

Submission**✓ Ran successfully**

Submitted by NinaV 20 hours ago

Public Score

0.452

```
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"))

# 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_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_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	c06297e1b.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
564d9aee9.jpg

Out[5]:



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 == "new_whale")]
# no new_whale dataset, used for training
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	153ab610b.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, MaxPooling2D, 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=DeprecationWarning)

%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_WIDTH, 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: 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 from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-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(values)
    # print(integer_encoded)

    onehot_encoder = OneHotEncoder(sparse=False)
```

```
    integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
    #     print(integer_encoded)
    onehot_encoded = onehot_encoder.fit_transform(integer_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]:

```
import keras.backend as K
from keras.callbacks import Callback

class GcCollectors(Callback):
    def __init__(self):
        super().__init__()

    def on_epoch_end(self, ep, logs=None):
        gc.collect()

    def on_epoch_begin(self, ep, logs=None):
        gc.collect()

    # def on_batch_end(self, batch, logs=None):
    #     gc.collect()

    # def on_batch_begin(self, batch, logs=None):
    #     gc.collect()

    def on_train_begin(self, logs=None):
        gc.collect()

    def on_train_end(self, logs=None):
        gc.collect()
```

In [19]:

```
from keras.optimizers import Adam

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

op = Adam(lr=0.003)

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 [=====] - 6s 0us/s
tep

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) 526526
```

```
block4_sepconvz (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]

add_11 (Add) (None, 8, 8, 728) 0
block12_sepconv2_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]

add_12 (Add) (None, 8, 8, 728) 0
block13_sepconv2_bn[0][0]

add_11[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 (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 [20]:

```
print("Train set shape: "+ str(np.shape(X)))
print(np.shape(whales_train['Id']))

gc.collect()
gc_collectors = GcCollectors()

history = model.fit(X, y, epochs=EPOCHS, batch_size=BATCH_SIZE,
verbose=1, callbacks = [gc_collectors])
```

```
Train set shape: (15697, 128, 128, 3)
(15697, )
Epoch 1/30
15697/15697 [=====] - 207s 13ms/step
- loss: 8.2963 - acc: 0.0076
Epoch 2/30
15697/15697 [=====] - 189s 12ms/step
- loss: 7.3789 - acc: 0.0270
```

```
Epoch 3/30
15697/15697 [=====] - 190s 12ms/step
- loss: 6.5583 - acc: 0.0579
Epoch 4/30
15697/15697 [=====] - 190s 12ms/step
- loss: 5.8202 - acc: 0.1061
Epoch 5/30
15697/15697 [=====] - 190s 12ms/step
- loss: 5.1195 - acc: 0.1643
Epoch 6/30
15697/15697 [=====] - 190s 12ms/step
- loss: 4.4224 - acc: 0.2360
Epoch 7/30
15697/15697 [=====] - 190s 12ms/step
- loss: 3.8185 - acc: 0.3057
Epoch 8/30
15697/15697 [=====] - 190s 12ms/step
- loss: 3.1726 - acc: 0.3967
Epoch 9/30
15697/15697 [=====] - 190s 12ms/step
- loss: 2.5557 - acc: 0.4798
Epoch 10/30
15697/15697 [=====] - 190s 12ms/step
- loss: 2.0208 - acc: 0.5703
Epoch 11/30
15697/15697 [=====] - 190s 12ms/step
```

Xception - lr optimization

Python notebook using data from [multiple data sources](#) · 20 views · multiple data sources Edit tags



```
- loss: 1.1242 - acc: 0.7323
Epoch 13/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.8228 - acc: 0.7985
Epoch 14/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.6051 - acc: 0.8447
Epoch 15/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.4475 - acc: 0.8839
Epoch 16/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.3187 - acc: 0.9162
Epoch 17/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.2310 - acc: 0.9401
Epoch 18/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.1918 - acc: 0.9498
Epoch 19/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.1564 - acc: 0.9587
Epoch 20/30
15697/15697 [=====] - 189s 12ms/step
- loss: 0.1352 - acc: 0.9639
Epoch 21/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.1250 - acc: 0.9676
Epoch 22/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.1142 - acc: 0.9694
Epoch 23/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.0983 - acc: 0.9747
Epoch 24/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.1086 - acc: 0.9698
Epoch 25/30
```

Version 1

3 commits

forked from Xception

Notebook

Data

Output

Log

Comments

```
Epoch 25/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.1019 - acc: 0.9717
Epoch 26/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.1031 - acc: 0.9704
Epoch 27/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.1002 - acc: 0.9717
Epoch 28/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.0837 - acc: 0.9765
Epoch 29/30
15697/15697 [=====] - 190s 12ms/step
- loss: 0.0979 - acc: 0.9721
Epoch 30/30
15697/15697 [=====] - 189s 12ms/step
- loss: 0.1012 - acc: 0.9711
```

In [21]:

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

 Notebook  Data  Output  Log  Comments

```
5000/5000 [=====] - 21s 4ms/step
acc: 97.58%
```

In [22]:

```
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 [23]:

```
gc.collect()
```

Out[23]:

```
22
```

In [24]:

```
# 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")

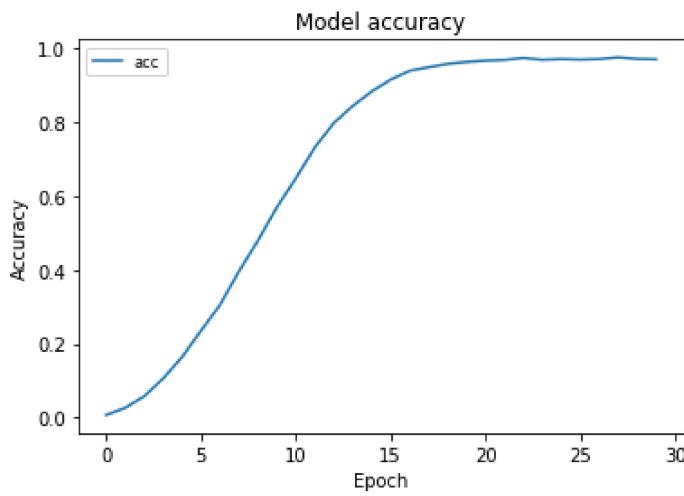
# model.load_weights(WEIGHTS)
# print("Loaded model weights from disk")
# model.summary()

# gc.collect()
```

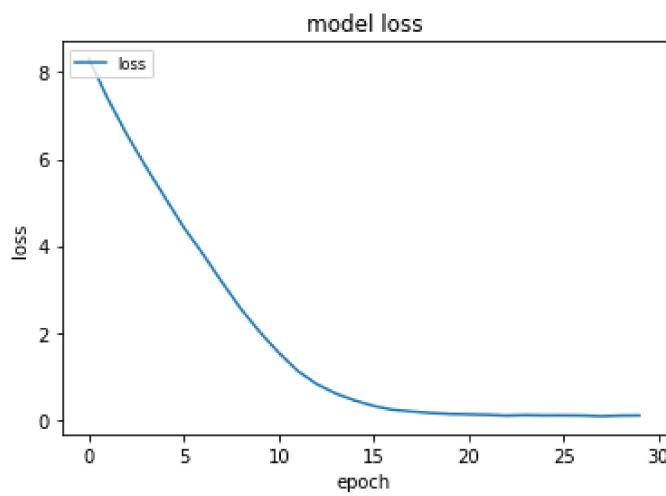
Plot train results

In [25]:

```
In [25]:  
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 [26]:  
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 [27]:  
test = os.listdir(TEST_CROPPED_IN)  
print("Test set length: "+str(len(test)))
```

Test set length: 7960

```
In [28]:  
col = ['Image']  
test_df = pd.DataFrame(test, columns=col)  
test_df['Id'] = ''
```

```
In [29]:  
X = prepareImages(test_df, test_df.shape[0], TEST_CROPPED)
```

```
# X /= 255
```

```
Preparing images
Processing image: 1 , aabc5cf3b.jpg
Processing image: 501 , 317181820.jpg
Processing image: 1001 , c1fec8ca9.jpg
Processing image: 1501 , be22d0a88.jpg
Processing image: 2001 , 45627d1e5.jpg
Processing image: 2501 , 5fb33057d.jpg
Processing image: 3001 , 0c0277706.jpg
Processing image: 3501 , 8d8c7a728.jpg
Processing image: 4001 , 7541a5a86.jpg
Processing image: 4501 , c89ab7891.jpg
Processing image: 5001 , dbf3b5552.jpg
Processing image: 5501 , 7d3e4531c.jpg
Processing image: 6001 , e66ddb0b0.jpg
Processing image: 6501 , 1d26cd932.jpg
Processing image: 7001 , 9f035e029.jpg
Processing image: 7501 , 763d8fdef.jpg
```

Test set prediction using generator and flow_from_dataframe

In [30]:

```
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 [=====] - 144s 18ms/step
Predictions shape:
(7960, 5004)
```

Test set predictions

In [31]:

```
# predictions = model.predict(np.array(X), verbose=1)
# print(np.shape(predictions))
```

In [32]:

```
predicted_class_indices=np.argmax(predictions, axis=1)

np.save("predictions.npy", predictions)
np.save("predicted_class_indices.npy", predicted_class_indices)
np.save('test_filenames_generator.npy', test_generator.filenames)
np.save('test_class_indices.npy', test_generator.class_indices)

print('predicted class indices:')
print(predicted_class_indices)
```

predicted class indices:

```
[4334 1276 3036 ... 824 1370 4675]
```

In [33]:

```
print(labels_list[:7])
labels_with_new_whale = np.concatenate(([['new_whale']], labels_list), axis=0)
print(labels_with_new_whale[:7])
```

```
['w_0003639' 'w_0003c59' 'w_0027efa' 'w_00289b1' 'w_002c810'
 'w_0032a46'
 'w_003bae6']
[['new_whale' 'w_0003639' 'w_0003c59' 'w_0027efa' 'w_00289b1'
 'w_002c810'
 'w_0032a46']]
```

In [34]:

```
def add_new_whale_to_predictions(preds):
    sorted_preds = np.sort(preds)
    avg_of_max_predictions = np.average(sorted_preds[:, -1:])
    print("Average of max probabilities column:" + str(avg_of_max_predictions))
    best_threshold = avg_of_max_predictions
    # print(np.shape(preds))
    shape_to_add = (np.shape(preds)[0], 1)
```

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 Whale I...
 - ▀ sa... 7960 x 2
 - ▀ trai... 25.4k x 2
- ▼ 📄 test.zip
 - ▀ 0027089a4.jpg



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

00313e2d2.jpg
004344e9f.jpg
008a4bc86.jpg
00ac0fcfa6.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
01dc420f.jpg
0202dfb29.jpg
020ab0f9b.jpg
... 1000+ more

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	About this file	Submit to Competition
<ul style="list-style-type: none"> submission.csv Model_Xception_flo... predicted_class_indi... predictions.npy test_class_indices.npy test_filenames_gene... Weights_Xception_fl... 	<p>This file was created from a Kernel, it does not have a description.</p>	

submission.csv



1	Image	Id
2	00028a005.jpg	w_dec7ffd new_whale w_df6b050 w_bb2c919 w_840061a
3	000dcf7d8.jpg	w_4132bb8 new_whale w_e3956f5 w_d72771c w_3155d04
4	000e7c7df.jpg	new_whale w_9ba4a9a w_c6c60c8 w_d76f21b w_7983f71
5	0019c34f4.jpg	new_whale w_a63cd8d w_179c9f0 w_e21b629 w_59052ad
6	001a4d292.jpg	new_whale w_502e72f w_da40b39 w_da08395 w_6e7cf1c

7	00247bc36.jpg	w_dc651ad new_whale w_df27de9 w_b397217 w_025911c	
8	0027089a4.jpg	new_whale w_c6c89db w_7e2eb3d w_5b227ec w_939a97b	
9	002de4d94.jpg	w_dd80742 new_whale w_fa52b0c w_896b139 w_17e8846	
10	002f52f0c.jpg	w_691b684 new_whale w_73b6699 w_025911c	

Run Info

Succeeded	True	Run Time	6083.7 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
5.4s    1 [NbConvertApp] Converting notebook __notebook__.ipynb to
          notebook
5.5s    2 [NbConvertApp] Executing notebook with kernel: python3
100.4s   3 2019-02-02 22:23:23.537653: 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
100.4s   4 2019-02-02 22:23:23.538141: 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 22:23:23.538191: I
          tensorflow/core/common_runtime/gpu/gpu_device.cc:1511] Adding
          visible gpu devices: 0
100.7s   5 2019-02-02 22:23:23.873002: I
          tensorflow/core/common_runtime/gpu/gpu_device.cc:982] Device
          interconnect StreamExecutor with strength 1 edge matrix:
          2019-02-02 22:23:23.873064: I
          tensorflow/core/common_runtime/gpu/gpu_device.cc:988]      0
          2019-02-02 22:23:23.873081: I
          tensorflow/core/common_runtime/gpu/gpu_device.cc:1001] 0:  N
100.8s   6 2019-02-02 22:23:23.873693: 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)
6079.2s   7 [NbConvertApp] Writing 126558 bytes to __notebook__.ipynb
6082.5s   8 [NbConvertApp] Converting notebook __notebook__.ipynb to html
6083.2s   9 [NbConvertApp] Support files will be in __results__files/
6083.2s  10 [NbConvertApp] Making directory __results__files
          [NbConvertApp] Making directory __results__files
          [NbConvertApp] Making directory __results__files
          [NbConvertApp] Writing 355026 bytes to __results__.html
6083.2s  11
6083.2s  13 Complete. Exited with code 0.

```

Comments (0)



Click here to enter a comment...

