In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.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.read_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_submission.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_Xception.h5'
WEIGHTS_F = 'Weights_Xception.h5'
MODEL = '../input/Xception/'+ MODEL_F
WEIGHTS = '../input/Xception/'+ WEIGHTS_F
```

In [4]:
```
train.describe()
```

Out[4]:

| | Image | Id |
|---|---|---|
| count | 25361 | 25361 |
| unique | 25361 | 5005 |
| top | e5982ded0.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
b576308d0.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 | b4284d5ef.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 ImageDataGene
rator

# from keras.applications.imagenet_utils import prepr
ocess_input
# from keras.applications.resnet50 import ResNet50, p
reprocess_input
from keras.applications.xception import Xception, p
reprocess_input
# from keras.applications.inception_resnet_v2 import
 InceptionResNetV2, preprocess_input

from keras.losses import binary_crossentropy
from keras.preprocessing.image import ImageDataGene
```

```
fator
from keras.layers import Input, Dense, Activation,
BatchNormalization, Flatten, Conv2D, GlobalAverageP
ooling2D
from keras.layers import AveragePooling2D, MaxPooli
ng2D, Dropout
from keras.models import Model
from keras.metrics import top_k_categorical_accurac
y

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

%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_pre
d, k=5)

# setup model
base_model = Xception(weights='imagenet', include_t
op=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=pred
ictions)

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

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy', 'mae', top_5_acc
uracy])

model.summary()
```

```
Downloading data from https://github.com/fchollet/d
eep-learning-models/releases/download/v0.4/xception
```

```
_weights_tf_dim_ordering_tf_kernels_notop.h5
83689472/83683744 [==============================]
- 2s 0us/step
_________________________________________________
_____________________________________________
Layer (type)                 Output Shape
  Param #     Connected to
=================================================
=============================================
input_1 (InputLayer)         (None, 128, 128, 3)
  0
_________________________________________________
_____________________________________________
block1_conv1 (Conv2D)        (None, 63, 63, 32)
  864         input_1[0][0]
_________________________________________________
_____________________________________________
block1_conv1_bn (BatchNormaliza (None, 63, 63, 32)
  128         block1_conv1[0][0]
_________________________________________________
_____________________________________________
block1_conv1_act (Activation)   (None, 63, 63, 32)
  0           block1_conv1_bn[0][0]
_________________________________________________
_____________________________________________
block1_conv2 (Conv2D)        (None, 61, 61, 64)
  18432       block1_conv1_act[0][0]
_________________________________________________
_____________________________________________
block1_conv2_bn (BatchNormaliza (None, 61, 61, 64)
  256         block1_conv2[0][0]
_________________________________________________
_____________________________________________
block1_conv2_act (Activation)   (None, 61, 61, 64)
  0           block1_conv2_bn[0][0]
_________________________________________________
_____________________________________________
block2_sepconv1 (SeparableConv2 (None, 61, 61, 128)
  8768        block1_conv2_act[0][0]
_________________________________________________
_____________________________________________
block2_sepconv1_bn (BatchNormal (None, 61, 61, 128)
  512         block2_sepconv1[0][0]
_________________________________________________
_____________________________________________
block2_sepconv2_act (Activation (None, 61, 61, 128)
  0           block2_sepconv1_bn[0][0]
_________________________________________________
_____________________________________________
block2_sepconv2 (SeparableConv2 (None, 61, 61, 128)
  17536       block2_sepconv2_act[0][0]
_________________________________________________
_____________________________________________
block2_sepconv2_bn (BatchNormal (None, 61, 61, 128)
  512         block2_sepconv2[0][0]
_________________________________________________
_____________________________________________
conv2d_1 (Conv2D)            (None, 31, 31, 128)
  8192        block1_conv2_act[0][0]
```

```
_________________________________________________
_________________________________________________
block2_pool (MaxPooling2D)      (None, 31, 31, 128)
  0           block2_sepconv2_bn[0][0]
_________________________________________________
_________________________________________________
batch_normalization_1 (BatchNor (None, 31, 31, 128)
  512         conv2d_1[0][0]
_________________________________________________
_________________________________________________
add_1 (Add)                     (None, 31, 31, 128)
  0           block2_pool[0][0]

              batch_normalization_1[0][0]
_________________________________________________
_________________________________________________
block3_sepconv1_act (Activation (None, 31, 31, 128)
  0           add_1[0][0]
_________________________________________________
_________________________________________________
block3_sepconv1 (SeparableConv2 (None, 31, 31, 256)
  33920       block3_sepconv1_act[0][0]
_________________________________________________
_________________________________________________
block3_sepconv1_bn (BatchNormal (None, 31, 31, 256)
  1024        block3_sepconv1[0][0]
_________________________________________________
_________________________________________________
block3_sepconv2_act (Activation (None, 31, 31, 256)
  0           block3_sepconv1_bn[0][0]
_________________________________________________
_________________________________________________
block3_sepconv2 (SeparableConv2 (None, 31, 31, 256)
  67840       block3_sepconv2_act[0][0]
_________________________________________________
_________________________________________________
block3_sepconv2_bn (BatchNormal (None, 31, 31, 256)
  1024        block3_sepconv2[0][0]
_________________________________________________
_________________________________________________
conv2d_2 (Conv2D)               (None, 16, 16, 256)
  32768       add_1[0][0]
_________________________________________________
_________________________________________________
block3_pool (MaxPooling2D)      (None, 16, 16, 256)
  0           block3_sepconv2_bn[0][0]
_________________________________________________
_________________________________________________
batch_normalization_2 (BatchNor (None, 16, 16, 256)
  1024        conv2d_2[0][0]
_________________________________________________
_________________________________________________
add_2 (Add)                     (None, 16, 16, 256)
  0           block3_pool[0][0]

              batch_normalization_2[0][0]
_________________________________________________
_________________________________________________
block4_sepconv1_act (Activation (None, 16, 16, 256)
  0           add_2[0][0]
```

```
__________________________________________________
_____________________________________
block4_sepconv1 (SeparableConv2 (None, 16, 16, 728)
  188672      block4_sepconv1_act[0][0]
__________________________________________________
_____________________________________
block4_sepconv1_bn (BatchNormal (None, 16, 16, 728)
  2912        block4_sepconv1[0][0]
__________________________________________________
_____________________________________
block4_sepconv2_act (Activation (None, 16, 16, 728)
  0           block4_sepconv1_bn[0][0]
__________________________________________________
_____________________________________
block4_sepconv2 (SeparableConv2 (None, 16, 16, 728)
  536536      block4_sepconv2_act[0][0]
__________________________________________________
_____________________________________
block4_sepconv2_bn (BatchNormal (None, 16, 16, 728)
  2912        block4_sepconv2[0][0]
__________________________________________________
_____________________________________
conv2d_3 (Conv2D)               (None, 8, 8, 728)
  186368      add_2[0][0]
__________________________________________________
_____________________________________
block4_pool (MaxPooling2D)      (None, 8, 8, 728)
  0           block4_sepconv2_bn[0][0]
__________________________________________________
_____________________________________
batch_normalization_3 (BatchNor (None, 8, 8, 728)
  2912        conv2d_3[0][0]
__________________________________________________
_____________________________________
add_3 (Add)                     (None, 8, 8, 728)
  0           block4_pool[0][0]

              batch_normalization_3[0][0]
__________________________________________________
_____________________________________
block5_sepconv1_act (Activation (None, 8, 8, 728)
  0           add_3[0][0]
__________________________________________________
_____________________________________
block5_sepconv1 (SeparableConv2 (None, 8, 8, 728)
  536536      block5_sepconv1_act[0][0]
__________________________________________________
_____________________________________
block5_sepconv1_bn (BatchNormal (None, 8, 8, 728)
  2912        block5_sepconv1[0][0]
__________________________________________________
_____________________________________
block5_sepconv2_act (Activation (None, 8, 8, 728)
  0           block5_sepconv1_bn[0][0]
__________________________________________________
_____________________________________
block5_sepconv2 (SeparableConv2 (None, 8, 8, 728)
  536536      block5_sepconv2_act[0][0]
__________________________________________________
_____________________________________
```

```
block5_sepconv2_bn (BatchNormal (None, 8, 8, 728)
  2912        block5_sepconv2[0][0]
__________________________________________________
_______________________________________________
block5_sepconv3_act (Activation (None, 8, 8, 728)
  0           block5_sepconv2_bn[0][0]
__________________________________________________
_______________________________________________
block5_sepconv3 (SeparableConv2 (None, 8, 8, 728)
  536536      block5_sepconv3_act[0][0]
__________________________________________________
_______________________________________________
block5_sepconv3_bn (BatchNormal (None, 8, 8, 728)
  2912        block5_sepconv3[0][0]
__________________________________________________
_______________________________________________
add_4 (Add)                     (None, 8, 8, 728)
  0           block5_sepconv3_bn[0][0]

              add_3[0][0]
__________________________________________________
_______________________________________________
block6_sepconv1_act (Activation (None, 8, 8, 728)
  0           add_4[0][0]
__________________________________________________
_______________________________________________
block6_sepconv1 (SeparableConv2 (None, 8, 8, 728)
  536536      block6_sepconv1_act[0][0]
__________________________________________________
_______________________________________________
block6_sepconv1_bn (BatchNormal (None, 8, 8, 728)
  2912        block6_sepconv1[0][0]
__________________________________________________
_______________________________________________
block6_sepconv2_act (Activation (None, 8, 8, 728)
  0           block6_sepconv1_bn[0][0]
__________________________________________________
_______________________________________________
block6_sepconv2 (SeparableConv2 (None, 8, 8, 728)
  536536      block6_sepconv2_act[0][0]
__________________________________________________
_______________________________________________
block6_sepconv2_bn (BatchNormal (None, 8, 8, 728)
  2912        block6_sepconv2[0][0]
__________________________________________________
_______________________________________________
block6_sepconv3_act (Activation (None, 8, 8, 728)
  0           block6_sepconv2_bn[0][0]
__________________________________________________
_______________________________________________
block6_sepconv3 (SeparableConv2 (None, 8, 8, 728)
  536536      block6_sepconv3_act[0][0]
__________________________________________________
_______________________________________________
block6_sepconv3_bn (BatchNormal (None, 8, 8, 728)
  2912        block6_sepconv3[0][0]
__________________________________________________
_______________________________________________
add_5 (Add)                     (None, 8, 8, 728)
  0           block6_sepconv3_bn[0][0]
```

```
                add_4[0][0]
==================================================
==============================================
block7_sepconv1_act (Activation (None, 8, 8, 728)
  0           add_5[0][0]
--------------------------------------------------
----------------------------------------------
block7_sepconv1 (SeparableConv2 (None, 8, 8, 728)
  536536      block7_sepconv1_act[0][0]
--------------------------------------------------
----------------------------------------------
block7_sepconv1_bn (BatchNormal (None, 8, 8, 728)
  2912        block7_sepconv1[0][0]
--------------------------------------------------
----------------------------------------------
block7_sepconv2_act (Activation (None, 8, 8, 728)
  0           block7_sepconv1_bn[0][0]
--------------------------------------------------
----------------------------------------------
block7_sepconv2 (SeparableConv2 (None, 8, 8, 728)
  536536      block7_sepconv2_act[0][0]
--------------------------------------------------
----------------------------------------------
block7_sepconv2_bn (BatchNormal (None, 8, 8, 728)
  2912        block7_sepconv2[0][0]
--------------------------------------------------
----------------------------------------------
block7_sepconv3_act (Activation (None, 8, 8, 728)
  0           block7_sepconv2_bn[0][0]
--------------------------------------------------
----------------------------------------------
block7_sepconv3 (SeparableConv2 (None, 8, 8, 728)
  536536      block7_sepconv3_act[0][0]
--------------------------------------------------
----------------------------------------------
block7_sepconv3_bn (BatchNormal (None, 8, 8, 728)
  2912        block7_sepconv3[0][0]
--------------------------------------------------
----------------------------------------------
add_6 (Add)                     (None, 8, 8, 728)
  0           block7_sepconv3_bn[0][0]

              add_5[0][0]
--------------------------------------------------
----------------------------------------------
block8_sepconv1_act (Activation (None, 8, 8, 728)
  0           add_6[0][0]
--------------------------------------------------
----------------------------------------------
block8_sepconv1 (SeparableConv2 (None, 8, 8, 728)
  536536      block8_sepconv1_act[0][0]
--------------------------------------------------
----------------------------------------------
block8_sepconv1_bn (BatchNormal (None, 8, 8, 728)
  2912        block8_sepconv1[0][0]
--------------------------------------------------
----------------------------------------------
block8_sepconv2_act (Activation (None, 8, 8, 728)
  0           block8_sepconv1_bn[0][0]
```

```
____________________________________________________
____________________________________________________
block8_sepconv2 (SeparableConv2 (None, 8, 8, 728)
  536536      block8_sepconv2_act[0][0]
____________________________________________________
____________________________________________________
block8_sepconv2_bn (BatchNormal (None, 8, 8, 728)
  2912        block8_sepconv2[0][0]
____________________________________________________
____________________________________________________
block8_sepconv3_act (Activation (None, 8, 8, 728)
  0           block8_sepconv2_bn[0][0]
____________________________________________________
____________________________________________________
block8_sepconv3 (SeparableConv2 (None, 8, 8, 728)
  536536      block8_sepconv3_act[0][0]
____________________________________________________
____________________________________________________
block8_sepconv3_bn (BatchNormal (None, 8, 8, 728)
  2912        block8_sepconv3[0][0]
____________________________________________________
____________________________________________________
add_7 (Add)                     (None, 8, 8, 728)
  0           block8_sepconv3_bn[0][0]

              add_6[0][0]
____________________________________________________
____________________________________________________
block9_sepconv1_act (Activation (None, 8, 8, 728)
  0           add_7[0][0]
____________________________________________________
____________________________________________________
block9_sepconv1 (SeparableConv2 (None, 8, 8, 728)
  536536      block9_sepconv1_act[0][0]
____________________________________________________
____________________________________________________
block9_sepconv1_bn (BatchNormal (None, 8, 8, 728)
  2912        block9_sepconv1[0][0]
____________________________________________________
____________________________________________________
block9_sepconv2_act (Activation (None, 8, 8, 728)
  0           block9_sepconv1_bn[0][0]
____________________________________________________
____________________________________________________
block9_sepconv2 (SeparableConv2 (None, 8, 8, 728)
  536536      block9_sepconv2_act[0][0]
____________________________________________________
____________________________________________________
block9_sepconv2_bn (BatchNormal (None, 8, 8, 728)
  2912        block9_sepconv2[0][0]
____________________________________________________
____________________________________________________
block9_sepconv3_act (Activation (None, 8, 8, 728)
  0           block9_sepconv2_bn[0][0]
____________________________________________________
____________________________________________________
block9_sepconv3 (SeparableConv2 (None, 8, 8, 728)
  536536      block9_sepconv3_act[0][0]
____________________________________________________
____________________________________________________
```

```
block9_sepconv3_bn (BatchNormal (None, 8, 8, 728)
  2912        block9_sepconv3[0][0]
__________________________________________________
_______________________________________________
add_8 (Add)                     (None, 8, 8, 728)
  0           block9_sepconv3_bn[0][0]

              add_7[0][0]
__________________________________________________
_______________________________________________
block10_sepconv1_act (Activatio (None, 8, 8, 728)
  0           add_8[0][0]
__________________________________________________
_______________________________________________
block10_sepconv1 (SeparableConv (None, 8, 8, 728)
  536536      block10_sepconv1_act[0][0]
__________________________________________________
_______________________________________________
block10_sepconv1_bn (BatchNorma (None, 8, 8, 728)
  2912        block10_sepconv1[0][0]
__________________________________________________
_______________________________________________
block10_sepconv2_act (Activatio (None, 8, 8, 728)
  0           block10_sepconv1_bn[0][0]
__________________________________________________
_______________________________________________
block10_sepconv2 (SeparableConv (None, 8, 8, 728)
  536536      block10_sepconv2_act[0][0]
__________________________________________________
_______________________________________________
block10_sepconv2_bn (BatchNorma (None, 8, 8, 728)
  2912        block10_sepconv2[0][0]
__________________________________________________
_______________________________________________
block10_sepconv3_act (Activatio (None, 8, 8, 728)
  0           block10_sepconv2_bn[0][0]
__________________________________________________
_______________________________________________
block10_sepconv3 (SeparableConv (None, 8, 8, 728)
  536536      block10_sepconv3_act[0][0]
__________________________________________________
_______________________________________________
block10_sepconv3_bn (BatchNorma (None, 8, 8, 728)
  2912        block10_sepconv3[0][0]
__________________________________________________
_______________________________________________
add_9 (Add)                     (None, 8, 8, 728)
  0           block10_sepconv3_bn[0][0]

              add_8[0][0]
__________________________________________________
_______________________________________________
block11_sepconv1_act (Activatio (None, 8, 8, 728)
  0           add_9[0][0]
__________________________________________________
_______________________________________________
block11_sepconv1 (SeparableConv (None, 8, 8, 728)
  536536      block11_sepconv1_act[0][0]
__________________________________________________
```

```
block11_sepconv1_bn (BatchNorma (None, 8, 8, 728)
  2912        block11_sepconv1[0][0]
__________________________________________________
__________________________________________________
block11_sepconv2_act (Activatio (None, 8, 8, 728)
  0           block11_sepconv1_bn[0][0]
__________________________________________________
__________________________________________________
block11_sepconv2 (SeparableConv (None, 8, 8, 728)
  536536      block11_sepconv2_act[0][0]
__________________________________________________
__________________________________________________
block11_sepconv2_bn (BatchNorma (None, 8, 8, 728)
  2912        block11_sepconv2[0][0]
__________________________________________________
__________________________________________________
block11_sepconv3_act (Activatio (None, 8, 8, 728)
  0           block11_sepconv2_bn[0][0]
__________________________________________________
__________________________________________________
block11_sepconv3 (SeparableConv (None, 8, 8, 728)
  536536      block11_sepconv3_act[0][0]
__________________________________________________
__________________________________________________
block11_sepconv3_bn (BatchNorma (None, 8, 8, 728)
  2912        block11_sepconv3[0][0]
__________________________________________________
__________________________________________________
add_10 (Add)                    (None, 8, 8, 728)
  0           block11_sepconv3_bn[0][0]
              add_9[0][0]
__________________________________________________
__________________________________________________
block12_sepconv1_act (Activatio (None, 8, 8, 728)
  0           add_10[0][0]
__________________________________________________
__________________________________________________
block12_sepconv1 (SeparableConv (None, 8, 8, 728)
  536536      block12_sepconv1_act[0][0]
__________________________________________________
__________________________________________________
block12_sepconv1_bn (BatchNorma (None, 8, 8, 728)
  2912        block12_sepconv1[0][0]
__________________________________________________
__________________________________________________
block12_sepconv2_act (Activatio (None, 8, 8, 728)
  0           block12_sepconv1_bn[0][0]
__________________________________________________
__________________________________________________
block12_sepconv2 (SeparableConv (None, 8, 8, 728)
  536536      block12_sepconv2_act[0][0]
__________________________________________________
__________________________________________________
block12_sepconv2_bn (BatchNorma (None, 8, 8, 728)
  2912        block12_sepconv2[0][0]
__________________________________________________
__________________________________________________
block12_sepconv3_act (Activatio (None, 8, 8, 728)
```

```
  0           block12_sepconv2_bn[0][0]
==================================================
================================================
block12_sepconv3 (SeparableConv (None, 8, 8, 728)
  536536      block12_sepconv3_act[0][0]
--------------------------------------------------
------------------------------------------------
block12_sepconv3_bn (BatchNorma (None, 8, 8, 728)
  2912        block12_sepconv3[0][0]
--------------------------------------------------
------------------------------------------------
add_11 (Add)                    (None, 8, 8, 728)
  0           block12_sepconv3_bn[0][0]

              add_10[0][0]
--------------------------------------------------
------------------------------------------------
block13_sepconv1_act (Activatio (None, 8, 8, 728)
  0           add_11[0][0]
--------------------------------------------------
------------------------------------------------
block13_sepconv1 (SeparableConv (None, 8, 8, 728)
  536536      block13_sepconv1_act[0][0]
--------------------------------------------------
------------------------------------------------
block13_sepconv1_bn (BatchNorma (None, 8, 8, 728)
  2912        block13_sepconv1[0][0]
--------------------------------------------------
------------------------------------------------
block13_sepconv2_act (Activatio (None, 8, 8, 728)
  0           block13_sepconv1_bn[0][0]
--------------------------------------------------
------------------------------------------------
block13_sepconv2 (SeparableConv (None, 8, 8, 1024)
  752024      block13_sepconv2_act[0][0]
--------------------------------------------------
------------------------------------------------
block13_sepconv2_bn (BatchNorma (None, 8, 8, 1024)
  4096        block13_sepconv2[0][0]
--------------------------------------------------
------------------------------------------------
conv2d_4 (Conv2D)               (None, 4, 4, 1024)
  745472      add_11[0][0]
--------------------------------------------------
------------------------------------------------
block13_pool (MaxPooling2D)     (None, 4, 4, 1024)
  0           block13_sepconv2_bn[0][0]
--------------------------------------------------
------------------------------------------------
batch_normalization_4 (BatchNor (None, 4, 4, 1024)
  4096        conv2d_4[0][0]
--------------------------------------------------
------------------------------------------------
add_12 (Add)                    (None, 4, 4, 1024)
  0           block13_pool[0][0]

              batch_normalization_4[0][0]
--------------------------------------------------
------------------------------------------------
block14_sepconv1 (SeparableConv (None, 4, 4, 1536)
```

```
  1582080     add_12[0][0]
__________________________________________________
_______________________________________________
block14_sepconv1_bn (BatchNorma (None, 4, 4, 1536)
  6144        block14_sepconv1[0][0]
__________________________________________________
_______________________________________________
block14_sepconv1_act (Activatio (None, 4, 4, 1536)
  0           block14_sepconv1_bn[0][0]
__________________________________________________
_______________________________________________
block14_sepconv2 (SeparableConv (None, 4, 4, 2048)
  3159552     block14_sepconv1_act[0][0]
__________________________________________________
_______________________________________________
block14_sepconv2_bn (BatchNorma (None, 4, 4, 2048)
  8192        block14_sepconv2[0][0]
__________________________________________________
_______________________________________________
block14_sepconv2_act (Activatio (None, 4, 4, 2048)
  0           block14_sepconv2_bn[0][0]
__________________________________________________
_______________________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)
  0           block14_sepconv2_act[0][0]
__________________________________________________
_______________________________________________
dropout_1 (Dropout)             (None, 2048)
  0           avg_pool[0][0]
__________________________________________________
_______________________________________________
dense_1 (Dense)                 (None, 5004)
  10253196    dropout_1[0][0]
==================================================
===============================================
Total params: 31,114,676
Trainable params: 31,060,148
Non-trainable params: 54,528
__________________________________________________
_______________________________________________
```

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, ModelCh
eckpoint

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.hd
f5',
                               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

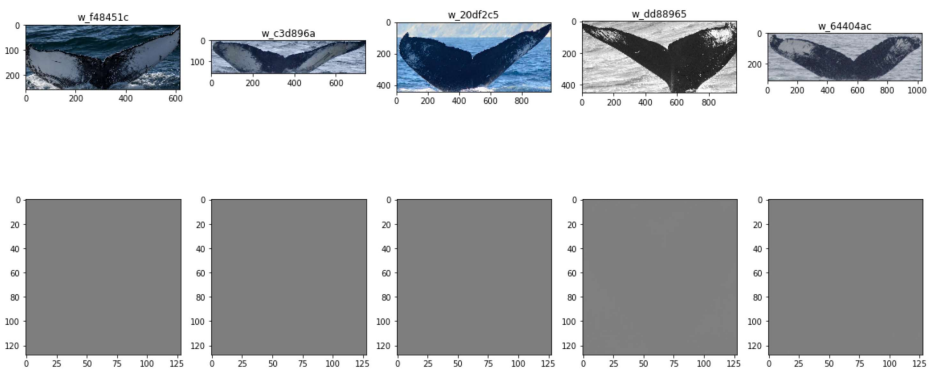
In [15]:
```
from skimage.io import imread
import PIL.Image as pimage

def plot_images(images_names, path):
    fig, m_axs = plt.subplots(1, len(images), figsi
ze = (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_label) 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 learnin
g rate range for your model + dataset.

    # Usage
        ```python
            lr_finder = LRFinder(min_lr=1e-5,
                                 max_lr=1e-2,
                                 steps_per_epoch=np.c
eil(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 th
e dataset. Calculated as `np.ceil(epoch_size/batch_si
ze)`.
        epochs: Number of epochs to run experiment. U
sually between 2 and 4 epochs is sufficient.

    # References
        Blog post: jeremyjordan.me/nn-learning-rate
        Original paper: https://arxiv.org/abs/1506.01
186
    '''

    def __init__(self, min_lr=1e-5, max_lr=1e-1, st
eps_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 * e
pochs
        self.iteration = 0
        self.history = {}
        self.beta = beta
        self.lr_mult = (max_lr/min_lr)**(1/steps_pe
r_epoch)

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

    def on_train_begin(self, logs=None):
        '''Initialize the learning rate to the minim
um 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.m
in_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 upda
te 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.b
eta**self.iteration)
        # Check if the loss is not exploding
        if self.iteration>1 and smoothed_loss > sel
f.best_loss * 1.5:
            self.model.stop_training = True
            return
        if smoothed_loss < self.best_loss or self.i
teration==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', []).a
ppend(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 l
earning rate schedule.'''
#         plt.plot(self.history['iterations'], self.h
istory['lr'])
#         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 le
arning rate 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 insta
nce 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.los
ses[i - sma]) / sma
            derivatives.append(derivative)

        plt.ylabel("d/loss")
        plt.xlabel("learning rate (log scale)")
        plt.plot(self.lrs[n_skip:], derivatives[n_s
kip:])
```

```
        plt.xscale('log')
        if should_lim_y:
            plt.ylim(y_lim)
```

## Train

In [17]:

```
# fits the model on batches with real-time data augme
ntation:
STEP_SIZE_TRAIN=train_generator.n//train_generator.
batch_size

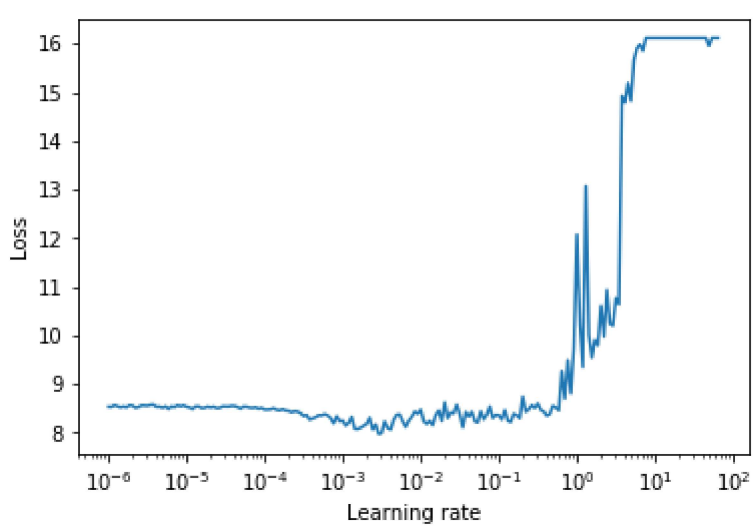
lr_finder = LRFinder(min_lr=1e-6,
                     max_lr=1,
                     steps_per_epoch=np.ceil(train_
generator.n//train_generator.batch_size),
                     epochs=EPOCHS)

# Train the loaded model for more epochs
history = model.fit_generator(generator=train_gener
ator,
                              steps_per_epoch=STEP_
SIZE_TRAIN,
                              epochs=EPOCHS,
                              callbacks=[batch_gc_c
allback, lr_finder])
```

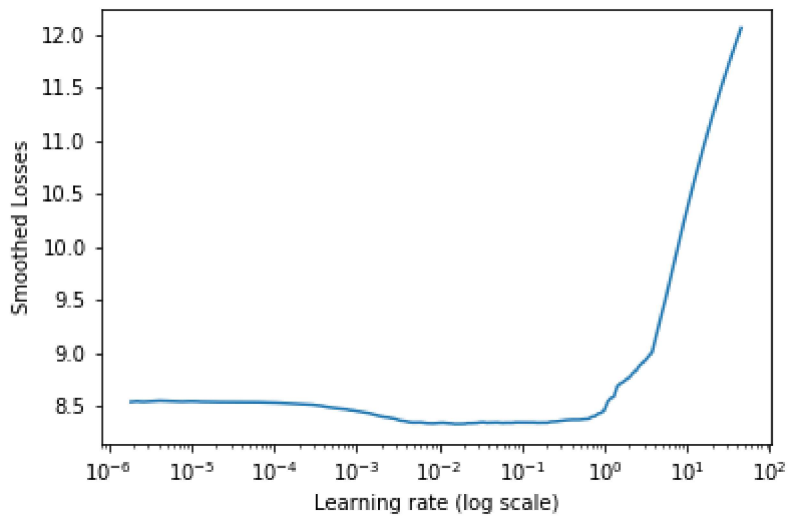
```
Epoch 1/5
 Epoch start learning rate: 1e-06
156/156 [==============================] - 217s 1s/
step - loss: 8.4317 - acc: 0.0040 - mean_absolute_e
rror: 3.9948e-04 - top_5_accuracy: 0.0104
 Epoch ended learning rate: 1.0
Epoch 2/5
 Epoch start learning rate: 1.0
 49/156 [========>.....................] - ETA: 2:1
3 - loss: 14.2739 - acc: 0.0012 - mean_absolute_err
or: 3.9945e-04 - top_5_accuracy: 0.5459 Epoch ended
learning rate: 70.17038
```

In [18]:

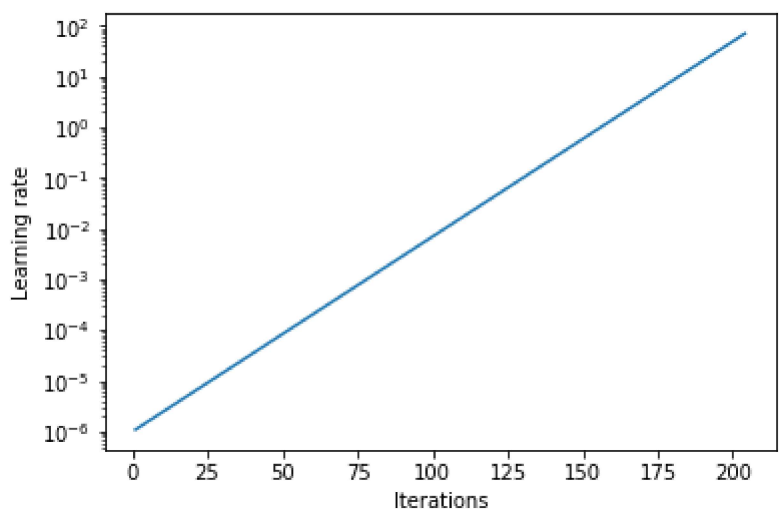
```
lr_finder.plot_loss()
```



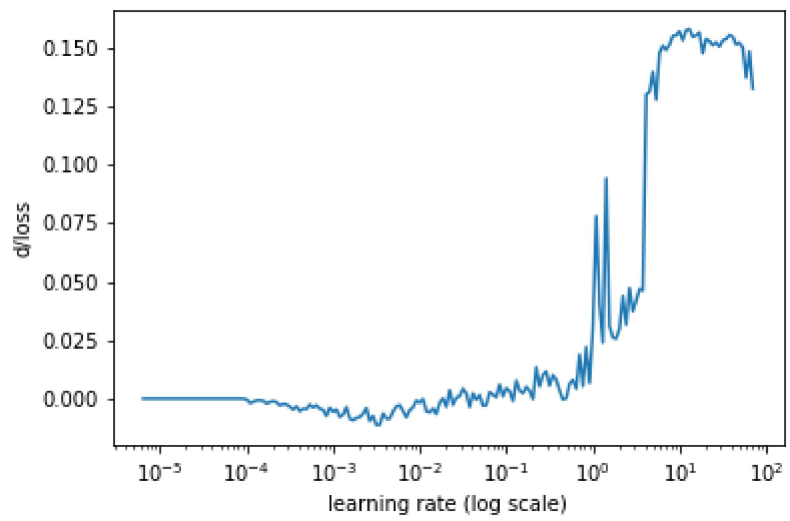
In [19]:
```
lr_finder.plot_smoothed_loss()
```



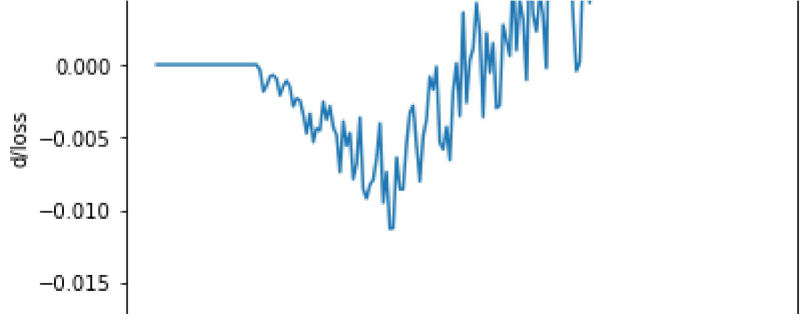

In [20]:
```
lr_finder.plot_lr()
```




In [21]:
```
lr_finder.plot_loss_change(sma=50)
```




In [22]:
```
lr_finder.plot_loss_change(sma=50, y_lim=(-0.02,0.0
1), should_lim_y=True)
```








# Plot train results

In [23]:
```
def plot_accuracy(history, should_plot_val = False,
 should_plot_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_accura
cy')

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



## Data

**Data Sources**

- Humpback Wh...
  - sa 7960 x 2
  - tr 25.4k x 2
  - test.zip
    - 0027089a...
    - 00313e2d...
    - 004344e9...

**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

008a4bc8...
00ac0fca6...
00ff45291...
012dbdb5...

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.

Notebook | Data | Output | Log | Comments

01b1ecf7b....
1000+ more
train.zip
002b4615...
00600ce1...
00d64188...
00eaedfab...
00fee397...
010a1f0eb....
01237f1ce....
01dcb420f...
0202dfb2...
020ab0f9...
1000+ more

- **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_Xception...
Weights_Xcepti...

**About this file**

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

## Model_Xception.h5

We don't support previews for this file yet

## Run Info

| | | | |
|---|---|---|---|
| Succeeded | True | Run Time | 341.3 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 |
|---|---|---|
| 2.9s | 1 | [NbConvertApp] Converting notebook __notebook__.ipynb to notebook |
| 2.9s | 2 | [NbConvertApp] Executing notebook with kernel: python3 |
| 7.9s | 3 | 2019-02-01 13:33:36.358697: 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 |
| 7.9s | 4 | 2019-02-01 13:33:36.359533: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1432] Found device 0 with properties:<br>name: Tesla K80 major: 3 minor: 7<br>memoryClockRate(GHz): 0.8235<br>pciBusID: 0000:00:04.0<br>totalMemory: 11.17GiB freeMemory: 11.10GiB<br>2019-02-01 13:33:36.359619: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1511] Adding visible gpu devices: 0 |
| 8.4s | 5 | 2019-02-01 13:33:36.890580: I tensorflow/core/common_runtime/gpu/gpu_device.cc:982] Device interconnect StreamExecutor with strength 1 edge matrix:<br>2019-02-01 13:33:36.890652: I tensorflow/core/common_runtime/gpu/gpu_device.cc:988] 0<br>2019-02-01 13:33:36.890669: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1001] 0: N |
| 8.4s | 6 | 2019-02-01 13:33:36.894585: 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) |
| 338.0s | 7 | [NbConvertApp] Writing 503091 bytes to __notebook__.ipynb |
| 340.3s | 8 | [NbConvertApp] Converting notebook __notebook__.ipynb to html |
| 340.9s | 9 | [NbConvertApp] Support files will be in __results___files/<br>[NbConvertApp] Making directory __results___files |
| 340.9s | 10 | [NbConvertApp] Making directory __results___files<br>[NbConvertApp] Making directory __results___files<br>[NbConvertApp] Making directory __results___files<br>[NbConvertApp] Making directory __results___files<br>[NbConvertApp] Making directory __results___files<br>[NbConvertApp] Making directory __results___files<br>[NbConvertApp] Making directory __results___files<br>[NbConvertApp] Making directory __results___files<br>[NbConvertApp] Making directory __results___files<br>[NbConvertApp] Writing 356565 bytes to __results__.html |
| 340.9s | 11 | |
| 340.9s | 13 | Complete. Exited with code 0. |

## Comments (0)

Click here to enter a comment...