL.Image as pimage

def plot_images(images_names, path):
    fig, m_axs = plt.subplots(1, len(images), figsize = (20,
10))
    #show the images and label them
    for ii, c_ax in enumerate(m_axs):
        img = imread(os.path.join(path,images_names[ii][0]))
        c_ax.imshow(img)
        c_ax.set_title(images_names[ii][1])

def plot_loaded_images(images_loaded, labels):
    fig, m_axs = plt.subplots(1, len(images_loaded), figsize
= (20, 10))
    #show the images and label them
    for ii, c_ax in enumerate(m_axs):
        img = pimage.fromarray(images_loaded[ii], "RGB")
        c_ax.imshow((images_loaded[ii] + 1) / 2)
#         c_ax.set_title(labels[ii])

#get the first 5 whale images
images = [(whale_img, whale_label) for (whale_img, whale_labe
l) in zip(whales_train.Image[:5], whales_train.Id[:5])]
plot_images(images, TRAIN_CROPPED_IN)

x_batch, y_batch = next(train_generator)
plot_loaded_images(x_batch[:5], y_batch[:5])
```



Learning rate finder

In [16]:
```
import keras.backend as K
from keras.callbacks import Callback


class LRFinder(Callback):
    '''
    A simple callback for finding the optimal learning rate ra
nge for your model + dataset.

    # Usage
        ```python
            lr_finder = LRFinder(min_lr=1e-5,
                                 max_lr=1e-2,
                                 steps_per_epoch=np.ceil(epoch
_size/batch_size),
                                 epochs=3)
```
```

```
            model.fit(X_train, Y_train, callbacks=[lr_finder])

            lr_finder.plot_loss()
        ```

    # Arguments
        min_lr: The lower bound of the learning rate range for the experiment.
        max_lr: The upper bound of the learning rate range for the experiment.
        steps_per_epoch: Number of mini-batches in the dataset. Calculated as `np.ceil(epoch_size/batch_size)`.
        epochs: Number of epochs to run experiment. Usually between 2 and 4 epochs is sufficient.

    # References
        Blog post: jeremyjordan.me/nn-learning-rate
        Original paper: https://arxiv.org/abs/1506.01186
    '''

    def __init__(self, min_lr=1e-5, max_lr=1e-1, steps_per_epoch=None, epochs=None, beta=.98):
        super().__init__()

        self.min_lr = min_lr
        self.max_lr = max_lr
        self.total_iterations = steps_per_epoch * epochs
        self.iteration = 0
        self.history = {}
        self.beta = beta
        self.lr_mult = (max_lr/min_lr)**(1/steps_per_epoch)

    def clr(self):
        '''Calculate the learning rate.'''
        x = self.iteration / self.total_iterations
        return self.min_lr * (self.lr_mult**self.iteration)

    def on_train_begin(self, logs=None):
        '''Initialize the learning rate to the minimum value at the start of training.'''
        self.best_loss = 1e9
        self.avg_loss = 0
        self.losses, self.smoothed_losses, self.lrs, self.iterations = [], [], [], []
        logs = logs or {}
        K.set_value(self.model.optimizer.lr, self.min_lr)

    def on_epoch_end(self, epoch, logs=None):
        print(" Epoch ended learning rate: " + str(K.get_value(self.model.optimizer.lr)))

    def on_epoch_begin(self, epoch, logs=None):
        print(" Epoch start learning rate: " + str(K.get_value(self.model.optimizer.lr)))

    def on_batch_end(self, epoch, logs=None):
        '''Record previous batch statistics and update the learning rate.'''
        logs = logs or {}
        self.iteration += 1

        #addition
        loss = logs.get('loss')
        self.avg_loss = self.beta * self.avg_loss + (1 - self.beta) * loss
        smoothed_loss = self.avg_loss / (1 - self.beta**self.
```
```

```
iteration)
        # Check if the loss is not exploding
        if self.iteration>1 and smoothed_loss > self.best_los
s * 1.5:
            self.model.stop_training = True
            return
        if smoothed_loss < self.best_loss or self.iteration==
1:
            self.best_loss = smoothed_loss

        lr = self.clr()
        self.losses.append(loss)
        self.smoothed_losses.append(smoothed_loss)
        self.lrs.append(lr)
        self.iterations.append(self.iteration)

        self.history.setdefault('lr', []).append(K.get_value(
self.model.optimizer.lr))
        self.history.setdefault('iterations', []).append(self
.iteration)

        for k, v in logs.items():
            self.history.setdefault(k, []).append(v)

        K.set_value(self.model.optimizer.lr,lr)

#     def plot_lr(self):
#         '''Helper function to quickly inspect the learning r
ate schedule.'''
#         plt.plot(self.history['iterations'], self.history['l
r'])
#         plt.yscale('log')
#         plt.xlabel('Iteration')
#         plt.ylabel('Learning rate')
#         plt.show()

    def plot_lr(self):
        plt.xlabel('Iterations')
        plt.ylabel('Learning rate')
        plt.plot(self.iterations, self.lrs)
        plt.yscale('log')
        plt.show()

    def plot_loss(self):
        '''Helper function to quickly observe the learning rat
e experiment results.'''
        plt.plot(self.history['lr'], self.history['loss'])
        plt.xscale('log')
        plt.xlabel('Learning rate')
        plt.ylabel('Loss')
        plt.show()

    def plot(self, n_skip=10):
        plt.ylabel('Loss')
        plt.xlabel('Learning rate (log scale)')
        plt.plot(self.lrs[n_skip:-5], self.losses[n_skip:-5])
        plt.xscale('log')

    def plot_loss2(self):
        plt.ylabel('Losses')
        plt.xlabel('Iterations')
        plt.plot(self.iterations[10:], self.losses[10:])

    def plot_smoothed_loss(self, n_skip=6):
        plt.ylabel('Smoothed Losses')
        plt.xlabel('Learning rate (log scale)')
```

```
        plt.plot(self.lrs[n_skip:-5], self.smoothed_losses[n_
skip:-5])
        plt.xscale('log')

    def plot_loss_change(self, sma=1, n_skip=20, y_lim=(-0.01
,0.01), should_lim_y = False):
        """
        Plots rate of change of the loss function.
        Parameters:
            sched - learning rate scheduler, an instance of LR
_Finder class.
            sma - number of batches for simple moving average
 to smooth out the curve.
            n_skip - number of batches to skip on the left.
            y_lim - limits for the y axis.
        """
        derivatives = [0] * (sma + 1)
        for i in range(1 + sma, len(self.lrs)):
            derivative = (self.losses[i] - self.losses[i - sm
a]) / sma
            derivatives.append(derivative)

        plt.ylabel("d/loss")
        plt.xlabel("learning rate (log scale)")
        plt.plot(self.lrs[n_skip:], derivatives[n_skip:])
        plt.xscale('log')
        if should_lim_y:
            plt.ylim(y_lim)
```

Train

In [17]:


```
lr_finder = LRFinder(min_lr=1e-6,
                     max_lr=1,
                     steps_per_epoch=np.ceil(train_generator.
n//train_generator.batch_size),
                     epochs=EPOCHS)

# Train the loaded model for more epochs
history = model.fit_generator(generator=train_generator,
                              steps_per_epoch=STEP_SIZE_TRAIN
,
                              epochs=EPOCHS,
                              callbacks=[batch_gc_callback, l
r_finder])
```

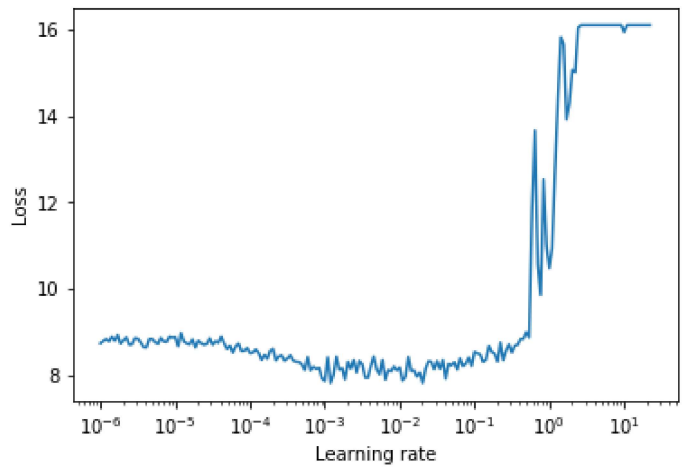


```
Epoch 1/5
 Epoch start learning rate: 1e-06
156/156 [==============================] - 165s 1s/step - los
s: 8.5787 - acc: 0.0049 - mean_absolute_error: 3.9936e-04 - t
op_5_accuracy: 0.0171
 Epoch ended learning rate: 1.0
Epoch 2/5
 Epoch start learning rate: 1.0
 37/156 [======>.......................] - ETA: 1:47 - loss:
15.4974 - acc: 2.7027e-04 - mean_absolute_error: 3.9955e-04 -
top_5_accuracy: 0.7973 Epoch ended learning rate: 24.24462
```

In [18]:

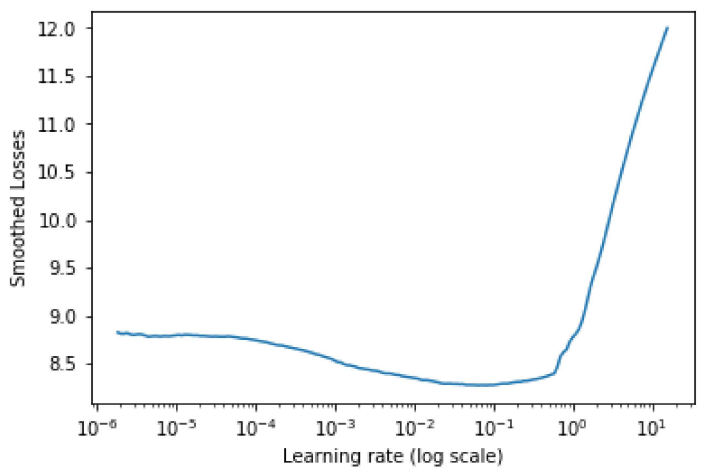
```
lr_finder.plot_loss()
```

```
lr_finder.plot_loss()
```



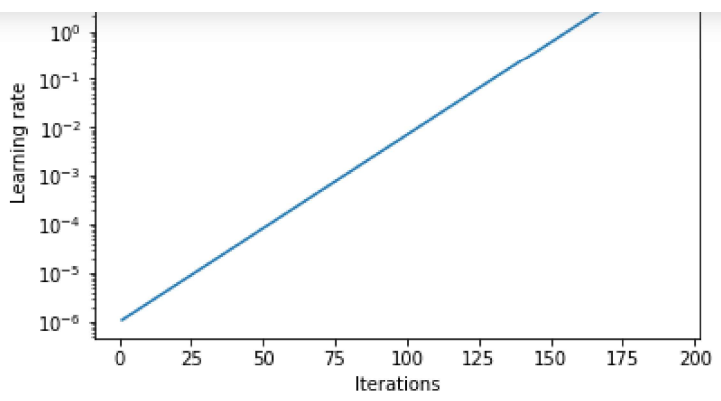
In [19]:

```
lr_finder.plot_smoothed_loss()
```



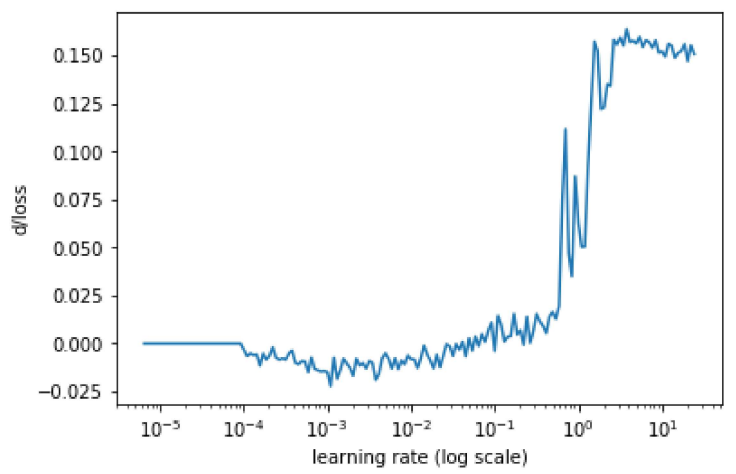
In [20]:

```
lr_finder.plot_lr()
```

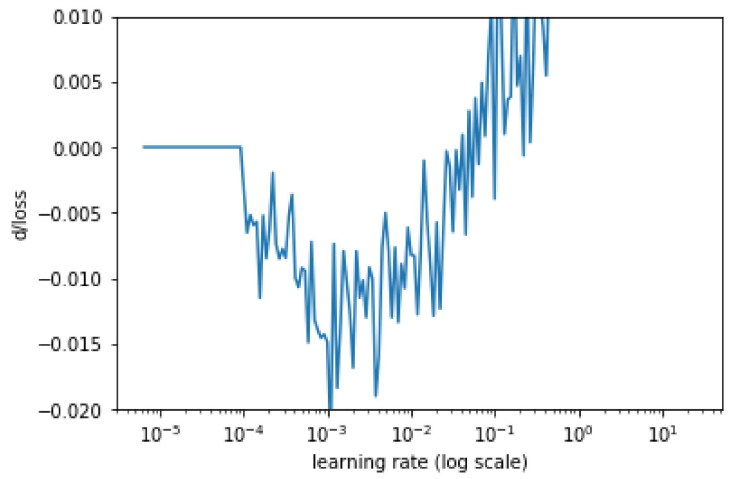



In [21]:

```
lr_finder.plot_loss_change(sma=50)
```



In [22]:
```
lr_finder.plot_loss_change(sma=50, y_lim=(-0.02,0.01), should
_lim_y=True)
```



Plot train results

In [23]:
```
def plot_accuracy(history, should_plot_val = False, should_pl
ot_top5 = False):
    acc = history.history['acc']
    l1 = plt.plot(acc, label='acc')

    if should_plot_val:
        val_acc = history.history['val_acc']
        l2 = plt.plot(val_acc, label='val_acc')

    if should_plot_top5:
        top5_acc = history.history['top_5_accuracy']
        l3 = plt.plot(top5_acc, label='top_5_accuracy')

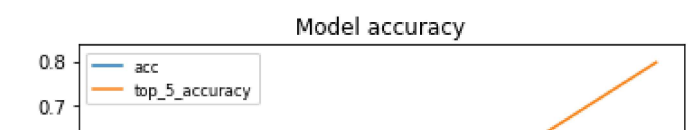
    plt.legend(loc=2, fontsize="small")
    plt.title('Model accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.show()

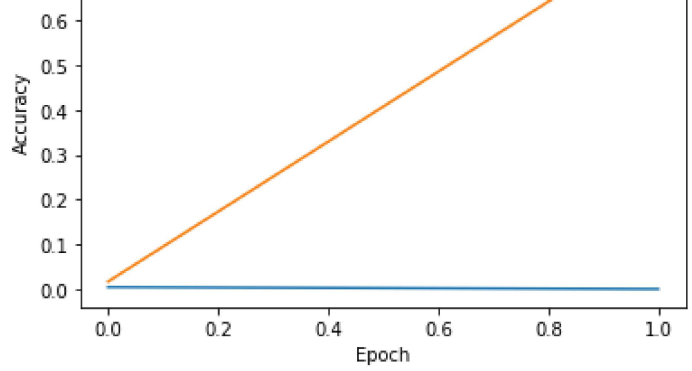
def plot_loss(history, should_plot_val = False):
    loss = history.history['loss']
    l1 = plt.plot(loss, label='loss')

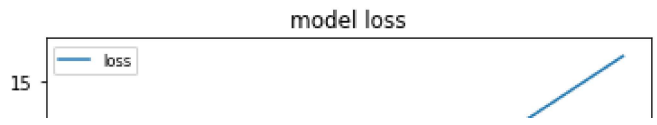
    if should_plot_val:
        val_loss = history.history['val_loss']
        plt.plot(val_loss, label='val_loss')

    plt.legend(loc=2, fontsize="small")
    plt.title('model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.show()
```

In [24]:
```
plot_accuracy(history, False, True)
plot_loss(history, False)
```







Did you find this Kernel useful?
Show your appreciation with an upvote

0

Data

Data Sources

- Humpback Whale I...
  - sa... 7960 x 2
  - trai... 25.4k x 2
  - test.zip
    - 0027089a4.jpg
    - 00313e2d2.jpg
    - 004344e9f.jpg
    - 008a4bc86.jpg
    - 00ac0fca6.jpg
    - 00ff45291.jpg
    - 012dbdb59.jpg
    - 0169cec0e.jpg
    - 01830c9cf.jpg
    - 01b1ecf7b.jpg
    - 1000+ more
  - train.zip
    - 002b4615d.jpg
    - 00600ce17.jpg
    - 00d641885.jpg
    - 00eaedfab.jpg
    - 00fee3975.jpg
    - 010a1f0eb.jpg
    - 01237f1ce.jpg
    - 01dcb420f.jpg
    - 0202dfb29.jpg
    - 020ab0f9b.jpg
    - 1000+ more

Humpback Whale Identification

Can you identify a whale by its tail?

Last Updated: 2 months ago

About this Competition

This training data contains thousands of images of humpback whale flukes. Individual whales have been identified by researchers and given an `Id`. The challenge is to predict the whale `Id` of images in the test set. What makes this such a challenge is that there are only a few examples for each of 3,000+ whale Ids.

File descriptions

- **train.zip -** a folder containing the training images
- **train.csv** - maps the training `Image` to the appropriate whale `Id`. Whales that are not predicted to have a label identified in the training data should be labeled as `new_whale`.
- **test.zip -** a folder containing the test images to predict the whale `Id`
- **sample_submission.csv** - a sample submission file in the correct format

Output Files

New Dataset | New Kernel | Download All

Output Files

- Model_ResNet50.h5
- Weights_ResNet50....

About this file

This file was created from a Kernel, it does not have a description.

Model_ResNet50.h5

We don't support previews for this file yet

Run Info

Succeeded	True	Run Time	277.9 seconds
Exit Code	0	Queue Time	0 seconds
Docker Image Name	/python(Dockerfile)	Output Size	0
Timeout Exceeded	False	Used All Space	False
Failure Message			

Log

Download Log

```
Time  Line #  Log Message
2.7s       1  [NbConvertApp] Converting notebook __notebook__.ipynb to
              notebook
2.8s       2  [NbConvertApp] Executing notebook with kernel: python3
7.4s       3  2019-02-02 20:46:15.098146: I
              tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:964]
              successful NUMA node read from SysFS had negative value (-1),
              but there must be at least one NUMA node, so returning NUMA
              node zero
7.4s       4  2019-02-02 20:46:15.104368: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:1432] Found
              device 0 with properties:
              name: Tesla K80 major: 3 minor: 7 memoryClockRate(GHz): 0.8235
              pciBusID: 0000:00:04.0
              totalMemory: 11.17GiB freeMemory: 11.10GiB
              2019-02-02 20:46:15.104406: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:1511] Adding
              visible gpu devices: 0
7.9s       5  2019-02-02 20:46:15.573820: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:982] Device
              interconnect StreamExecutor with strength 1 edge matrix:
7.9s       6  2019-02-02 20:46:15.573966: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:988]      0
              2019-02-02 20:46:15.574070: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:1001] 0:   N
              2019-02-02 20:46:15.574559: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created
              TensorFlow device
              (/job:localhost/replica:0/task:0/device:GPU:0 with 10758 MB
              memory) -> physical GPU (device: 0, name: Tesla K80, pci bus
              id: 0000:00:04.0, compute capability: 3.7)
274.7s     7  [NbConvertApp] Writing 718399 bytes to __notebook__.ipynb
276.9s     8  [NbConvertApp] Converting notebook __notebook__.ipynb to html
277.5s     9  [NbConvertApp] Support files will be in __results___files/
277.5s    10  [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Writing 365882 bytes to __results__.html
277.5s    11
277.5s    13  Complete. Exited with code 0.
```

Comments (0)

Click here to enter a comment...