

Submission**✓ Ran successfully**

Submitted by NinaV 10 days ago

Public Score

0.426

In [1]:

```
# This Python 3 environment comes with many helpful ana  
lytics libraries installed  
# It is defined by the kaggle/python docker image: http  
s://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.
```

```
[ 'resnet50-pretrained', '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()
```

With new_whale:

Out[2]:

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

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

In [3]:

```
MODEL_F = 'Model_ResNet50_30epochs.h5'
WEIGHTS_F = 'Weights_ResNet50_30epochs.h5'
MODEL = '../input/resnet50-pretrained/' + MODEL_F
WEIGHTS = '../input/resnet50-pretrained/' + WEIGHTS_F
```

In [4]:

```
train.describe()
```

Out[4]:

	Image	Id
count	25361	25361
unique	25361	5005
top	b3961a1c9.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
bfefec875.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	a5ad2b36d.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.layers import Input, Dense, Activation, BatchNormalization, Flatten, Conv2D
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
```

```
In [10]:
```

```

IMAGE_HEIGHT = 100
IMAGE_WIDTH = 100

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):
#            plt.imshow(img)
            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/ge  
neric.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 inste  
ad
```

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(val  
ues)  
    # print(integer_encoded)  
  
    onehot_encoder = OneHotEncoder(sparse=False)  
    integer_encoded = integer_encoded.reshape(len(int  
eger_encoded), 1)  
    # print(integer_encoded)  
    onehot_encoded = onehot_encoder.fit_transform(int  
eger_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/prepro  
cessing/_encoders.py:368: FutureWarning: The handling  
of integer data will change in version 0.22. Currentl  
y, the categories are determined based on the range  
[0, max(values)], while in the future they will be de  
termined based on the unique values.
```

If you want the future behaviour and silence this war
ning, 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]:  
from keras.models import Model  
from keras.layers import Dense, GlobalAveragePooling2D, Dropout  
from keras.applications.resnet50 import ResNet50, preprocess_input  
from keras.losses import binary_crossentropy  
  
CLASSES = 5004  
  
# setup model
```

```
# Setup model
# base_model = ResNet50(weights='imagenet', include_top=False, input_shape = (100, 100, 3))

# 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()
```

In [19]:

```
# print("Train set shape: " + str(np.shape(X)))
# print(np.shape(whales_train['Id']))
# history = model.fit(X, y, epochs=30, batch_size=100,
# verbose=1)
```

In [20]:

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

In [21]:

```
gc.collect()
```

Out[21]:

```
4
```

In [22]:

```
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()
```

Loaded model architecture from disk

Loaded model weights from disk

Layer (type)	Output Shape
--------------	--------------

Param #	Connected to
	=====
	=====
input_1 (InputLayer)	(None, 100, 100, 3)
0	

conv1_pad (ZeroPadding2D)	(None, 106, 106, 3)
0	input_1[0][0]

conv1 (Conv2D)	(None, 50, 50, 64)
9472	conv1_pad[0][0]

bn_conv1 (BatchNormalization)	(None, 50, 50, 64)
256	conv1[0][0]

activation_1 (Activation)	(None, 50, 50, 64)
0	bn_conv1[0][0]

pool1_pad (ZeroPadding2D)	(None, 52, 52, 64)
0	activation_1[0][0]

max_pooling2d_1 (MaxPooling2D)	(None, 25, 25, 64)
0	pool1_pad[0][0]

res2a_branch2a (Conv2D)	(None, 25, 25, 64)
4160	max_pooling2d_1[0][0]

bn2a_branch2a (BatchNormalizati	(None, 25, 25, 64)
256	res2a_branch2a[0][0]

activation_2 (Activation)	(None, 25, 25, 64)
0	bn2a_branch2a[0][0]

res2a_branch2b (Conv2D)	(None, 25, 25, 64)
36928	activation_2[0][0]

bn2a_branch2b (BatchNormalizati	(None, 25, 25, 64)
256	res2a_branch2b[0][0]

activation_3 (Activation)	(None, 25, 25, 64)
0	bn2a_branch2b[0][0]

res2a_branch2c (Conv2D)	(None, 25, 25, 256)
16640	activation_3[0][0]

res2a_branch1 (Conv2D)	(None, 25, 25, 256)

16640 max_pooling2d_1[0][0]

bn2a_branch2c (BatchNormalizati (None, 25, 25, 256)
1024 res2a_branch2c[0][0]

bn2a_branch1 (BatchNormalizatio (None, 25, 25, 256)
1024 res2a_branch1[0][0]

add_1 (Add) (None, 25, 25, 256)
0 bn2a_branch2c[0][0]

bn2a_branch1[0][0]

activation_4 (Activation) (None, 25, 25, 256)
0 add_1[0][0]

res2b_branch2a (Conv2D) (None, 25, 25, 64)
16448 activation_4[0][0]

bn2b_branch2a (BatchNormalizati (None, 25, 25, 64)
256 res2b_branch2a[0][0]

activation_5 (Activation) (None, 25, 25, 64)
0 bn2b_branch2a[0][0]

res2b_branch2b (Conv2D) (None, 25, 25, 64)
36928 activation_5[0][0]

bn2b_branch2b (BatchNormalizati (None, 25, 25, 64)
256 res2b_branch2b[0][0]

activation_6 (Activation) (None, 25, 25, 64)
0 bn2b_branch2b[0][0]

res2b_branch2c (Conv2D) (None, 25, 25, 256)
16640 activation_6[0][0]

bn2b_branch2c (BatchNormalizati (None, 25, 25, 256)
1024 res2b_branch2c[0][0]

add_2 (Add) (None, 25, 25, 256)
0 bn2b_branch2c[0][0]

activation_4[0][0]

activation_7 (Activation) (None, 25, 25, 256)

0 add_2[0][0]

res2c_branch2a (Conv2D) (None, 25, 25, 64)
16448 activation_7[0][0]

bn2c_branch2a (BatchNormalizati (None, 25, 25, 64)
256 res2c_branch2a[0][0]

activation_8 (Activation) (None, 25, 25, 64)
0 bn2c_branch2a[0][0]

res2c_branch2b (Conv2D) (None, 25, 25, 64)
36928 activation_8[0][0]

bn2c_branch2b (BatchNormalizati (None, 25, 25, 64)
256 res2c_branch2b[0][0]

activation_9 (Activation) (None, 25, 25, 64)
0 bn2c_branch2b[0][0]

res2c_branch2c (Conv2D) (None, 25, 25, 256)
16640 activation_9[0][0]

bn2c_branch2c (BatchNormalizati (None, 25, 25, 256)
1024 res2c_branch2c[0][0]

add_3 (Add) (None, 25, 25, 256)
0 bn2c_branch2c[0][0]

activation_7[0][0]

activation_10 (Activation) (None, 25, 25, 256)
0 add_3[0][0]

res3a_branch2a (Conv2D) (None, 13, 13, 128)
32896 activation_10[0][0]

bn3a_branch2a (BatchNormalizati (None, 13, 13, 128)
512 res3a_branch2a[0][0]

activation_11 (Activation) (None, 13, 13, 128)
0 bn3a_branch2a[0][0]

res3a_branch2b (Conv2D) (None, 13, 13, 128)
147584 activation_11[0][0]

bn3a_branch2b (BatchNormalizati (None, 13, 13, 128)
512 res3a_branch2b[0][0]

activation_12 (Activation) (None, 13, 13, 128)
0 bn3a_branch2b[0][0]

res3a_branch2c (Conv2D) (None, 13, 13, 512)
66048 activation_12[0][0]

res3a_branch1 (Conv2D) (None, 13, 13, 512)
131584 activation_10[0][0]

bn3a_branch2c (BatchNormalizati (None, 13, 13, 512)
2048 res3a_branch2c[0][0]

bn3a_branch1 (BatchNormalizatio (None, 13, 13, 512)
2048 res3a_branch1[0][0]

add_4 (Add) (None, 13, 13, 512)
0 bn3a_branch2c[0][0]

bn3a_branch1[0][0]

activation_13 (Activation) (None, 13, 13, 512)
0 add_4[0][0]

res3b_branch2a (Conv2D) (None, 13, 13, 128)
65664 activation_13[0][0]

bn3b_branch2a (BatchNormalizati (None, 13, 13, 128)
512 res3b_branch2a[0][0]

activation_14 (Activation) (None, 13, 13, 128)
0 bn3b_branch2a[0][0]

res3b_branch2b (Conv2D) (None, 13, 13, 128)
147584 activation_14[0][0]

bn3b_branch2b (BatchNormalizati (None, 13, 13, 128)
512 res3b_branch2b[0][0]

activation_15 (Activation) (None, 13, 13, 128)
0 bn3b_branch2b[0][0]

res3b_branch2c (Conv2D) (None, 13, 13, 512)

66048 activation_15[0][0]

bn3b_branch2c (BatchNormalizati (None, 13, 13, 512)
2048 res3b_branch2c[0][0]

add_5 (Add) (None, 13, 13, 512)
0 bn3b_branch2c[0][0]

activation_13[0][0]

activation_16 (Activation) (None, 13, 13, 512)
0 add_5[0][0]

res3c_branch2a (Conv2D) (None, 13, 13, 128)
65664 activation_16[0][0]

bn3c_branch2a (BatchNormalizati (None, 13, 13, 128)
512 res3c_branch2a[0][0]

activation_17 (Activation) (None, 13, 13, 128)
0 bn3c_branch2a[0][0]

res3c_branch2b (Conv2D) (None, 13, 13, 128)
147584 activation_17[0][0]

bn3c_branch2b (BatchNormalizati (None, 13, 13, 128)
512 res3c_branch2b[0][0]

activation_18 (Activation) (None, 13, 13, 128)
0 bn3c_branch2b[0][0]

res3c_branch2c (Conv2D) (None, 13, 13, 512)
66048 activation_18[0][0]

bn3c_branch2c (BatchNormalizati (None, 13, 13, 512)
2048 res3c_branch2c[0][0]

add_6 (Add) (None, 13, 13, 512)
0 bn3c_branch2c[0][0]

activation_16[0][0]

activation_19 (Activation) (None, 13, 13, 512)
0 add_6[0][0]

res3d_branch2a (Conv2D) (None, 13, 13, 128)

```
res3d_branch2a (Conv2D)           (None, 13, 13, 128)
65664      activation_19[0][0]
-----
bn3d_branch2a (BatchNormalizati (None, 13, 13, 128)
512        res3d_branch2a[0][0]
-----
activation_20 (Activation)       (None, 13, 13, 128)
0          bn3d_branch2a[0][0]
-----
res3d_branch2b (Conv2D)           (None, 13, 13, 128)
147584     activation_20[0][0]
-----
bn3d_branch2b (BatchNormalizati (None, 13, 13, 128)
512        res3d_branch2b[0][0]
-----
activation_21 (Activation)       (None, 13, 13, 128)
0          bn3d_branch2b[0][0]
-----
res3d_branch2c (Conv2D)           (None, 13, 13, 512)
66048      activation_21[0][0]
-----
bn3d_branch2c (BatchNormalizati (None, 13, 13, 512)
2048        res3d_branch2c[0][0]
-----
add_7 (Add)                      (None, 13, 13, 512)
0          bn3d_branch2c[0][0]
activation_19[0][0]
-----
activation_22 (Activation)       (None, 13, 13, 512)
0          add_7[0][0]
-----
res4a_branch2a (Conv2D)           (None, 7, 7, 256)
131328     activation_22[0][0]
-----
bn4a_branch2a (BatchNormalizati (None, 7, 7, 256)
1024        res4a_branch2a[0][0]
-----
activation_23 (Activation)       (None, 7, 7, 256)
0          bn4a_branch2a[0][0]
-----
res4a_branch2b (Conv2D)           (None, 7, 7, 256)
590080     activation_23[0][0]
-----
bn4a_branch2b (BatchNormalizati (None, 7, 7, 256)
1024        res4a_branch2b[0][0]
```

activation_24 (Activation) (None, 7, 7, 256)
0 bn4a_branch2b[0][0]

res4a_branch2c (Conv2D) (None, 7, 7, 1024)
263168 activation_24[0][0]

res4a_branch1 (Conv2D) (None, 7, 7, 1024)
525312 activation_22[0][0]

bn4a_branch2c (BatchNormalizati (None, 7, 7, 1024)
4096 res4a_branch2c[0][0]

bn4a_branch1 (BatchNormalizatio (None, 7, 7, 1024)
4096 res4a_branch1[0][0]

add_8 (Add) (None, 7, 7, 1024)
0 bn4a_branch2c[0][0]

bn4a_branch1[0][0]

activation_25 (Activation) (None, 7, 7, 1024)
0 add_8[0][0]

res4b_branch2a (Conv2D) (None, 7, 7, 256)
262400 activation_25[0][0]

bn4b_branch2a (BatchNormalizati (None, 7, 7, 256)
1024 res4b_branch2a[0][0]

activation_26 (Activation) (None, 7, 7, 256)
0 bn4b_branch2a[0][0]

res4b_branch2b (Conv2D) (None, 7, 7, 256)
590080 activation_26[0][0]

bn4b_branch2b (BatchNormalizati (None, 7, 7, 256)
1024 res4b_branch2b[0][0]

activation_27 (Activation) (None, 7, 7, 256)
0 bn4b_branch2b[0][0]

res4b_branch2c (Conv2D) (None, 7, 7, 1024)
263168 activation_27[0][0]

bn4b_branch2c (BatchNormalizati (None, 7, 7, 1024)
4096 res4b_branch2c[0][0]

add_9 (Add) (None, 7, 7, 1024)
0 bn4b_branch2c[0][0]

activation_25[0][0]

activation_28 (Activation) (None, 7, 7, 1024)
0 add_9[0][0]

res4c_branch2a (Conv2D) (None, 7, 7, 256)
262400 activation_28[0][0]

bn4c_branch2a (BatchNormalizati (None, 7, 7, 256)
1024 res4c_branch2a[0][0]

activation_29 (Activation) (None, 7, 7, 256)
0 bn4c_branch2a[0][0]

res4c_branch2b (Conv2D) (None, 7, 7, 256)
590080 activation_29[0][0]

bn4c_branch2b (BatchNormalizati (None, 7, 7, 256)
1024 res4c_branch2b[0][0]

activation_30 (Activation) (None, 7, 7, 256)
0 bn4c_branch2b[0][0]

res4c_branch2c (Conv2D) (None, 7, 7, 1024)
263168 activation_30[0][0]

bn4c_branch2c (BatchNormalizati (None, 7, 7, 1024)
4096 res4c_branch2c[0][0]

add_10 (Add) (None, 7, 7, 1024)
0 bn4c_branch2c[0][0]

activation_28[0][0]

activation_31 (Activation) (None, 7, 7, 1024)
0 add_10[0][0]

res4d_branch2a (Conv2D) (None, 7, 7, 256)
262400 activation_31[0][0]

```
bn4d_branch2a (BatchNormalizati (None, 7, 7, 256)
1024      res4d_branch2a[0][0]
-----
activation_32 (Activation)      (None, 7, 7, 256)
0          bn4d_branch2a[0][0]
-----
res4d_branch2b (Conv2D)         (None, 7, 7, 256)
590080      activation_32[0][0]
-----
bn4d_branch2b (BatchNormalizati (None, 7, 7, 256)
1024      res4d_branch2b[0][0]
-----
activation_33 (Activation)      (None, 7, 7, 256)
0          bn4d_branch2b[0][0]
-----
res4d_branch2c (Conv2D)         (None, 7, 7, 1024)
263168      activation_33[0][0]
-----
bn4d_branch2c (BatchNormalizati (None, 7, 7, 1024)
4096      res4d_branch2c[0][0]
-----
add_11 (Add)                  (None, 7, 7, 1024)
0          bn4d_branch2c[0][0]
activation_31[0][0]
-----
activation_34 (Activation)      (None, 7, 7, 1024)
0          add_11[0][0]
-----
res4e_branch2a (Conv2D)         (None, 7, 7, 256)
262400      activation_34[0][0]
-----
bn4e_branch2a (BatchNormalizati (None, 7, 7, 256)
1024      res4e_branch2a[0][0]
-----
activation_35 (Activation)      (None, 7, 7, 256)
0          bn4e_branch2a[0][0]
-----
res4e_branch2b (Conv2D)         (None, 7, 7, 256)
590080      activation_35[0][0]
-----
bn4e_branch2b (BatchNormalizati (None, 7, 7, 256)
1024      res4e_branch2b[0][0]
-----
activation_36 (Activation)      (None, 7, 7, 256)
0          bn4e_branch2b[0][0]
```

res4e_branch2c (Conv2D) (None, 7, 7, 1024)
263168 activation_36[0][0]

bn4e_branch2c (BatchNormalizati (None, 7, 7, 1024)
4096 res4e_branch2c[0][0]

add_12 (Add) (None, 7, 7, 1024)
0 bn4e_branch2c[0][0]

activation_34[0][0]

activation_37 (Activation) (None, 7, 7, 1024)
0 add_12[0][0]

res4f_branch2a (Conv2D) (None, 7, 7, 256)
262400 activation_37[0][0]

bn4f_branch2a (BatchNormalizati (None, 7, 7, 256)
1024 res4f_branch2a[0][0]

activation_38 (Activation) (None, 7, 7, 256)
0 bn4f_branch2a[0][0]

res4f_branch2b (Conv2D) (None, 7, 7, 256)
590080 activation_38[0][0]

bn4f_branch2b (BatchNormalizati (None, 7, 7, 256)
1024 res4f_branch2b[0][0]

activation_39 (Activation) (None, 7, 7, 256)
0 bn4f_branch2b[0][0]

res4f_branch2c (Conv2D) (None, 7, 7, 1024)
263168 activation_39[0][0]

bn4f_branch2c (BatchNormalizati (None, 7, 7, 1024)
4096 res4f_branch2c[0][0]

add_13 (Add) (None, 7, 7, 1024)
0 bn4f_branch2c[0][0]

activation_37[0][0]

activation_40 (Activation) (None, 7, 7, 1024)
0 add_13[0][0]

res5a_branch2a (Conv2D) (None, 4, 4, 512)
524800 activation_40[0][0]

bn5a_branch2a (BatchNormalizati (None, 4, 4, 512)
2048 res5a_branch2a[0][0]

activation_41 (Activation) (None, 4, 4, 512)
0 bn5a_branch2a[0][0]

res5a_branch2b (Conv2D) (None, 4, 4, 512)
2359808 activation_41[0][0]

bn5a_branch2b (BatchNormalizati (None, 4, 4, 512)
2048 res5a_branch2b[0][0]

activation_42 (Activation) (None, 4, 4, 512)
0 bn5a_branch2b[0][0]

res5a_branch2c (Conv2D) (None, 4, 4, 2048)
1050624 activation_42[0][0]

res5a_branch1 (Conv2D) (None, 4, 4, 2048)
2099200 activation_40[0][0]

bn5a_branch2c (BatchNormalizati (None, 4, 4, 2048)
8192 res5a_branch2c[0][0]

bn5a_branch1 (BatchNormalizatio (None, 4, 4, 2048)
8192 res5a_branch1[0][0]

add_14 (Add) (None, 4, 4, 2048)
0 bn5a_branch2c[0][0]
bn5a_branch1[0][0]

activation_43 (Activation) (None, 4, 4, 2048)
0 add_14[0][0]

res5b_branch2a (Conv2D) (None, 4, 4, 512)
1049088 activation_43[0][0]

bn5b_branch2a (BatchNormalizati (None, 4, 4, 512)
2048 res5b_branch2a[0][0]

```
activation_44 (Activation)      (None, 4, 4, 512)
0          bn5b_branch2a[0][0]

res5b_branch2b (Conv2D)         (None, 4, 4, 512)
2359808      activation_44[0][0]

bn5b_branch2b (BatchNormalizati (None, 4, 4, 512)
2048        res5b_branch2b[0][0]

activation_45 (Activation)      (None, 4, 4, 512)
0          bn5b_branch2b[0][0]

res5b_branch2c (Conv2D)         (None, 4, 4, 2048)
1050624      activation_45[0][0]

bn5b_branch2c (BatchNormalizati (None, 4, 4, 2048)
8192        res5b_branch2c[0][0]

add_15 (Add)                  (None, 4, 4, 2048)
0          bn5b_branch2c[0][0]

activation_43[0][0]

activation_46 (Activation)      (None, 4, 4, 2048)
0          add_15[0][0]
```

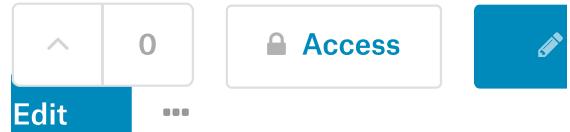
```
res5c_branch2a (Conv2D)         (None, 4, 4, 512)
1049088      activation_46[0][0]

bn5c_branch2a (BatchNormalizati (None, 4, 4, 512)
2048        res5c_branch2a[0][0]
```

Resnet50 pretrained (0.406)

Python notebook using data from [multiple data sources](#) · 34 views ·  multiple data sources

 Edit tags



Version 4

 5 commits

forked from Whales draft

Notebook

Data

Output

Log

Comments

```
res5c_branch2b (Conv2D)         (None, 4, 4, 512)
2359808      activation_47[0][0]

bn5c_branch2b (BatchNormalizati (None, 4, 4, 512)
2048        res5c_branch2b[0][0]

activation_48 (Activation)      (None, 4, 4, 512)
0          bn5c_branch2b[0][0]

res5c_branch2c (Conv2D)         (None, 4, 4, 2048)
```

```
1050624      activation_48[0][0]
-----
bn5c_branch2c (BatchNormalizati (None, 4, 4, 2048)
8192        res5c_branch2c[0][0]
-----
add_16 (Add)          (None, 4, 4, 2048)
0            bn5c_branch2c[0][0]
activation_46[0][0]
-----
activation_49 (Activation)    (None, 4, 4, 2048)
0            add_16[0][0]
-----
avg_pool (GlobalAveragePooling2 (None, 2048)
0            activation_49[0][0]
-----
dropout_1 (Dropout)        (None, 2048)
0            avg_pool[0][0]
-----
dense_1 (Dense)           (None, 5004)
10253196    dropout_1[0][0]
=====
=====
Total params: 33,840,908
Trainable params: 33,787,788
Non-trainable params: 53,120
```

Out[22]:

0

Train more with augmented data

 Notebook  Data  Output  Log  Comments

In [23]:

```
ROTATE = 20
EPOCHS = 15
BATCH = 100

train_datagen = ImageDataGenerator(
    fill_mode='nearest',
    validation_split = 0.1,
    rotation_range=ROTATE,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=False)

# compute quantities required for featurewise normalization
# first pass over training data to calculate statistics
# note that validation data is also used here
```

```

# (std, mean, and principal components if ZCA whitening
# is applied)
train_datagen.fit(X)

# fits the model on batches with real-time data augmentation:
history = model.fit_generator(train_datagen.flow(X, y
, batch_size=BATCH),
                               steps_per_epoch=len(X) / BATCH, e
pochs=EPOCHS)

```

```

Epoch 1/15
157/156 [=====] - 137s 870m
s/step - loss: 3.4658 - acc: 0.2786
Epoch 2/15
157/156 [=====] - 120s 767m
s/step - loss: 1.9396 - acc: 0.5194
Epoch 3/15
157/156 [=====] - 121s 768m
s/step - loss: 1.3341 - acc: 0.6511
Epoch 4/15
157/156 [=====] - 120s 767m
s/step - loss: 0.9436 - acc: 0.7439
Epoch 5/15
157/156 [=====] - 121s 768m
s/step - loss: 0.7496 - acc: 0.7895
Epoch 6/15
157/156 [=====] - 120s 767m
s/step - loss: 0.6095 - acc: 0.8284
Epoch 7/15
157/156 [=====] - 120s 767m
s/step - loss: 0.4930 - acc: 0.8614
Epoch 8/15
157/156 [=====] - 120s 766m
s/step - loss: 0.4119 - acc: 0.8803
Epoch 9/15
157/156 [=====] - 121s 768m
s/step - loss: 0.3699 - acc: 0.8908
Epoch 10/15
157/156 [=====] - 120s 765m
s/step - loss: 0.3144 - acc: 0.9101
Epoch 11/15
157/156 [=====] - 120s 766m
s/step - loss: 0.2795 - acc: 0.9171
Epoch 12/15
157/156 [=====] - 120s 766m
s/step - loss: 0.2506 - acc: 0.9283
Epoch 13/15
157/156 [=====] - 120s 767m
s/step - loss: 0.2363 - acc: 0.9286
Epoch 14/15
157/156 [=====] - 120s 767m
s/step - loss: 0.2210 - acc: 0.9344
Epoch 15/15
157/156 [=====] - 120s 766m
s/step - loss: 0.2018 - acc: 0.9407

```

In [24]:

```
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
```

Plot train results

In [25]:

```
def plot_accuracy(history, should_plot_val = 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')

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

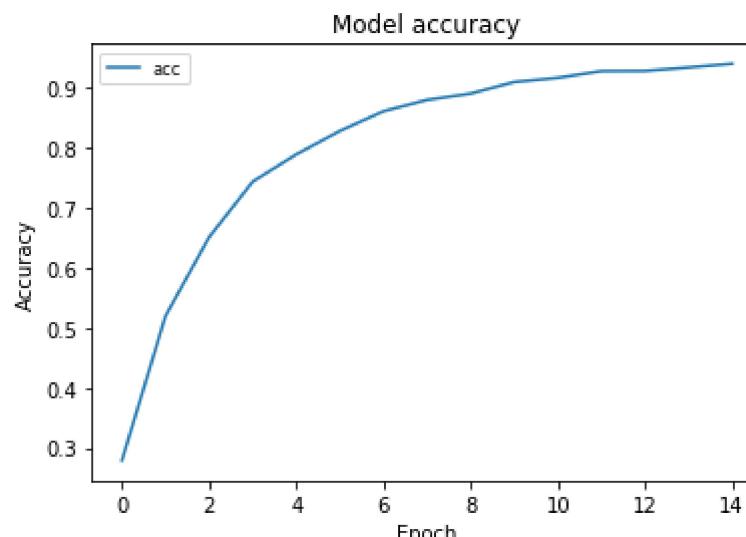
def plot_loss(history, should_plot_val = False):
    loss = history.history[ 'loss' ]
    l1 = plt.plot(loss, label='loss')

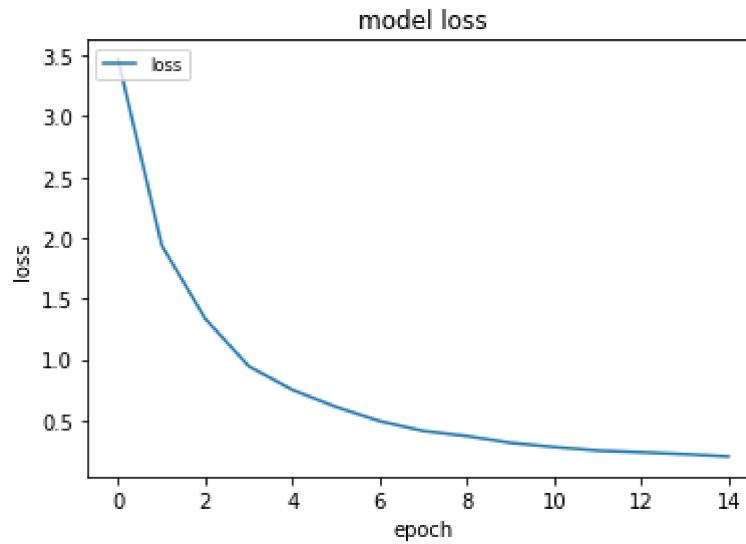
    if should_plot_val:
        val_loss = history.history[ 'val_loss' ]
        plt.plot(val_loss, label='val_loss')

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

In [26]:

```
plot_accuracy(history)
plot_loss(history)
```





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
```

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 [=====] - 163s 20m
s/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)
```

```
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_0
02c810' 'w_0032a46'
 'w_003bae6']
[ 'new_whale' 'w_0003639' 'w_0003c59' 'w_0027efa' 'w_0
0289b1' '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)

    # Add a column with the best threshold probability
```

```
to the predictions  
column_to_add = np.zeros(shape_to_add) + best_thr  
eshold
```

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 ... ▼ 📁 resnet50_pretrained ▀ su... 7960 x 2 ▀ Model_ResNet5... ▀ Weights_ResNet... » 📁 whales_cropped	 resnet50_pretrained  Last Updated: 12 days ago (Version 1) About this Dataset No description yet
---	--

Output Files

New Dataset

New Kernel

Download All



Output Files ▀ submission.csv ▀ Model_ResNet50_3... ▀ predicted_class_in... ▀ predictions.npy ▀ test_class_indices.... ▀ test_filenames_gen... ▀ Weights_ResNet50...	About this file This file was created from a Kernel, it does not have a description. 
--	---

▀ submission.csv



1	Image	Id
2	00028a005.jpg	new_whale w_df6b050 w_0e9f07a w_dec7ffd w_d6666c9
3	000dcf7d8.jpg	w_e3956f5 new_whale w_4132bb8 w_8431ae8 w_a491df7
4	000e7c7df.jpg	w_6658081 new_whale w_5b463cf w_4c218b5 w_9b89c88

5	0019c34f4.jpg	new_whale w_59052ad w_f765256 w_5a3e0de w_5ff2049	
6	001a4d292.jpg	new_whale w_eaa5dbb w_2a946f2 w_26b24d4 w_9e18daa	
7	00247bc36.jpg	new_whale w_2d6bb89 w_0a31a3a w_900c68f w_cda3971	
8	0027089a4.jpg	w_7e2eb3d new_whale w_faf5287 w_d405854 w_5b227ec	
9	002de4d94.jpg	new_whale w_633ead1 w_12e7c23 w_3a63e33 w_691b684	

Run Info

Succeeded	True	Run Time	2177.5 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 script.ipynb to html
3.0s    2  [NbConvertApp] Executing notebook with kernel: python3
115.5s   3  2019-01-21 19:49:25.096020: 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
115.5s   4  2019-01-21 19:49:25.096551: 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
              2019-01-21 19:49:25.096578: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:1511]
              Adding visible gpu devices: 0
115.8s   5  2019-01-21 19:49:25.424761: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:982]
              Device interconnect StreamExecutor with strength 1 edge
              matrix:
              2019-01-21 19:49:25.424855: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:988]
              0
              2019-01-21 19:49:25.424867: I
              tensorflow/core/common_runtime/gpu/gpu_device.cc:1001] 0:
              N
115.8s   6  2019-01-21 19:49:25.425277: 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)

```

```
2177.0s    7 [NbConvertApp] Support files will be in __results__files/
2177.0s    8 [NbConvertApp] Making directory __results__files
[NbConvertApp] Making directory __results__files
[NbConvertApp] Making directory __results__files
[NbConvertApp] Writing 358391 bytes to __results__.html
2177.0s    9
2177.0s   11 Complete. Exited with code 0.
```

Comments (0)



Click here to enter a comment...

