

Submission**✓ Ran successfully**

Submitted by NinaV 16 days ago

Public Score

0.406

```
In [1]:  
# This Python 3 environment comes with many helpful analytics libr  
# It is defined by the kaggle/python docker image: https://github.  
# 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  
  
# Input data files are available in the "../input/" directory.  
# For example, running this (by clicking run or pressing Shift+Ent  
in the input directory  
  
import os  
print(os.listdir("../input"))  
  
# Any results you write to the current directory are saved as outp
```

```
[ 'humpback-whale-identification', 'metadata' ]
```

```
In [2]:  
HW = 'humpback-whale-identification'  
TRAIN = '../input/humpback-whale-identification/train/'  
TEST = '../input/humpback-whale-identification/test/'  
LABELS = '../input/humpback-whale-identification/train.csv'  
SAMPLE_SUB = '../input/humpback-whale-identification/sample_subm  
BBOX = '../input/metadata/bounding_boxes.csv'  
  
whales_bbox = pd.read_csv(BBOX).set_index('Image')  
  
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/whales-draft/' + MODEL_F  
WEIGHTS = '../input/whales-draft/' + WEIGHTS_F
```

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

```
Out[4]:
```

	Image	Id
count	25361	25361
unique	25361	5005
top	54f94899b.jpg	new_whale
freq	1	9664

In [5]:

```
import random
from IPython.display import Image
# whales_train_modified.describe()

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

0d05262d7.jpg

Out[5]:



In [6]:

```
train_images = train.set_index('Image')
new_whale_train = train_images[train_images.Id == "new_whale"]
# whales_train = train_images[~(train_images.Id == "new_whale")]
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	64c977440.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.applications.imagenet_utils import preprocess_input
from keras.layers import Input, Dense, Activation, BatchNormalization
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]:

```
def prepareImages(data, m, dataset):
    print("Preparing images")
    X_train = np.zeros((m, 100, 100, 3))
    count = 0

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

        # load bounding box
        bbox = whales_bbox.loc[fig]
        x0, y0, x1, y1 = bbox['x0'], bbox['y0'], bbox['x1'], bbox['y1']

        # crop according to bounding box
        if not (x0 >= x1 or y0 >= y1):
            img = img.crop((x0, y0, x1, y1))
        #load images into images of size 100x100x3
        img = img.resize((100, 100))
        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/generic.py:44  
ng:  
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/p  
ng.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_encode  
#     print(integer_encoded)  
    onehot_encoded = onehot_encoder.fit_transform(integer_encode  
#     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/_en  
ning: The handling of integer data will change in version 0.22.  
s are determined based on the range [0, max(values)], while in t  
etermined based on the unique values.  
If you want the future behaviour and silence this warning, you c  
='auto'.  
In case you used a LabelEncoder before this OneHotEncoder to con  
tegers, 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], "humpback  
ain")  
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_in  
from keras.losses import binary_crossentropy
```

```
CLASSES = 5004
```

```

# setup model
base_model = ResNet50(weights='imagenet', include_top=False, input_tensor=None)

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()

```

```

/opt/conda/lib/python3.6/site-packages/keras_applications/resnet50.py:105: UserWarning: The output shape of `ResNet50(include_top=False)` has been changed in Keras 2.0.0. Please use `include_top=True` if you want to use this model as a feature extractor.
  warnings.warn('The output shape of `ResNet50(include_top=False)` has been changed in Keras 2.0.0. Please use `include_top=True` if you want to use this model as a feature extractor.')

```

```

  Downloading data from https://github.com/fchollet/deep-learning-models/v0.2/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
  94658560/94653016 [=====] - 1s 0us/step

```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 100, 100, 3)	0
conv1_pad (ZeroPadding2D)	(None, 106, 106, 3)	0
conv1 (Conv2D)	(None, 50, 50, 64)	9472
bn_conv1 (BatchNormalization)	(None, 50, 50, 64)	256
activation_1 (Activation)	(None, 50, 50, 64)	0
pool1_pad (ZeroPadding2D)	(None, 52, 52, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 25, 25, 64)	0
res2a_branch2a (Conv2D)	(None, 25, 25, 64)	4160

bn2a_branch2a (BatchNormalizati (None, 25, 25, 64) 256

activation_2 (Activation) (None, 25, 25, 64) 0

res2a_branch2b (Conv2D) (None, 25, 25, 64) 36928

bn2a_branch2b (BatchNormalizati (None, 25, 25, 64) 256

activation_3 (Activation) (None, 25, 25, 64) 0

res2a_branch2c (Conv2D) (None, 25, 25, 256) 16640

res2a_branch1 (Conv2D) (None, 25, 25, 256) 16640

bn2a_branch2c (BatchNormalizati (None, 25, 25, 256) 1024

bn2a_branch1 (BatchNormalizatio (None, 25, 25, 256) 1024

add_1 (Add) (None, 25, 25, 256) 0

activation_4 (Activation) (None, 25, 25, 256) 0

res2b_branch2a (Conv2D) (None, 25, 25, 64) 16448

bn2b_branch2a (BatchNormalizati (None, 25, 25, 64) 256

activation_5 (Activation) (None, 25, 25, 64) 0

res2b_branch2b (Conv2D) (None, 25, 25, 64) 36928

bn2b_branch2b (BatchNormalizati (None, 25, 25, 64) 256

activation_6 (Activation)	(None, 25, 25, 64)	0
res2b_branch2c (Conv2D)	(None, 25, 25, 256)	16640
bn2b_branch2c (BatchNormalizati	(None, 25, 25, 256)	1024
add_2 (Add)	(None, 25, 25, 256)	0
activation_7 (Activation)	(None, 25, 25, 256)	0
res2c_branch2a (Conv2D)	(None, 25, 25, 64)	16448
bn2c_branch2a (BatchNormalizati	(None, 25, 25, 64)	256
activation_8 (Activation)	(None, 25, 25, 64)	0
res2c_branch2b (Conv2D)	(None, 25, 25, 64)	36928
bn2c_branch2b (BatchNormalizati	(None, 25, 25, 64)	256
activation_9 (Activation)	(None, 25, 25, 64)	0
res2c_branch2c (Conv2D)	(None, 25, 25, 256)	16640
bn2c_branch2c (BatchNormalizati	(None, 25, 25, 256)	1024
add_3 (Add)	(None, 25, 25, 256)	0
activation_10 (Activation)	(None, 25, 25, 256)	0
res3a_branch2a (Conv2D)	(None, 13, 13, 128)	32896

bn3a_branch2a (BatchNormalizati (None, 13, 13, 128) 512

activation_11 (Activation) (None, 13, 13, 128) 0

res3a_branch2b (Conv2D) (None, 13, 13, 128) 147584

bn3a_branch2b (BatchNormalizati (None, 13, 13, 128) 512

activation_12 (Activation) (None, 13, 13, 128) 0

res3a_branch2c (Conv2D) (None, 13, 13, 512) 66048

res3a_branch1 (Conv2D) (None, 13, 13, 512) 131584

bn3a_branch2c (BatchNormalizati (None, 13, 13, 512) 2048

bn3a_branch1 (BatchNormalizatio (None, 13, 13, 512) 2048

add_4 (Add) (None, 13, 13, 512) 0

activation_13 (Activation) (None, 13, 13, 512) 0

res3b_branch2a (Conv2D) (None, 13, 13, 128) 65664

bn3b_branch2a (BatchNormalizati (None, 13, 13, 128) 512

activation_14 (Activation) (None, 13, 13, 128) 0

res3b_branch2b (Conv2D) (None, 13, 13, 128) 147584

bn3b_branch2b (BatchNormalizati (None, 13, 13, 128) 512

activation_15 (Activation) (None, 13, 13, 128) 0

res3b_branch2c (Conv2D) (None, 13, 13, 512) 66048

bn3b_branch2c (BatchNormalizati (None, 13, 13, 512) 2048

add_5 (Add) (None, 13, 13, 512) 0

activation_16 (Activation) (None, 13, 13, 512) 0

res3c_branch2a (Conv2D) (None, 13, 13, 128) 65664

bn3c_branch2a (BatchNormalizati (None, 13, 13, 128) 512

activation_17 (Activation) (None, 13, 13, 128) 0

res3c_branch2b (Conv2D) (None, 13, 13, 128) 147584

bn3c_branch2b (BatchNormalizati (None, 13, 13, 128) 512

activation_18 (Activation) (None, 13, 13, 128) 0

res3c_branch2c (Conv2D) (None, 13, 13, 512) 66048

bn3c_branch2c (BatchNormalizati (None, 13, 13, 512) 2048

add_6 (Add) (None, 13, 13, 512) 0

activation_19 (Activation) (None, 13, 13, 512) 0

res3d_branch2a (Conv2D) (None, 13, 13, 128) 65664

bn3d_branch2a (BatchNormalizati (None, 13, 13, 128) 512

activation_20 (Activation) (None, 13, 13, 128) 0

res3d_branch2b (Conv2D) (None, 13, 13, 128) 147584

bn3d_branch2b (BatchNormalizati (None, 13, 13, 128) 512

activation_21 (Activation) (None, 13, 13, 128) 0

res3d_branch2c (Conv2D) (None, 13, 13, 512) 66048

bn3d_branch2c (BatchNormalizati (None, 13, 13, 512) 2048

add_7 (Add) (None, 13, 13, 512) 0

activation_22 (Activation) (None, 13, 13, 512) 0

res4a_branch2a (Conv2D) (None, 7, 7, 256) 131328

bn4a_branch2a (BatchNormalizati (None, 7, 7, 256) 1024

activation_23 (Activation) (None, 7, 7, 256) 0

res4a_branch2b (Conv2D) (None, 7, 7, 256) 590080

bn4a_branch2b (BatchNormalizati (None, 7, 7, 256) 1024

activation_24 (Activation) (None, 7, 7, 256) 0

res4a_branch2c (Conv2D) (None, 7, 7, 1024) 263168

res4a_branch1 (Conv2D) (None, 7, 7, 1024) 525312

bn4a_branch2c (BatchNormalizati (None, 7, 7, 1024) 4096

bn4a_branch1 (BatchNormalizatio (None, 7, 7, 1024) 4096

add_8 (Add) (None, 7, 7, 1024) 0

activation_25 (Activation) (None, 7, 7, 1024) 0

res4b_branch2a (Conv2D) (None, 7, 7, 256) 262400

bn4b_branch2a (BatchNormalizati (None, 7, 7, 256) 1024

activation_26 (Activation) (None, 7, 7, 256) 0

res4b_branch2b (Conv2D) (None, 7, 7, 256) 590080

bn4b_branch2b (BatchNormalizati (None, 7, 7, 256) 1024

activation_27 (Activation) (None, 7, 7, 256) 0

res4b_branch2c (Conv2D) (None, 7, 7, 1024) 263168

bn4b_branch2c (BatchNormalizati (None, 7, 7, 1024) 4096

add_9 (Add) (None, 7, 7, 1024) 0

activation_28 (Activation) (None, 7, 7, 1024) 0

res4c_branch2a (Conv2D) (None, 7, 7, 256) 262400

bn4c_branch2a (BatchNormalizati (None, 7, 7, 256) 1024

activation_29 (Activation) (None, 7, 7, 256) 0

res4c_branch2b (Conv2D) (None, 7, 7, 256) 590080

Whales draft

Python notebook using data from [multiple data sources](#) · 153 views ·  multiple data sources  Edit tags



 Edit

...

Version 12

 22 commits

activation_30 (Activation) (None, 7, 7, 256) 0

res4c_branch2c (Conv2D) (None, 7, 7, 1024) 263168

Data

Output

Log

Comments

bn4c_branch2c (BatchNormalizati (None, 7, 7, 1024) 4096

add_10 (Add) (None, 7, 7, 1024) 0

activation_31 (Activation) (None, 7, 7, 1024) 0

res4d_branch2a (Conv2D) (None, 7, 7, 256) 262400

bn4d_branch2a (BatchNormalizati (None, 7, 7, 256) 1024

activation_32 (Activation) (None, 7, 7, 256) 0

res4d_branch2b (Conv2D) (None, 7, 7, 256) 590080

bn4d_branch2b (BatchNormalizati (None, 7, 7, 256) 1024

activation_33 (Activation) (None, 7, 7, 256) 0

```
res4d_branch2c (Conv2D)           (None, 7, 7, 1024) 263168
-----
bn4d_branch2c (BatchNormalizati (None, 7, 7, 1024) 4096
-----
add_11 (Add)                      (None, 7, 7, 1024) 0
```

 Notebook  Data  Output  Log  Comments

```
res4e_branch2a (Conv2D)           (None, 7, 7, 256) 262400
-----
```

```
bn4e_branch2a (BatchNormalizati (None, 7, 7, 256) 1024
-----
```

```
activation_35 (Activation)        (None, 7, 7, 256) 0
-----
```

```
res4e_branch2b (Conv2D)           (None, 7, 7, 256) 590080
-----
```

```
bn4e_branch2b (BatchNormalizati (None, 7, 7, 256) 1024
-----
```

```
activation_36 (Activation)        (None, 7, 7, 256) 0
-----
```

```
res4e_branch2c (Conv2D)           (None, 7, 7, 1024) 263168
-----
```

```
bn4e_branch2c (BatchNormalizati (None, 7, 7, 1024) 4096
-----
```

```
add_12 (Add)                      (None, 7, 7, 1024) 0
-----
```

```
activation_37 (Activation)        (None, 7, 7, 1024) 0
-----
```

```
res4f_branch2a (Conv2D)           (None, 7, 7, 256) 262400
-----
```

```
bn4f_branch2a (BatchNormalizati (None, 7, 7, 256) 1024
```

activation_38 (Activation)	(None, 7, 7, 256)	0
res4f_branch2b (Conv2D)	(None, 7, 7, 256)	590080
bn4f_branch2b (BatchNormalizati	(None, 7, 7, 256)	1024
activation_39 (Activation)	(None, 7, 7, 256)	0
res4f_branch2c (Conv2D)	(None, 7, 7, 1024)	263168
bn4f_branch2c (BatchNormalizati	(None, 7, 7, 1024)	4096
add_13 (Add)	(None, 7, 7, 1024)	0
activation_40 (Activation)	(None, 7, 7, 1024)	0
res5a_branch2a (Conv2D)	(None, 4, 4, 512)	524800
bn5a_branch2a (BatchNormalizati	(None, 4, 4, 512)	2048
activation_41 (Activation)	(None, 4, 4, 512)	0
res5a_branch2b (Conv2D)	(None, 4, 4, 512)	2359808
bn5a_branch2b (BatchNormalizati	(None, 4, 4, 512)	2048
activation_42 (Activation)	(None, 4, 4, 512)	0
res5a_branch2c (Conv2D)	(None, 4, 4, 2048)	1050624
res5a_branch1 (Conv2D)	(None, 4, 4, 2048)	2099200

bn5a_branch2c (BatchNormalizati (None, 4, 4, 2048) 8192

bn5a_branch1 (BatchNormalizatio (None, 4, 4, 2048) 8192

add_14 (Add) (None, 4, 4, 2048) 0

activation_43 (Activation) (None, 4, 4, 2048) 0

res5b_branch2a (Conv2D) (None, 4, 4, 512) 1049088

bn5b_branch2a (BatchNormalizati (None, 4, 4, 512) 2048

activation_44 (Activation) (None, 4, 4, 512) 0

res5b_branch2b (Conv2D) (None, 4, 4, 512) 2359808

bn5b_branch2b (BatchNormalizati (None, 4, 4, 512) 2048

activation_45 (Activation) (None, 4, 4, 512) 0

res5b_branch2c (Conv2D) (None, 4, 4, 2048) 1050624

bn5b_branch2c (BatchNormalizati (None, 4, 4, 2048) 8192

add_15 (Add) (None, 4, 4, 2048) 0

activation_46 (Activation) (None, 4, 4, 2048) 0

res5c_branch2a (Conv2D) (None, 4, 4, 512) 1049088

bn5c_branch2a (BatchNormalizati (None, 4, 4, 512) 2048

```
-----  
activation_47 (Activation)      (None, 4, 4, 512)    0  
  
-----  
res5c_branch2b (Conv2D)        (None, 4, 4, 512)    2359808  
  
-----  
bn5c_branch2b (BatchNormalizati (None, 4, 4, 512)    2048  
  
-----  
activation_48 (Activation)      (None, 4, 4, 512)    0  
  
-----  
res5c_branch2c (Conv2D)        (None, 4, 4, 2048)   1050624  
  
-----  
bn5c_branch2c (BatchNormalizati (None, 4, 4, 2048)   8192  
  
-----  
add_16 (Add)                  (None, 4, 4, 2048)   0  
  
-----  
activation_49 (Activation)      (None, 4, 4, 2048)   0  
  
-----  
avg_pool (GlobalAveragePooling2 (None, 2048)        0  
  
-----  
dropout_1 (Dropout)            (None, 2048)         0  
  
-----  
dense_1 (Dense)               (None, 5004)        10253196  
=====
```

Total params: 33,840,908
Trainable params: 33,787,788
Non-trainable params: 53,120

In [19]:

```
# 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")
```

In [20]:

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

```
print(np.shape(wdices_train[1]))  
history = model.fit(X, y, epochs=30, batch_size=100, verbose=1)
```

```
Train set shape: (15697, 100, 100, 3)  
(15697,)  
Epoch 1/30  
15697/15697 [=====] - 142s 9ms/step - 1  
6  
Epoch 2/30  
15697/15697 [=====] - 123s 8ms/step - 1  
0  
Epoch 3/30  
15697/15697 [=====] - 124s 8ms/step - 1  
7  
Epoch 4/30  
15697/15697 [=====] - 124s 8ms/step - 1  
0  
Epoch 5/30  
15697/15697 [=====] - 123s 8ms/step - 1  
5  
Epoch 6/30  
15697/15697 [=====] - 123s 8ms/step - 1  
3  
Epoch 7/30  
15697/15697 [=====] - 123s 8ms/step - 1  
2  
Epoch 8/30  
15697/15697 [=====] - 123s 8ms/step - 1  
1  
Epoch 9/30  
15697/15697 [=====] - 123s 8ms/step - 1  
0  
Epoch 10/30  
15697/15697 [=====] - 123s 8ms/step - 1  
5  
Epoch 11/30  
15697/15697 [=====] - 123s 8ms/step - 1  
3  
Epoch 12/30  
15697/15697 [=====] - 123s 8ms/step - 1  
7  
Epoch 13/30  
15697/15697 [=====] - 123s 8ms/step - 1  
0  
Epoch 14/30  
15697/15697 [=====] - 124s 8ms/step - 1  
3  
Epoch 15/30  
15697/15697 [=====] - 124s 8ms/step - 1  
3  
Epoch 16/30  
15697/15697 [=====] - 124s 8ms/step - 1  
3  
Epoch 17/30  
15697/15697 [=====] - 123s 8ms/step - 1  
9  
Epoch 18/30  
15697/15697 [=====] - 124s 8ms/step - 1  
0  
Epoch 19/30  
15697/15697 [=====] - 124s 8ms/step - 1  
9  
Epoch 20/30  
15697/15697 [=====] - 124s 8ms/step - 1  
2  
Epoch 21/30  
15697/15697 [=====] - 124s 8ms/step - 1
```

```
15697/15697 [=====] - 124s 8ms/step - 1  
1  
Epoch 22/30  
15697/15697 [=====] - 123s 8ms/step - 1  
7  
Epoch 23/30  
15697/15697 [=====] - 124s 8ms/step - 1  
2  
Epoch 24/30  
15697/15697 [=====] - 124s 8ms/step - 1  
8  
Epoch 25/30  
15697/15697 [=====] - 124s 8ms/step - 1  
7  
Epoch 26/30  
15697/15697 [=====] - 124s 8ms/step - 1  
7  
Epoch 27/30  
15697/15697 [=====] - 124s 8ms/step - 1  
7  
Epoch 28/30  
15697/15697 [=====] - 124s 8ms/step - 1  
1  
Epoch 29/30  
15697/15697 [=====] - 124s 8ms/step - 1  
3  
Epoch 30/30  
15697/15697 [=====] - 123s 8ms/step - 1  
3
```

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]:

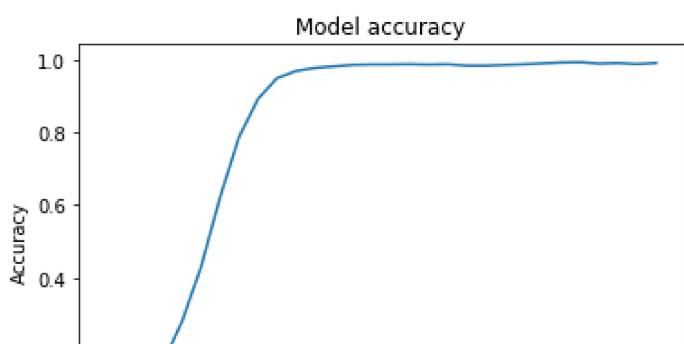
0

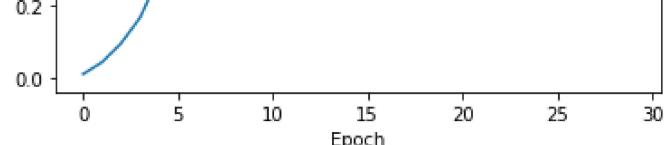
In [23]:

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

In [24]:

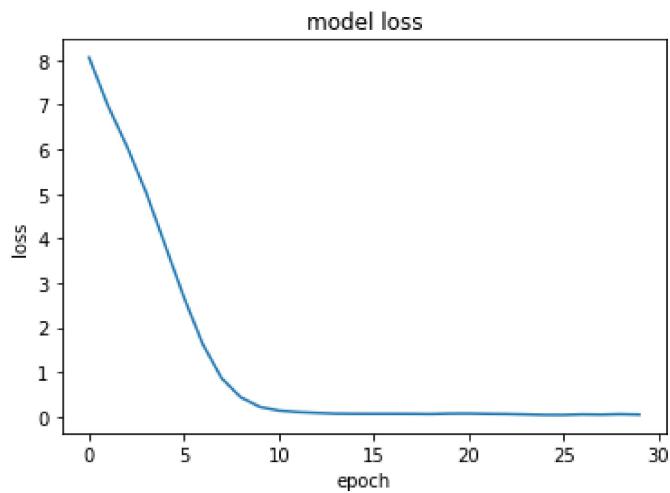
```
plt.plot(history.history['acc'])  
plt.title('Model accuracy')  
plt.ylabel('Accuracy')  
plt.xlabel('Epoch')  
plt.show()
```





In [25]:

```
plt.plot(history.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.show()
```



In [26]:

```
test = os.listdir(TEST)
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], "humpback-whale-ide
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
```

In [29]:

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

```
7960/7960 [=====] - 24s 3ms/step  
(7960, 5004)
```

```
np.save("test_files.npy", test) np.save("one_hot_predss.npy", predictions) np.save("one_hot_encoder.r
```

```
In [30]:  
    print(labels_list[:7])  
    labels_with_new_whale = ['new_whale']  
  
    for s in labels_list:  
        labels_with_new_whale.append(s)  
  
['w_0003639' 'w_0003c59' 'w_0027efa' 'w_00289b1' 'w_002c810' 'w_  
'w_003bae6']
```

```
In [31]:  
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_ma  
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 pre  
column_to_add = np.zeros(shape_to_add) + best_threshold  
predictions_w_new_whale = np.concatenate([column_to_add, pre  
return predictions_w_new_whale
```

```
In [32]:  
def create_results_csv(preds, labels_with_new_whale, test_file_n  
sample_df = pd.read_csv(SAMPLE_SUB)  
sample_images = list(sample_df.Image)  
  
print(test_file_names[:7])  
pred_list = [[labels_with_new_whale[i] for i in p.argsort()[-  
ds]  
  
pred_dic = dict((key, value) for (key, value) in zip(test_fi  
pred_list_for_test = [' '.join(pred_dic[id]) for id in sampl  
  
print(np.shape(pred_list))  
print(np.shape(test_file_names))  
df = pd.DataFrame({'Image': sample_images, 'Id': pred_list_f  
df.to_csv(output_filename, header=True, index=False)  
return df
```

```
In [33]:  
p = add_new_whale_to_predictions(predictions)  
test_df = create_results_csv(p, labels_with_new_whale, test, "su  
print(test_df[:10])
```

```
Average of max probabilities column:0.5941413  
['f28e2a7e7.jpg', 'f1a620ed9.jpg', '1613db994.jpg', 'dac7f10b4.j  
f32c1b9d.jpg', 'ef0628116.jpg']  
(7960, 5)  
(7960,)  
Image
```

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

0

Data

Data Sources » 🐋 Humpback Whale I... » 📁 metadata	 metadata Last Updated: a month ago (Version 3) About this Dataset No description yet
--	---

Output Files

New Dataset

New Kernel

Download All



Output Files grid submission.csv file Model_ResNet50_3... file Weights_ResNet50_...	About this file This file was created from a Kernel, it does not have a description. Submit to Competition
---	---

grid submission.csv



1	Image	Id
2	00028a005.jpg	new_whale w_abd456f w_2f67f0a w_4c07458 w_6e0a45b
3	000dcf7d8.jpg	new_whale w_50f9d24 w_4132bb8 w_e3956f5 w_6e7763b
4	000e7c7df.jpg	w_c5ee28a new_whale w_9ba4a9a w_c8d5ad5 w_90abfeb
5	0019c34f4.jpg	w_59052ad new_whale w_9c506f6 w_efc0825 w_5a3e0de
6	001a4d292.jpg	new_whale w_2a946f2 w_334d7eb w_d0a7600 w_bc6171b
7	00247bc36.jpg	new_whale w_900c68f w_43f3361 w_0ca546b w_73282ec
8	0027089a4.jpg	w_7e2eb3d new_whale w_2c94198 w_5b227ec w_30f2f0f

9	002de4d94.jpg	w_633ead1 new_whale w_94dc2b w_7a8ce16 w_7e9df91	
10	002f52f0c	new_whale	

Run Info

Succeeded	True	Run Time	4104.7 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
4.5s    1  [NbConvertApp] Converting notebook script.ipynb to html
4.5s    2  [NbConvertApp] Executing notebook with kernel: python3
197.2s   3  2019-01-16 22:35:20.615795: 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
197.2s   4  2019-01-16 22:35:20.618593: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:1411] 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
           2019-01-16 22:35:20.618638: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:1490] Adding
           visible gpu devices: 0
197.5s   5  2019-01-16 22:35:20.967398: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:971] Device
           interconnect StreamExecutor with strength 1 edge matrix:
           2019-01-16 22:35:20.967468: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:977]      0
           2019-01-16 22:35:20.967487: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:990] 0:  N
197.5s   6  2019-01-16 22:35:20.971149: I
           tensorflow/core/common_runtime/gpu/gpu_device.cc:1103] 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)
4104.0s   7  [NbConvertApp] Support files will be in __results__files/
           [NbConvertApp] Making directory __results__files
4104.0s   8  [NbConvertApp] Making directory __results__files
           [NbConvertApp] Making directory __results__files
           [NbConvertApp] Writing 352420 bytes to __results__.html
4104.0s   9
4104.0s  11  Complete. Exited with code 0.

```

Comments (0)



Click here to enter a comment...