Whale Identification Challenge

By Keith Carlson, Alex Lockwood, Eric Keefe & Carlos Luevanos

Willamette University, CS 429, Intro to Data Science Spring 2018

Introduction:

We took up a Kaggle challenge to identify different species of whales based off their flukes. We were given two datasets of whale flukes: a training set with 9,621 images and a test set with 15,610 images, as well as a CSV file mapping each image name to a species of whale in the training set. Our task was to identify the species of whales present in the test folder as accurately as possible.

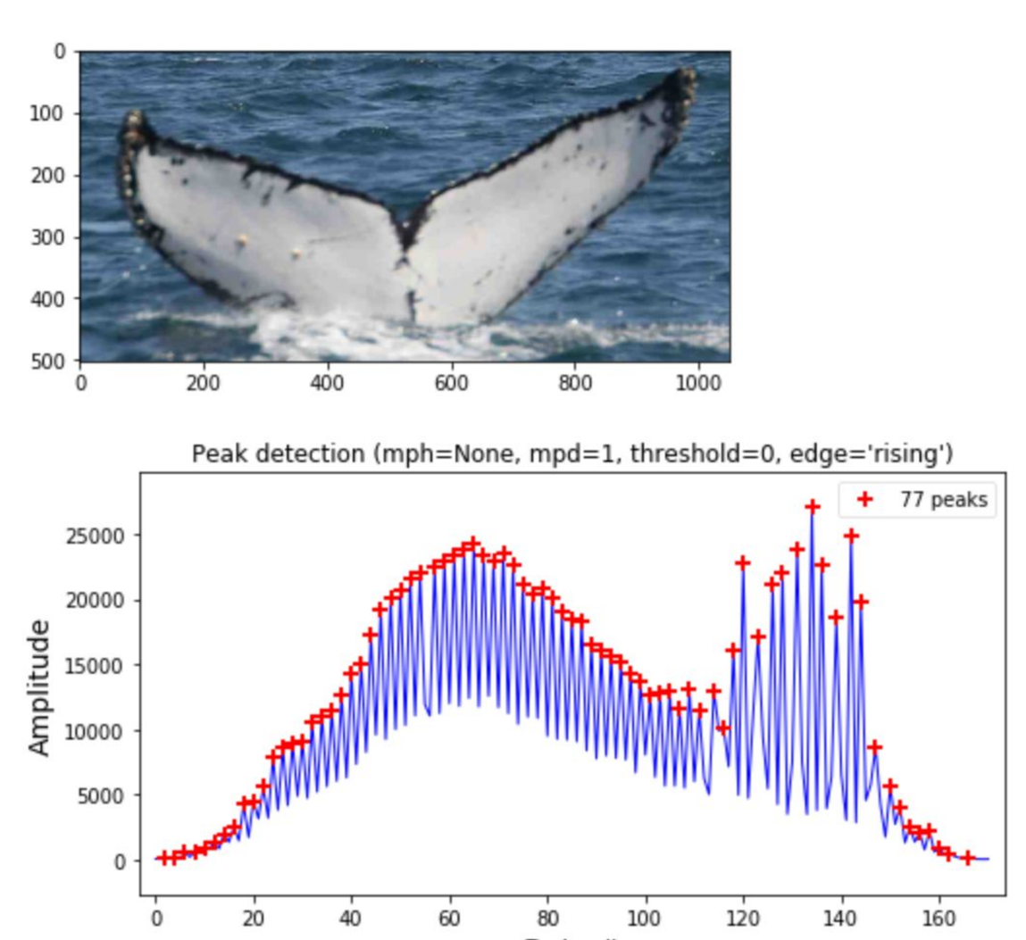
Results:

After excluding species with less than 15 images we were able to classify images in the test set with 12% accuracy, using a Keras classifier.

Exploration

- We read in the CSV file as a dataframe using Pandas
  - Further exploration of the data showed there were 4000+ unique whale species
  - Most of the classified whale species only had 1 to a handful of images
  - There was a class 'Id' known as 'new_whale" with 810 images but it was ambiguous so it had to be removed.
- We did some basic EDA by looking at pixel density and looking at peak detection.

	A	B
1	Image	Id
2	00022e1a.j	w_e15442c
3	000466c4.j	w_1287fbc
4	00087b01.j	w_da2efe0
5	001296d5.j	w_19e5482
6	0014cfdf.jp	w_f22f3e3
7	0025e8c2.j	w_8b1ca89
8	0026a8ab.j	w_eaad6a8
9	0031c258.j	new_whale
10	0035632e.j	w_3d0bc7a
11	0037e7d3.j	w_50db782
12	00389cd7.j	w_2863d51
13	0042dcc4.j	w_6dc7db6
14	0042ea34.j	w_968f2ca
15	00467ae9.j	w_fd1cb9d
16	004a97f3.j	w_60759c2
17	004c5fb9.j	w_ab6bb0a
18	005c57e7.j	w_79b42cd
19	006d0aaf.j	w_c9ba30c
20	0078af23.j	w_e6ec8ee

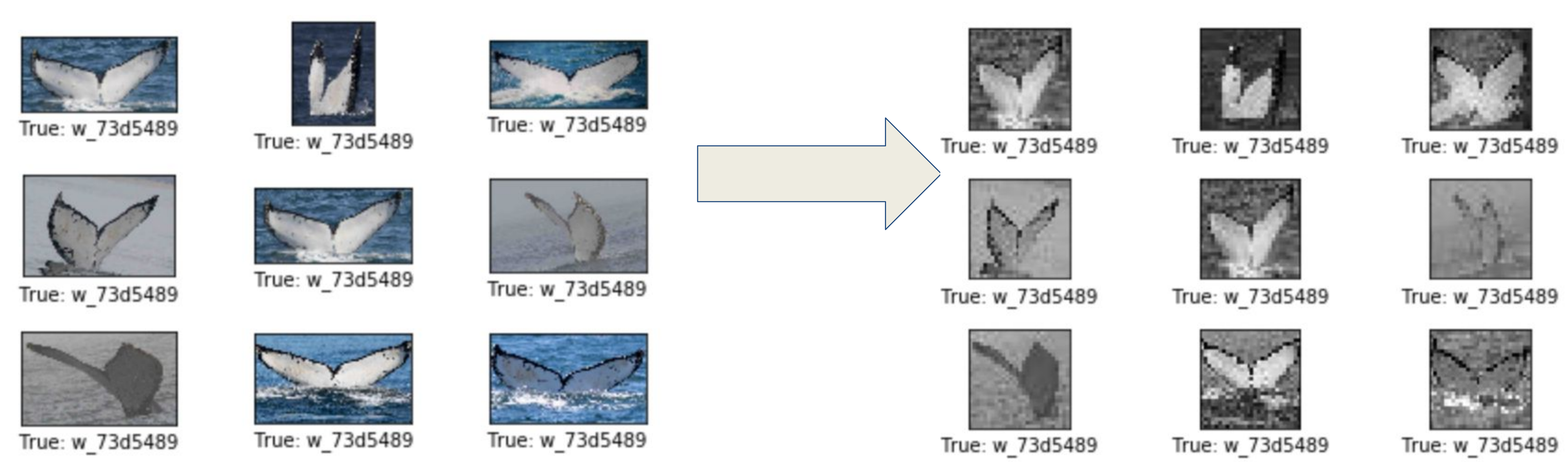


Keras and TensorFlow

- Keras is a user-friendly neural-network library that allows for easy, rapid deployment of neural networks. Keras runs a TensorFlow backend
- TensorFlow is a machine learning library developed by Google with the main focus of deep learning research and applications
- The design of TensorFlow allows for parallel processing on multiple CPUs, and GPUs, significantly reducing the time of computation for various problems.

Preprocessing

- Converted all the images into the same size
- Converted all images to grayscale for a uniform color
- Removed most of the whales due to lack of images for training



Keras Classifier

- We used Keras to create an image classifier on our whale dataset.
- We removed the new whale species and any species with less than 15 whales per species to improve the accuracy of our classifier
- Used a sigmoid classifier with 14 units.
- Loss was categorical cross entropy and class mode was categorical
- After playing with parameters our training accuracy ranged from 6%-12%
- Identification of images in the test set using this classifier was poor

Tensorflow Classifier

- We tried to train a TensorFlow model by following the "Hello World" for Tensorflow tutorial and trying to plug it into our images. This tutorial was from the Tensorflow.org website.
  - Took the top top 10 whale species containing the most images.
  - Had to convert images to ubyte format (thanks to gskielian on github for the code to do this.
- We got it to print saying it was classifying with 80% accuracy. The issue was getting this to deploy properly. As seen below it classifies every image with a zero. We are not sure if used out model right. Apparently actually deploying the model is a whole other separate process.
- We are also suspicious of the 80%, it might have to do with the fact that we divided out data

```
print((sess.run(accuracy, feed_dict = {x: test_images, y_: test_labels})) * 100)
80.0000011920929
```

```
print(sess.run(tf.argmax(y, 1), feed_dict = {x:train_images}))
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

Achievements and Conclusions

- Yielding a higher accuracy would require a lot more data
  - With so little images available for each whale class, 12% is not substantial but a good start
- The learning process of understanding TensorFlow, Keras, and convolutional neural networks was challenging
  - While we weren't able to complete the exact specifications of the Kaggle challenge, we applied what we learned from this class to put forward a result despite many difficulties.
- Deploying a CNN is totally different from training it. While we were able to make progress training our network, deploying it would take more time than we had.

Future Work

- Reach higher accuracy for the top whales (90% or more)
- Reach an overall higher accuracy for all whales (50% or more)
- Better pre-processing of images (Removing water, keeping color in images) .
- Learning more about how to properly use and deploy TensorFlow and Keras classifiers. This way we could see if our TensorFlow Classifier was actually 80% accurate.

Acknowledgements and References

- We would like to thank Google for their free, open access of the TensorFlow library.
- We acknowledge Kaggle for providing a large and clean set of whale fluke images.
- We acknowledge Francois Chollet for his devotion to making the Keras Library.
- We would like to thank Professor Cheng for her support and encouragement of this daunting task of image classification.
- Thanks to Magnus Erik Hvass Pedersen for code that helped in plotting images and he provided a great tutorial on training a Tensorflow classifier.