

Submission**✓ Ran successfully**

Submitted by NinaV 7 days ago

Public Score

0.449

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"))

# Any results you write to the current directory are saved as output.
```

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

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'

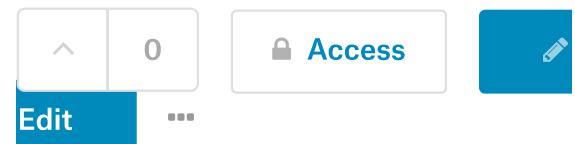
train = pd.read_csv(LABELS)
print("With new_whale:")
train.head()
```



Xception

Python notebook using data from [multiple data sources](#) · 86 views · multiple data sources

Edit tags



Version 1

14 commits

forked from InceptionResNetV2

	Image	Id
0	0000e88ab.jpg	w_f48451c
1	0001f9222.jpg	w_c3d896a
2	00029d126.jpg	w_20df2c5

[Notebook](#)

Data

Output

Log

Comments

3	00050a15a.jpg	new_whale
4	0005c1ef8.jpg	new_whale

In [3]:

```
MODEL_F = 'Model_Xception_flow.h5'
WEIGHTS_F = 'Weights_Xception_flow.h5'
MODEL = '../input/Xception-pretrained/' + MODEL_F
WEIGHTS = '../input/Xception-pretrained/' + WEIGHTS_
F
```

In [4]:

```
train.describe()
```

Out[4]:

	Image	Id
count	25361	25361
unique	25361	5005
top	b7e9a09c5.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
81b501cee.jpg

Out[5]:

[Notebook](#)[Data](#)[Output](#)[Log](#)[Comments](#)

In [6]:

```
train_images = train.set_index('Image')
new_whale_train = train_images[train_images.Id ==
    "new_whale"] # only new_whale dataset
# whales_train = train_images[~(train_images.Id == "n
ew_whale")] # no new_whale dataset, used for trainin
g
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)
```

In [8]:

```
whales_train.describe()
```

Out[8]:

	Image	Id
count	15697	15697
unique	15697	5004
top	b7e9a09c5.jpg	w_23a388d
freq	1	73

In [9]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
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_input
from keras.applications.xception import Xception, preprocess_input

from keras.losses import binary_crossentropy

from keras.layers import Input, Dense, Activation,
BatchNormalization, Flatten, Conv2D, GlobalAveragePooling2D
from keras.layers import AveragePooling2D, MaxPooli
```

```
ng2D, Dropout
from keras.models import Model

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
```

Using TensorFlow backend.

In [10]:

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

def prepareImages(data, m, dataset):
    print("Preparing images")
    X_train = np.zeros((m, IMAGE_HEIGHT, IMAGE_WIDT
H, 3))
    count = 0

    for fig in data['Image']:
        filepath = "../input/"+dataset+"/"+fig
        img = image.load_img(filepath)
        img = img.convert(mode="RGB")

        #load images into images of required size
        img = img.resize((IMAGE_HEIGHT, IMAGE_WIDTH
))
        x = image.img_to_array(img)
        x = preprocess_input(x)

        X_train[count] = x
        if (count%500 == 0):
            print("Processing image: ", count+1, ", "
", fig)

        count += 1

    return X_train
```

In [11]:

```
def remove_new_whale():
    labels_dict = dict()
    labels_list = []

    for i in range(len(unique_labels)):
        labels_dict[unique_labels[i]] = i
        labels_list.append(unique_labels[i])

    print("Number of classes: {}".format(len(unique
_labels)))
```

```
print(np.shape(labels_list))
labels_list = np.array(labels_list)
return labels_list, labels_dict
```

In [12]:

```
labels_list, labels_dict = remove_new_whale()
```

Number of classes: 5004
(5004,)

In [13]:

```
whales_train.Id = whales_train.Id.apply(lambda x: 1
                                         labels_dict[x])
```

```
/opt/conda/lib/python3.6/site-packages/pandas/core/
generic.py:4405: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice fr
om a DataFrame.
Try using .loc[row_indexer,col_indexer] = value ins
tead
```

See the caveats in the documentation: <http://panda
s.pydata.org/pandas-docs/stable/indexing.html#index
ing-view-versus-copy>

```
self[name] = value
```

In [14]:

```
print(whales_train.head())
```

	Image	Id
0	0000e88ab.jpg	4785
1	0001f9222.jpg	3807
2	00029d126.jpg	661
6	000a6daec.jpg	4314
8	0016b897a.jpg	1928

In [15]:

```
def prepare_labels(y):
    values = np.array(y)
    label_encoder = LabelEncoder()
    integer_encoded = label_encoder.fit_transform(v
values)
    # print(integer_encoded)

    onehot_encoder = OneHotEncoder(sparse=False)
    integer_encoded = integer_encoded.reshape(len(i
nteger_encoded), 1)
    #     print(integer_encoded)
    onehot_encoded = onehot_encoder.fit_transform(i
nteger_encoded)
    #     print(onehot_encoded)

    y = onehot_encoded
    print(y.shape)
    return y, label_encoder
```

```
In [16]:  
y, label_encoder = prepare_labels(whales_train['Id'])  
y.shape
```

```
(15697, 5004)
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/preprocessing/_encoders.py:368: FutureWarning: The handling of integer data will change in version 0.22. Currently, the categories are determined based on the range [0, max(values)], while in the future they will be determined based on the unique values.  
If you want the future behaviour and silence this warning, you can specify "categories='auto'".  
In case you used a LabelEncoder before this OneHotEncoder to convert the categories to integers, then you can now use the OneHotEncoder directly.  
warnings.warn(msg, FutureWarning)
```

```
Out[16]:  
(15697, 5004)
```

```
In [17]:  
%matplotlib inline  
X = prepareImages(whales_train, whales_train.shape[0], TRAIN_CROPPED)  
X /= 255
```

```
Preparing images  
Processing image: 1 , 0000e88ab.jpg  
Processing image: 501 , 0823f9df3.jpg  
Processing image: 1001 , 10b694367.jpg  
Processing image: 1501 , 195805c52.jpg  
Processing image: 2001 , 21e28ae02.jpg  
Processing image: 2501 , 2a1146baa.jpg  
Processing image: 3001 , 32533a7fb.jpg  
Processing image: 3501 , 3a8173905.jpg  
Processing image: 4001 , 42f134dea.jpg  
Processing image: 4501 , 4aa4de13a.jpg  
Processing image: 5001 , 5297b6c40.jpg  
Processing image: 5501 , 5b7f0e6e6.jpg  
Processing image: 6001 , 6311688b7.jpg  
Processing image: 6501 , 6b29760e3.jpg  
Processing image: 7001 , 7390cbfab.jpg  
Processing image: 7501 , 7b949f512.jpg  
Processing image: 8001 , 83336c385.jpg  
Processing image: 8501 , 8b369569b.jpg  
Processing image: 9001 , 92f450203.jpg  
Processing image: 9501 , 9b984102a.jpg  
Processing image: 10001 , a39bab55.jpg  
Processing image: 10501 , ab6f8bddd.jpg  
Processing image: 11001 , b36da6f7c.jpg  
Processing image: 11501 , bb9ffa8b2.jpg  
Processing image: 12001 , c4160ee65.jpg  
Processing image: 12501 , cb7153d51.jpg  
Processing image: 13001 , d3b15e280.jpg  
Processing image: 13501 , dbb2088f4.jpg
```

```
Processing image: 14001 , e3fe27a84.jpg
Processing image: 14501 , ebde74948.jpg
Processing image: 15001 , f3f3f8b92.jpg
Processing image: 15501 , fc54db327.jpg
```

In [18]:

```
CLASSES = 5004
EPOCHS = 30
BATCH_SIZE = 100

# setup model
base_model = Xception(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'])

model.summary()
```

```
Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.4/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83689472/83683744 [=====]
- 1s 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]
```

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 (Activation (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 (BatchNorm (None, 8, 8, 728)
2912 block12_sepconv1[0][0]

block12_sepconv2_act (Activation (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 (BatchNorm (None, 8, 8, 728)
2912 block12_sepconv2[0][0]

block12_sepconv3_act (Activation (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 (BatchNorm (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 (Activation (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 (BatchNorm (None, 8, 8, 728)

2912 block13_sepconv1[0][0]

block13_sepconv2_act (Activation (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 (BatchNorm (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 (BatchNorm (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 (BatchNorm (None, 4, 4, 1536)
6144 block14_sepconv1[0][0]

block14_sepconv1_act (Activation (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 (BatchNorm (None, 4, 4, 2048)
8192 block14_sepconv2[0][0]

block14_sepconv2_act (Activation (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 [19]:

```
print("Train set shape: "+ str(np.shape(X)))  
print(np.shape(whales_train['Id']))  
history = model.fit(X, y, epochs=EPOCHS, batch_size  
=BATCH_SIZE, verbose=1)
```

```
Train set shape: (15697, 128, 128, 3)  
(15697,)  
Epoch 1/30  
15697/15697 [=====] - 206s  
13ms/step - loss: 8.0717 - acc: 0.0120  
Epoch 2/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 7.0071 - acc: 0.0482  
Epoch 3/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 5.9815 - acc: 0.1165  
Epoch 4/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 4.9976 - acc: 0.2006  
Epoch 5/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 4.0928 - acc: 0.2997  
Epoch 6/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 3.2467 - acc: 0.4017  
Epoch 7/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 2.4744 - acc: 0.5092  
Epoch 8/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 1.8233 - acc: 0.6135  
Epoch 9/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 1.2500 - acc: 0.7208  
Epoch 10/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 0.8375 - acc: 0.8016  
Epoch 11/30  
15697/15697 [=====] - 188s  
12ms/step - loss: 0.5660 - acc: 0.8604  
Epoch 12/30  
15697/15697 [=====] - 188s
```

```
12ms/step - loss: 0.3838 - acc: 0.9035
Epoch 13/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.2719 - acc: 0.9309
Epoch 14/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.2026 - acc: 0.9481
Epoch 15/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.1671 - acc: 0.9573
Epoch 16/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.1390 - acc: 0.9648
Epoch 17/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.1100 - acc: 0.9729
Epoch 18/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.0945 - acc: 0.9740
Epoch 19/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.0841 - acc: 0.9771
Epoch 20/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.0888 - acc: 0.9770
Epoch 21/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.1083 - acc: 0.9694
Epoch 22/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.1048 - acc: 0.9707
Epoch 23/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.1004 - acc: 0.9714
Epoch 24/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.1011 - acc: 0.9713
Epoch 25/30
15697/15697 [=====] - 187s
12ms/step - loss: 0.0847 - acc: 0.9756
Epoch 26/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.0877 - acc: 0.9758
Epoch 27/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.0772 - acc: 0.9779
Epoch 28/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.0860 - acc: 0.9755
Epoch 29/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.0782 - acc: 0.9782
Epoch 30/30
15697/15697 [=====] - 188s
12ms/step - loss: 0.0644 - acc: 0.9809
```

In [20]:

```
# validate
val_set_x = X[1000:6000]
val_set_y = y[1000:6000]
```

```
val_set_y = y[1000:6000]
scores = model.evaluate(val_set_x, val_set_y, verbose=1)
print("%s: %.2f%" % (model.metrics_names[1], scores[1]*100))
```

```
5000/5000 [=====] - 21s 4m
s/step
acc: 95.84%
```

In [21]:

```
model.save(MODEL_F)
print("Saved model architecture to disk")
model.save_weights(WEIGHTS_F)
print("Saved model weights to disk")
```

```
Saved model architecture to disk
Saved model weights to disk
```

In [22]:

```
gc.collect()
```

Out[22]:

```
22
```

In [23]:

```
# 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")

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

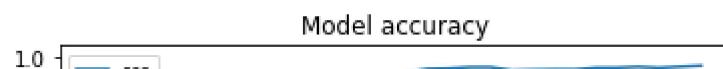
# gc.collect()
```

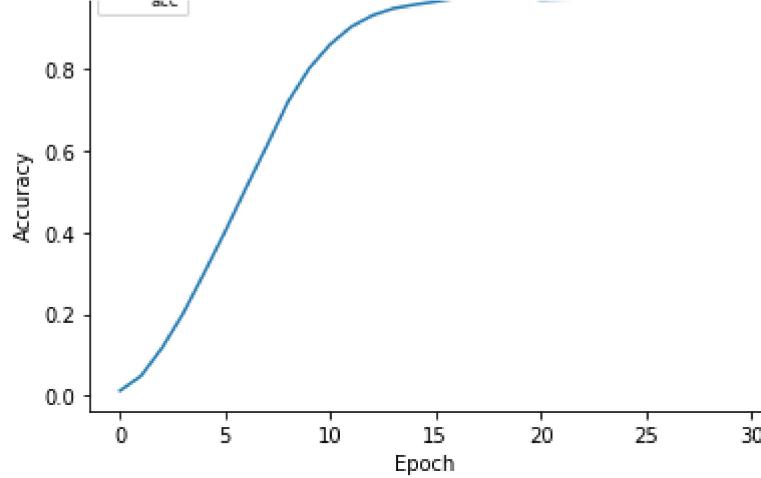
Plot train results

In [24]:

```
acc = history.history['acc']
# val_acc = history.history['val_acc']

l1 = plt.plot(acc, label='acc')
# l2 = plt.plot(val_acc, label='val_acc')
plt.legend(loc=2, fontsize="small")
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.show()
```

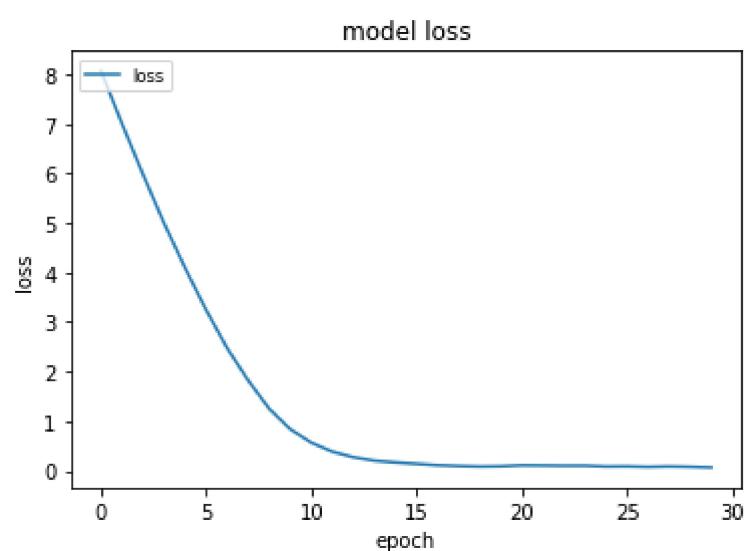




In [25]:

```
loss = history.history['loss']
# val_loss = history.history['val_loss']

l1 = plt.plot(loss, label='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 [26]:

```
test = os.listdir(TEST_CROPPED_IN)
print("Test set length: "+str(len(test)))
```

Test set length: 7960

In [27]:

```
col = ['Image']
test_df = pd.DataFrame(test, columns=col)
test_df['Id'] = ''
```

In [28]:

```
X = prepareImages(test_df, test_df.shape[0], TEST_CROPPED)
# X /= 255
```

Preparing images

Processing image: 1 , f28e2a7e7.jpg

Processing image: 501 , fd3f864b8.jpg

```
Processing image: 1001 , 7861b6fce.jpg
Processing image: 1501 , 697c3eb70.jpg
Processing image: 2001 , 2363e6db8.jpg
Processing image: 2501 , 85fa8ce46.jpg
Processing image: 3001 , 37954bd1d.jpg
Processing image: 3501 , 49ef1402e.jpg
Processing image: 4001 , 9d6218450.jpg
Processing image: 4501 , 504633756.jpg
Processing image: 5001 , 3d145ac9c.jpg
Processing image: 5501 , 2a52b5d42.jpg
Processing image: 6001 , c4b81b9a7.jpg
Processing image: 6501 , 41086e3b8.jpg
Processing image: 7001 , cf038a370.jpg
Processing image: 7501 , 28951b661.jpg
```

Test set prediction using generator and flow_from_dataframe

```
In [29]:  
test_datagen = ImageDataGenerator(  
    preprocessing_function=preprocess_input,  
    rescale=1./255,  
    fill_mode='nearest')  
  
test_generator = test_datagen.flow_from_dataframe(  
    dataframe=test_df,  
    directory=TEST_CROPPED_IN,  
    x_col="Image",  
    y_col=None,  
    class_mode=None,  
    shuffle=False,  
    color_mode= "rgb",  
    target_size=(IMAGE_HEIGHT, IMAGE_WIDTH),  
    batch_size=1)  
  
#we need to use .reset() here otherwise  
#the other of predictions will be different  
#then the expected  
test_generator.reset()  
predictions = model.predict_generator(test_generator,  
verbose = 1,steps=7960)  
  
print("Predictions shape:")  
print(np.shape(predictions))
```

```
Found 7960 images.  
7960/7960 [=====] - 157s 2  
0ms/step  
Predictions shape:  
(7960, 5004)
```

Test set predictions

```
In [30]:  
    # predictions = model.predict(np.array(X), verbose=1)  
    # print(np.shape(predictions))
```

```
In [31]:  
predicted_class_indices=np.argmax(predictions, axis=1)  
  
np.save("predictions.npy", predictions)  
np.save("predicted_class_indices.npy", predicted_cl  
ass_indices)  
np.save('test_filenames_generator.npy', test genera  
tor.filenames)  
np.save('test_class_indices.npy', test_generator.cl  
ass_indices)  
  
print('predicted class indices:')  
print(predicted_class_indices)
```

```
predicted class indices:  
[4334 1276 3036 ... 3670 1912 3048]
```

```
In [32]:  
print(labels_list[:7])  
labels_with_new_whale = np.concatenate(([ 'new_whal  
e'], labels_list), axis=0)  
print(labels_with_new_whale[:7])
```

```
[ 'w_0003639' 'w_0003c59' 'w_0027efa' 'w_00289b1' 'w  
_002c810' 'w_0032a46'  
'w_003bae6']  
[ 'w_0003639' 'w_0003c59' 'w_0027efa' 'w_00289b1' 'w  
_002c810' 'w_0032a46'  
'w_003bae6']
```

This kernel has been released under the [Apache 2.0](#) open source license.

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

0

Data

Data Sources

- ▼ 🏆 Humpback Wh...
 - ▀ s: 7960 x 2
 - ▀ t: 25.4k x 2
- ▼ 📁 test.zip
 - ▀ 0027089a...
 - ▀ 00313e2d...
 - ▀ 004344e9...
 - ▀ 008a4bc8...

 **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

■	00ac0fc...
■	00ff45291...
■	012dbdb5...
■	0169cec0...
■	01830c9cf...
■	01b1ecf7b....
...	1000+ more
▼	train.zip
■	002b4615...
■	00600ce1...
■	00d64188...
■	00eaedfab...
■	00fee397...
■	010a1f0eb....
■	01237f1ce....
■	01dcbb420f...
■	0202dfb2...
■	020ab0f9...
...	1000+ more

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

- [submission.csv](#)
- [Model_Xception...](#)
- [predicted_class_...](#)
- [predictions.npy](#)
- [test_class_indic...](#)
- [test_filenames_...](#)
- [Weights_Xcepti...](#)

About this file

[Submit to Competition](#)

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

■ [submission.csv](#)



1	Image	Id
2	00028a005.jpg	w_dec7ffd new_whale w_e918d4c w_22d96e7 w_11d8c70
3	000dcf7d8.jpg	w_4132bb8 new_whale w_950c471 w_d72771c w_e19c1d6
4	000e7c7df.jpg	new_whale w_9ba4a9a w_dcde3f2 w_687c6d3 w_7c1435e
5	0019c34f4.jpg	new_whale w_fa2ec89 w_59052ad w_e27bbf6 w_778e474

6	001a4d292.jpg	new_whale w_cf5d69c w_502e72f w_5b40032 w_be1dbc1	
7	00247bc36.jpg	new_whale w_8e1e84c w_025911c w_be65810 w_0a31a3a	
8	0027089a4.jpg	w_c6c89db new_whale w_2745292 w_7e2eb3d w_472b51c	
9	002de4d94.jpg	w_b3c3fa5 new_whale w_6091feb w_be1dbc1 w_038dfc0	
10	002f52f0c	new_whale	

Run Info

Succeeded	True	Run Time	6045.1 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
2.8s    1  [NbConvertApp] Converting notebook script.ipynb to
           html
2.9s    2  [NbConvertApp] Executing notebook with kernel:
           python3
106.3s   3  2019-01-25 22:36:56.555652: 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
106.3s   4  2019-01-25 22:36:56.556465: 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
2019-01-25 22:36:56.556505: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:1511]
           Adding visible gpu devices: 0
106.6s   5  2019-01-25 22:36:56.910219: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:982]
           Device interconnect StreamExecutor with strength 1
           edge matrix:
2019-01-25 22:36:56.910295: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:988]
           0
106.6s   6  2019-01-25 22:36:56.910310: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:1001]
           0: N
2019-01-25 22:36:56.910823: 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)
6044.5s   7  [NbConvertApp] Support files will be in

```

```
--results__files/  
[NbConvertApp] Making directory --results__files  
[NbConvertApp] Making directory --results__files  
[NbConvertApp] Making directory --results__files  
6044.5s     8 [NbConvertApp] Writing 351478 bytes to  
--results__.html  
6044.5s     9  
6044.5s    11 Complete. Exited with code 0.
```

Comments (0)



Click here to enter a comment...

