Saving the whales with image processing

Youssef Ahres
School of Engineering
Stanford University
yahres@stanford.edu

Juhana Kangaspunta
School of Engineering
Stanford University
juhana@stanford.edu

I. Introduction

At the moment, only 500 North Atlantic right whales are left in the world. To ensure the survival of this endangered species, marine biologists are tracking all of them to know their status and health at all times. However, manual recognition is tricky and very few researchers can perform it on the fly [1].

In this project, we implement a whale recognition system that allows researchers to reliably identify the individual whales from aerial photographs. This project is intended to be an entry to the Kaggle right whale recognition competition for which data set of labeled aerial photographs of North Atlantic right whales is provided.

Throughout this paper, we will propose a whale recognition system based on two main steps: head detection and recognition. As part of this pipeline, a novel image segmentation-based filtering method is applied. We evaluate our system and compare it to the top submissions on Kaggle discussing further improvements towards the end.

II. Data

The Kaggle dataset consists of 11468 aerial images of North Atlantic Right Whales. Every image contains only one individual whale heading towards an arbitrary direction. Figure 1 shows an example image from the training data.

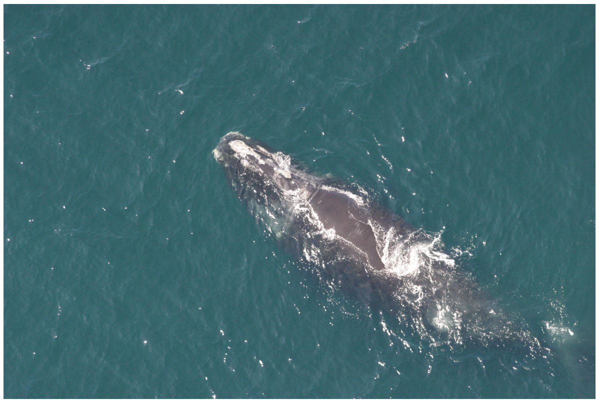

Fig. 1. Example image

Overall, there are 447 individual whales that we would like to recognize with high confidence. To do so, we have 4545 labeled images, that is, images showing a single whale associated with its whale identification number. The rest of the data set is not labeled. The first challenge with this data set relates to the number of labeled images we have for a given whale. Figure 2 shows the distribution of the labeled images among the individual whales.

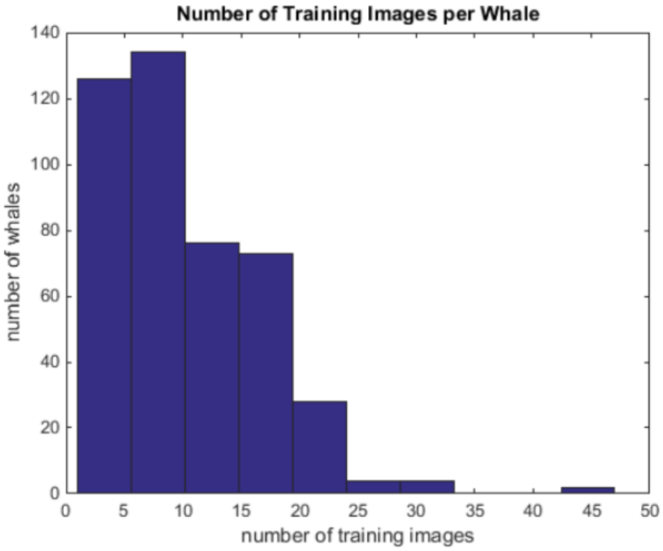

Fig. 2. Histogram of whales images

We see that most of the whales we are trying to recognize have between 1 and 10 images. This is a big limitation because the whales can have arbitrary directions and it is very hard to recognize a whale heading south based on a reference image of the same whale heading east. The head is often largely occluded hindering our capability to match keypoints or other recognition features.

Finally, as we will describe in details in the next section, grayscaling these images to perform classic object detection performs very poorly. This is due to the similar grayscale intensity between the whale and the water as well as glare in the water appearing very similar to the unique identifying callosity patterns of the whale.

III. Methodology

We approached this problem using a 2-step algorithm. First, we detect the head of the whale in the query image to reduce the dimensions and the probability of false matches due to the water and other sources of noise. Then, we use these isolated boxes to match the query image to the reference boxes. In the rest of this section, we focus on the methodology of each of these two steps.

A. Head Detection

Detection using cascade classifiers: To accurately detect a given object on an image, three widely used techniques seemed prominent in the literature: the Viola-Jones Haar-feature based cascade classifier[2], [13], [14] used in face detection, Histogram of Oriented Gradients (HOG) based methods [5] used in detecting humans and deep learning approaches [6], [10]. Convolutional neural networks are generally considered state-of-the-art [20]. However, they require a very large labeled data set. Lacking a simple to use implementation of a HOG detector and given the limited amount of data we had, we decided to work with the cascade classifier. In order to train this classifier, we labeled 1000 images by noting the bounding box around the whales head as positive examples and established a database of negative examples based on a random sample of water-only boxes and parts of the whales that are not intersecting the head. We also labeled the direction towards which the whale was heading. The direction is important because the cascade classifier is not rotationally invariant and is more effective when trained and optimized for a single direction using cross validation. Therefore, every query image is run through four cascade classifiers, one for each direction. This approach helped us detecting good bounding boxes for most of the images, as we will discuss in the experimental part.

Filtering false positives using image segmentation: Running every query image through 4 Haar cascade classifiers increased the rate of false positives, most of which happen on the water due to various patterns of the waves that sometimes match the heads characteristics. In addition, the grayscale preprocessing step by the Haar classifier further increases the likelihood of a false positive match. Figure 3 shows an example image after grayscaling. From this image, we can see that grayscaling makes it hard to see the whale and that could easily hinder the performance of any object recognition system.

Identifying the causes of false positive matches allowed us to develop an image processing pipeline to filter them out. The basic idea is to find the shape whale using image segmentation and apply it as a mask to ignore all the boxes that are not found to be on the whales body. To build such a system, we start by using K-means for image segmentation [7], [18], [19]. We attempted both RGB and HSV color spaces achieving a better separation on the HSV especially in the saturation dimension. Once the clusters are found, we can confidently assume that the largest cluster consists of the water pixels, which we set to zero, creating a noisy mask of the whale as shown below. In order to clean out the noise and have a single unified whale shape in the mask, we use morphological transformation followed by object labeling. More precisely, we erode the image using a 5x5 square structured element to isolate the noise out the whale. Then, we labeled the connected components on the image and sorted them by size. The largest was considered as the whales body. This is a fundamental assumption, we are making in this approach. We will discuss its results and limitation in the experimental part of the paper. The result of this step is shown in Figure 5. Based on the identified whale shape, we can filter the boxes that are not overlapping enough with the mask. Formally, we define the overlap as

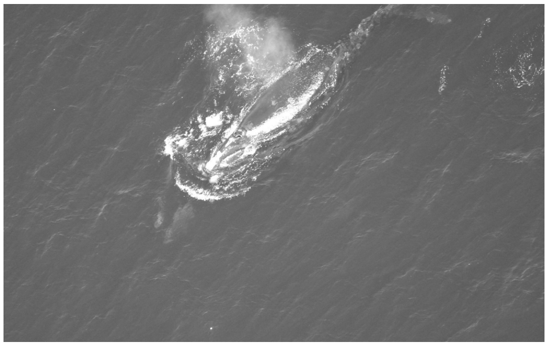

Fig. 3. Grayscale example image

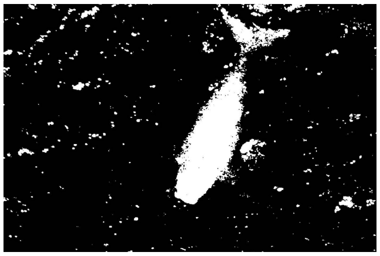

Fig. 4. Mask after image segmentation using K-Means

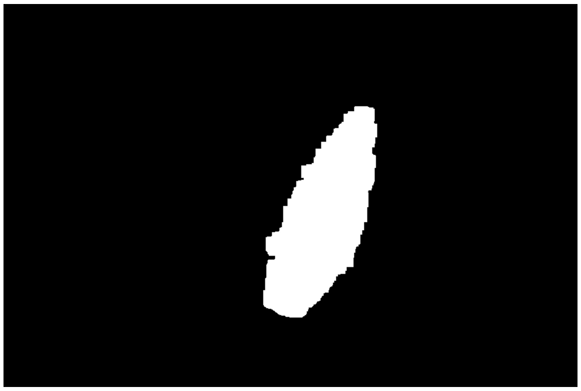

Fig. 5. Mask after post processing

$$OVERLAP = size(b \cap mask)/size(b) \tag{1}$$

Where b is the box and mask is the extracted mask.

We compute the defined overlap measure for every box keeping the ones that have an overlap score above 0.45. This threshold was optimized empirically to maximize our filtering operation without hindering the recall.

Summary: The pipeline in Figure 6 summarizes the methodology used to detect the whales head using a real example: a- Detection step using Haar features cascade classifiers b1- Image segmentation using K-Means in HSV color space b2- Erosion and connected element labeling to identify the whale c- Use the identified whale mask to filter false positive boxes

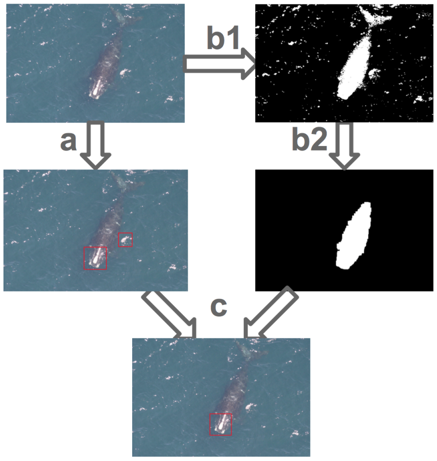

Fig. 6. Pipeline

B. Head Recognition

The goal of the project was to assign each unlabeled aerial image of a whale to a specific individual that is displayed in one or more images in the training set. Since the whales can appear in the image in various rotations, scales and with varying degrees of occlusion by the water, we needed a method that was rotationally invariant, invariant to scaling as well as robust against occlusion. Due to limited training data (approximately 10 training images per whale) we concluded that deep neural networks, especially without transfer learning, were difficult to train without overfitting. When researchers identify right whales manually, they use white callosity patterns on the whales head as the main method [11]. However, due to significant glare from the water, developing a method to extract these callosity pattern features reliably was deemed both outside the scope of this class and most likely inaccurate. Finally, we decided to use rotationally and scale invariant feature descriptors that were presented in class, namely SIFT [8], [16] and SURF [9], [17]. Initially, we experimented with SIFT, but decided to use SURF because of the faster computation time and the large number of images we would have to match against for each unlabeled image.

The steps of the whale matching algorithm are as follows:

1) Detect SURF keypoints for each of the whale head bounding boxes in the training set and save the feature descriptors with the corresponding whale identifier label
2) For each unlabeled image detect SURF keypoints and match the feature descriptors against all saved training set feature descriptors and compute the number of matches between the query image and the training set image. For each bounding box of the head, pick the one with the highest number of matches
3) For each unlabeled image, each individual whale is assigned a probability based on the number of matched keypoints.

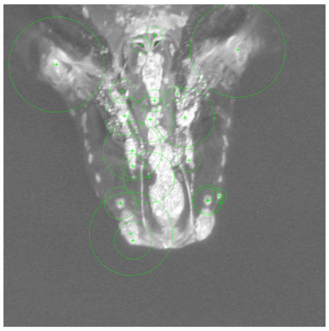

Fig. 7. Whale SURF features

Finally each unlabeled image will have a vector v of probabilities so that each element v_i in the vector represents the probability of the image representing whale i. Finally to optimize the hyperparameters of the SURF detection and matching, we used cross validation on a holdout set.

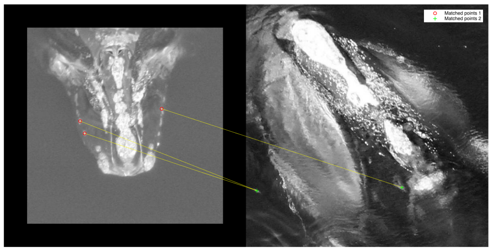

Fig. 8. Different whale SURF feature matches

IV. Experimental Results

In this section, we discuss the experimental setup of this algorithm and its implementation. Then, for each of the steps described above, we establish a success metric and discuss its performance.

A. Head Detection

This step was implemented using Matlab Image Processing toolbox and OpenCV [12] on Python. We train the Haar cascade classifiers using OpenCV (step a) and perform the image

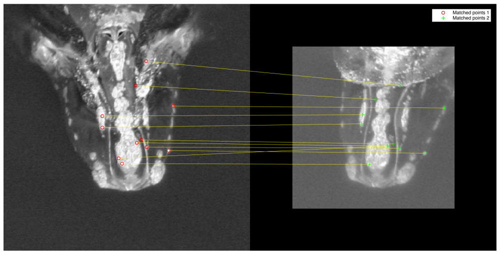

Fig. 9. Same whale SURF feature matches

segmentation (step b) using Matlab. We split the data randomly into 700 training examples and 300 testing examples.

Initially, we would like to estimate the effectiveness of the filtering step. First, we determine the probability of a correctly extracted mask obtained by image segmentation and postprocessing. To do so, we compute the overlap of the ground truth bounding box and the mask following equation[1] and check whether it is above the defined threshold. If so, we consider the mask as being valid, otherwise it is not valid. Using this approach, we compute the probability of a correctly found mask and find:

$$p = 0.875 \tag{2}$$

Intuitively, this means that in 87.5% of the images, the approach using the K-means image segmentation and morphological transformation successfully finds the whale in the image.

We also establish an evaluation metric related to the mask filtering and how many false positive boxes this approach manages to filter. To do so, we compute the number of predicted boxes on the testing set, before and after the filtering. Running this on our training set, we see that, on average, before filtering the pipeline outputs 26.27 boxes per image and after filtering 9.27 boxes per image. It is reasonable since most of these boxes at least partially match the whales head. We take advantage of these partially matched boxes during the prediction step to boost our performance.

Finally, to evaluate the head detection overall performance, we define the following metric:

$$bbp = max_{(b \in boxes)} size(b \cap b_{gt}) / size(b \cup b_{gt}) \tag{3}$$

Where bbp stands for best box precision and b_{gt} is the ground truth bounding box manually labeled.

The intuition behind such a metric is that we use various heuristic techniques in the recognition step to take advantage of this number of boxes and aggregate them.

On our testing set, we obtain:

$$bbp = 0.78 \tag{4}$$

B. Head Recognition

The Kaggle competition submissions are evaluated using multi-class logarithmic loss which we also use to evaluate the success of our algorithm:

$$logloss = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log(p_{ij}) \tag{5}$$

In the equation N is the number of images in the test set, M is the number of whale labels, $\log$ is the natural logarithm and $y_{ij} = 1$ if observation i belongs to whale j and 0 otherwise. p_{ij} is the predicted probability that observation i belongs to whale j [1].

On an isolated test set, our algorithm achieves a logarithmic loss of 5.55438 which would place us in the top 30 out of 207 teams in the Kaggle competition. Unfortunately, due to time constraints and execution time challenges we were unable to run on the full Kaggle dataset and make a final submission. To put this number into perspective the Kaggle sample submission benchmark achieves a log loss of 34.48176 and the top submission achieves a log loss of 3.07203.

V. Conclusion

In this paper, we presented a methodology and an implementation for detecting and recognizing individual right whales from aerial photographs. For whale detection, the proposed image segmentation through k-means clustering in the HSV colorspace proved to be effective in filtering the high number of false positive bounding boxes arising from the Viola-Jones [2] cascade classifier. High amount of water reflection and occlusion of the whale made detection challenging but the detected bounding boxes were of sufficient precision and recall to be used in further recognition.

For recognizing the right whales from the obtained bounding boxes, we proposed an approach using SURF keypoint detection and matching SURF features between the unlabeled image and training set images. Based on our initial results, this method appears to be effective in comparison to the large majority of other Kaggle competition submissions yielding a score that would earn a top 30 standing out of 207 teams.

References

[1] https://www.kaggle.com/c/noaa-right-whale-recognition

[2] Viola P., Jones M. Rapid object detection using a boosted cascade of simple features Computer Vision and Pattern Recognition (2001)

[3] Arriaga-Gomez, M.I.F., de Mendizabal-Vazquez I.; Ros-Gomez R.; Sanchez-Avila C. A comparative survey on supervised classifiers for face recognition Security Technology (ICCST), 2014 International Carnahan Conference on (2014)

[4] Lu Wang; Yung, N.H.C. "Extraction of Moving Objects From Their Background Based on Multiple Adaptive Thresholds and Boundary Evaluation", Intelligent Transportation Systems, IEEE Transactions on, On page(s): 40 - 51 Volume: 11, Issue: 1, March 2010

[5] Dalal N., Triggs B. Histograms of oriented gradients for human detection Computer Vision and Pattern Recognition (2005)

[6] Szegedy C., Toshev A., Dumitru E. Deep Neural Networks for Object Detection Advances in Neural Information Processing Systems 26 (NIPS 2013)

[7] Tse-Wei Chen, Yi-Ling Chen, Shao-Yi Chien Fast image segmentation based on K-Means clustering with histograms in HSV color space 2008 IEEE 10th Workshop on Multimedia Signal Processing

[8] Lowe D.G. Object recognition from local scale-invariant features The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999

[9] Bay, Herbert, et al. ”Speeded-up robust features (SURF).” Computer vision and image understanding 110.3 (2008): 346-359.

[10] Girshick, Ross, et al. ”Rich feature hierarchies for accurate object detection and semantic

[11] New England Aquarium Right Whale Callosity Pattern Identification

[12] Learning OpenCV: Computer vision with the OpenCV library

[13] Papageorgiou, Oren and Poggio, ”A general framework for object detection”, International Conference on Computer Vision, 1998

[14] Lienhart, R. and Maydt, J., ”An extended set of Haar-like features for rapid object detection”, ICIP02, pp. I: 900903, 2002

[15] Messom, C.H. and Barczak, A.L.C., ”Fast and Efficient Rotated Haar-like Features Using Rotated Integral Images”, Australian Conference on Robotics and Automation (ACRA2006), pp. 16, 2006

[16] Lazebnik, S., Schmid, C., and Ponce, J., Semi-Local Affine Parts for Object Recognition, BMVC, 2004

[17] P. M. Panchal, S. R. Panchal, S. K. Shah, ”A Comparison of SIFT and SURF ”, International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013

[18] Arbelaez, Pablo, et al. ”Contour detection and hierarchical image segmentation.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 33.5 (2011): 898-916.

[19] Jain, Anil K. ”Data clustering: 50 years beyond K-means.” Pattern recognition letters 31.8 (2010): 651-666.

[20] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.