Math. Appl. **5** (2016), 155–170
DOI: 10.13164/ma.2016.11

NORTH ATLANTIC RIGHT WHALE LOCALIZATION AND RECOGNITION USING VERY DEEP AND LEAKY NEURAL NETWORK

ABDULWAHAB KABANI AND MAHMOUD R. EL-SAKKA

Abstract. We describe a deep learning model that can be used to recognize individual right whales in aerial images. We developed our model using a data set provided by the National Oceanic and Atmospheric Administration. The main challenge we faced when working on this data set is that the size of the training set is very small (4,544 images) with some classes having only 1 image. While this data set is by far the largest of its kind, it is very difficult to train a deep neural network with such a small data set. However, we were able to overcome this challenge by dividing this problem into smaller tasks and by reducing the viewpoint variance in the data set. First, we localize the body and the head of the whale using deep learning. Then, we align the whale and normalize it with respect to rotation. Finally, a network is used to recognize the whale by analyzing its callosities. The top-1 accuracy of the model is 69.7% and the top-5 accuracy is 85%. The solution we describe in this paper was ranked 5^{th} (out of 364 teams) in a challenge to solve this problem.

1. Introduction

The North Atlantic right whales [6] is an endangered species with around 450 whales left. Historically, right whales have been subject to harsh hunting since the 17^{th} century. Some researchers believe that the name 'right whale' comes from the fact that this is the 'right' type of whale for hunting. These whales were considered to be ideal for hunting for many reasons such as their tendency to live close to the shore, being rich in whale oil, and the fact that their bodies float when killed. Despite becoming protected species in 1949, the population is still endangered with being entangled in fishing gear, and collision with ships accounting for around 50% of deaths.

Right whales can be recognized by studying the callosity pattern on their heads. Manually classifying whales is a very time consuming process and automating this

MSC (2010): primary 82C32; secondary 68T45.

Keywords: whale localization, whale detection, whale recognition, deep learning, convolutional neural network, localization, detection, recognition, image classification.

This research is partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC). This support is greatly appreciated. We would also like to thank Kaggle and the National Oceanic Atmospheric Administration Fisheries for providing this data set. Also, we would like to thank Nervana Systems for sharing their approach and annotating the bonnet and the blow hole locations. Their contribution helped us complete this research.

process can help scientists focus on their conservation efforts. We used a unique data set provided by the National Oceanic and Atmospheric Administration [6,17] to develop a model that can automatically recognize individual whales by analyzing the callosity pattern on their heads. The solution we describe in this paper was ranked 5^{th} (out of 364 teams) in a Kaggle challenge [17] to solve this problem.

When working on this problem, we faced several challenges. First, the size of the training set is very small while the number of classes (individual whales) is large. Second, there is a huge variation in the clarity of each image. Finally, the size of each image is very large making it very difficult to load these images into the GPU.

To overcome these challenges, we first localize and normalize the body with respect to rotation. After that, we localize the head of the whale. Finally, the whale is recognized using the callosity pattern on its head. These steps reduce the image size and ensure that we can fit the data into the GPU.

In this paper, we will start by talking about deep learning in Section 2. We will describe the approaches taken by other teams in Section 3. General overview and information about the data that we used will be presented in Section 4. Sections 5 and 6 present the methods we used to localize the head of the whale. In Section 7, we present the model that we used to recognize the whales. The results are introduced in Section 8. Finally, we conclude our work in Section 9.

2. Deep learning background

Deep Learning involves training models that consist of several layers of abstraction. In other words, each layer builds on the abstraction in the previous layer by applying non-linear transformation (or a linear transformation, if desired). Training the network requires optimizing the parameters of all the layers in the network. This can be done by optimizing an objective function (error or loss function). Using stochastic gradient descent, the objective function is optimized by updating the parameters at each layer by taking small steps.

A Convolutional Neural Network (convnet or CNN) [14] is a special type of neural network that contains some layers with restricted connectivity. This restriction results in a behavior similar to convolution in signal processing.In convolution, an image is convolved with a kernel of certain size and weights. The restriction of connectivity in convolutional layers produces a similar behavior with the exception that the kernel weights are learned during training rather than defined by user in most signal processing applications.

CNNs performed really well in the problem of digit recognition where they achieved state of the art performance on the famous MNIST data set [26]. In general object recognition problems, CNNs achieved excellent results in many competitions such as the ImageNet large scale classification challenge [13, 21]. Several models [1, 7, 16] achieved excellent results in this competition. The success was possible thanks to the development in computing power, regularization techniques such as Dropout [11, 19], initialization methods [25], ReLU activations [24], and data augmentation.

3. Related work

The National Oceanic and Atmospheric Administration [6] ran a competition on Kaggle [17] to automate the recognition of right whales from aerial survey images. Each right whale has a unique callosity pattern on its head. These callosity pattern can be used to identify right whales just like a fingerprint pattern can be used to identify humans. The number of training images is very low (4,544 images) while the number of testing images is 6,925 images. Deep learning has been very successful in recent years with visual recognition problems. However, it was very difficult to make it work for this type of data. This is mainly because the size of the data is very small. Anil Thomas from Nervana Systems [2] suggested locating the bonnet and the blow hole on the whale. The solution which is based on a deep learning library called Neon [20] used these two points, extracting a patch that contains the whale's head. This approach proved to be very effective. First, it made the training process much faster because the training is being done on smaller images. Second, these head patches are better than the original images for training a deep learning model. This is mainly because training on these head patches made the model focus only on the most discriminative features (the head callosities) and ignore unimportant features such as features from the surrounding water.

To our knowledge, most of the solutions (including the solution described in this paper) that ranked at the top on the competition's leaderboard followed this idea that was proposed in [2]. The team that was ranked in the second position [9] used a similar multi-stage approach. First, the head is localized by regressing a bounding box. Then, the head is aligned in an approach that is loosely inspired by a human face alignment approach that was introduced [27]. This team used a classifier that is an ensemble of deep neural networks with different variations of the VGG-Net [16] and ResNet [15].

The DeepSense.io team [22] that ranked first also used a multi-stage approach. This team produced a solution [22] that involves localizing the head of the whale and aligning it afterwards. They report that aligning the head of the whale is very important to achieve good results. Their final solution is based on combining the predictions of different deep learning models.

4. Method overview

An overview of our method is presented in Figure 1. While the main task is to recognize the individual whales IDs, we could not do that directly on the original images. This is because the original images are huge and fitting them in the GPU during recognition is not feasible.

We divided the problem into smaller tasks (as shown in Figure 1). First, two models are trained to localize the bonnet and the blow hole, respectively. Then, using these points, the body of the whale is localized and rotated so that the body has angle=0 with the x axis and the head is pointing east. After that, a model is trained to localize the head of the whale. Finally, a model is trained to produce a probability distribution over all possible whales.

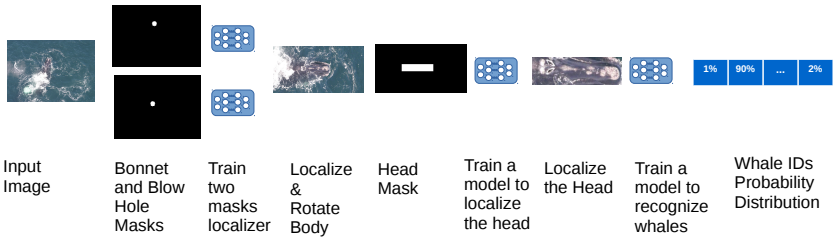

Figure 1. The overview of the method: First, two models are trained to localize the bonnet and the blow hole, respectively. Then, using these points, the body of the whale is localized and rotated so that the body has angle=0 with the x axis and the head is pointing east. After that, a model is trained to localize the head of the whale. Finally, a model is trained to produce a probability distribution over all possible whales.

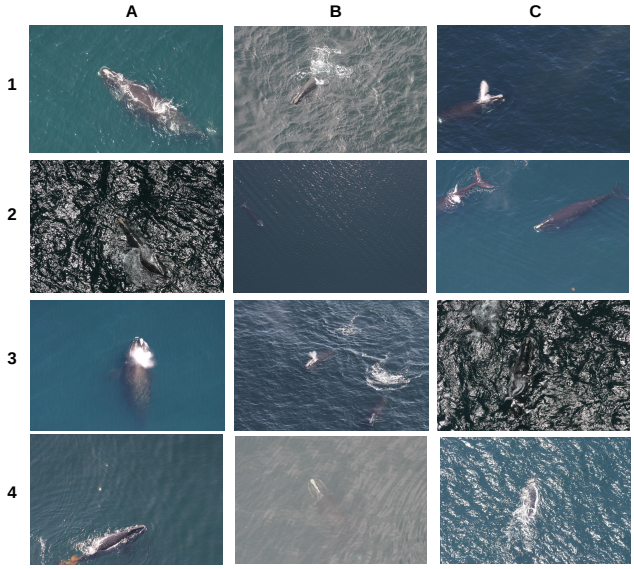

Figure 2. A random sample of images: These images were captured during several aerial surveys and under different lighting and environmental conditions (note images A1, B2, B4, C4). In addition to these variations, there are many other obstacles, which make these images very challenging. For instance, for many whales the head is not clear because the whale is blowing water as in images C1 and C3. Some images contain more than two whales as in images C2 and B3. In images A2 , C3, and B4, the foreground/background contrast is very low. Water reflection can make recognition very difficult as in image C4. The size of the whale with respect to the background (such as the one in image B2) is another challenge.

5. Body localization

In this section, we describe how we trained a model to localize the body of the whale and normalize it with respect to rotation. The trained model will be able to take the original image as input and produce an output where the image is

cropped and the whale is facing east. This is very important in order to train the classification model (presented in Section 7). Figure 2 shows a random sample of the input images that we pass to the model in order to localize the whale body.

This is a very important step for many reasons. First, localizing the body and the head (in Section 6) helps the model focus on the important features on the whale body rather than on features in the surrounding water. Because there are many classes with very few training images, focusing on the important features in the image (the callosities on the top of the head) is essential to alleviate overfitting.

Second, rotating the body so that the angle between the body and the x axis is 0 is important because it helps in extracting head crops with minimum amount of surrounding water. Otherwise, if the whale body has different orientation, extracting head crops may include a lot of surrounding water. For example, Figure 3 shows two head crops, one crop is for a whale that has 0 degrees angle with the x axis and another with around 135 degrees angle. It is clear that the one in the latter crop contains far more water.

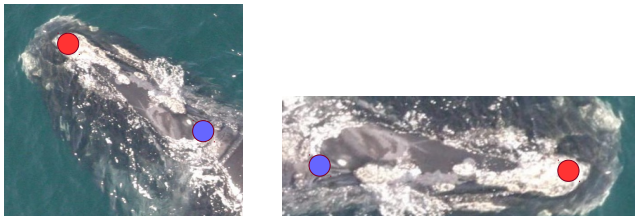

Figure 3. Comparison between two head crops. The one on the left is taken from an image where the angle of the body of the whale with the x axis is 135 degrees. On the other hand, the one on the right is the result of normalizing the angle of the body so that it is 0 degrees. The figure also shows the locations of the bonnet and the blow hole. Two localizations models are trained to recognize the bonnet and the blow hole, respectively. Then, using these two points, we can calculate the angle with the x axis and rotate the whale accordingly.

To be a able to localize the body, we train two models: one will be used to recognize the bonnet (shown in Figure 3 as a red point) and the other is used to recognize the blow hole (shown in Figure 3 as a blue point). Using these two points, we can easily calculate the angle between the body and the x axis and rotate the body accordingly. This simple and powerful idea was originally introduced in [2] and we found it to be very helpful. In [2], two points are used to train two convolutional autoencoders. After that, the head was extracted and rotated. We use a similar approach. However, rather than training a convolutional autoencoder, we trained a deep neural network with the units in the output layer corresponding to individual pixels in the masked image. Figure 4 shows a summary of the body localization stage.

Each mask is used to train a deep neural network. The mask and the image are re-sized to size 128 ($height$) $\times$ 192 ($width$).The labels (ground truth) for this network are the elements-wise multiplication of the resized mask with the gray scale image.

To train a neural network to localize the interest point (bonnet or blow hole), the re-sized original image is used as input and the predicted output of the network

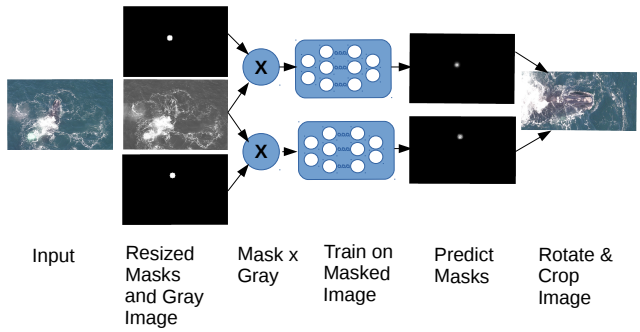

Figure 4. Body Localization Overview: In order to locate the body, the bonnet and blow hole masks are re-sized to 128×192. Also, the original image is re-sized to 128×192 and converted to gray scale. Each of the two masks is multiplied with the gray scale image to produce masked images. Each of these masked images are passed into two networks to train a network to predict the location of the interest point (bonnet or blow hole). The network is used to predict the location of the interest points. These interest points are used to rotate the whale and localize the body.

is the masked image. The architecture of the network we used for training is shown in Figure 5.

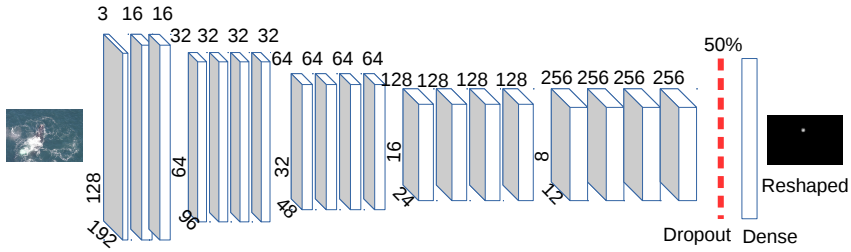

Figure 5. Localization architecture: the same architecture is used to localize the bonnet and blow hole, and later the head (in Section 6). The input of the architecture is an image of size 128×192. The output of the network is a layer with $128 \times 192 = 24576$ possible classes. The output layer is simply a flattened mask and reshaping this layer gives us back the predict mask. The pixels with the highest intensities represent the location of the interest point.

The loss function we minimize is the categorical cross-entropy (or mutli-class logloss). The learning rate was initially set at 0.01 and reduced automatically if the validation loss does not improve after 10 epochs. The activation for each convolutional layer is ReLU [24] while the activation for the output layer is softmax. Softmax activation ensures that each pixel in the predicted mask is in the range of (0,1) and the pixels sum up to 1. In other words, it is a measure of certainty that a certain pixel in the predicted mask corresponds to the location of the interest point. The dropout rate is 50% and the max-pooling is done over size (2,2).

Because the last layer in the network is softmax, pixels in the predicted mask are probabilities that range between 0 and 1 and that sum up to 1. It is very

important to ensure that the ground truth (the labels) follow the same rules. In order to do that, we rescale the true mask (or the true labels) by dividing each pixel by the sum of the mask as shown in Equation (5.1):

$$y_{ij} = \frac{pixel_{ij}}{\sum_{i=1}^{H}\sum_{j=1}^{W} pixel_{ij}} \tag{5.1}$$

where $pixel_{ij}$ is the pixel value at row i and column j. y_{ij} is the normalized pixel value such that $y_{ij} \in [0, 1]$ and $\sum_{i=1}^{H}\sum_{j=1}^{W} y_{ij} = 1$.

Once the two models that predict the bonnet and the blow hole are trained, they can be used to predict the locations of bonnet and the blow hole. For each mask, the pixel with the highest intensity is considered to be the one where the model predicts the interest point to be. To make the prediction more robust, we take the mean location of the top 5 pixels with the highest intensities.

To localize the body and rotate it, we first enlarge the mask from the size 128×192 to the original size. The top 5 pixels with the highest intensities are averaged for each of the two masks. Then, the angle of the whale with respect to the x axis is estimated by Equation (5.2).

$$\theta = \tan^{-1}\left(\frac{y_{bonnet} - y_{blowHole}}{x_{bonnet} - x_{blowHole}}\right), \tag{5.2}$$

where y_{bonnet} and x_{bonnet} are the predicted coordinates of the bonnet on the y and x axes. As discussed earlier, this point is the result of averaging the coordinates of the top 5 pixels with the highest intensities in the bonnet predicted mask. The same thing applies to the $y_{blowHole}$ and $x_{blowHole}$ which are the predicted coordinates of the blow hole.

The image is rotated around the estimated center of the whale head, which is given by the equation:

$$x_{headCenter} = 0.5 \times (x_{bonnet} - x_{blowHole})$$
$$y_{headCenter} = 0.5 \times (y_{bonnet} - y_{blowHole})$$

The image is cropped from the center of the head and the size of the crop is $4 \times distance$ along the x axis and $2 \times distance$ along the y axis. The *distance* value is the distance between the blow hole and the bonnet. Figure 10 in Section 8 shows a sample of images produced using the information we described in this section. As shown in Figure 10, the resulting images all show the whale bodies localized and pointing in the same direction.

6. Head localization

Once the body is localized and rotated so that it is pointing east, we are ready to localize the head of the whale. A head mask is used to train a network to localize the head. The mask and the input image (the whale body image we produced in Section 5) are re-sized to size 112 (*height*) $\times$ 224 (*width*). The labels (ground truth) for this network are created by multiplying (element-wise) the re-sized mask with the gray scale image.

The architecture of the network we used for training is the same one we used in the previous section (shown in Figure 5). The only difference is that the body size

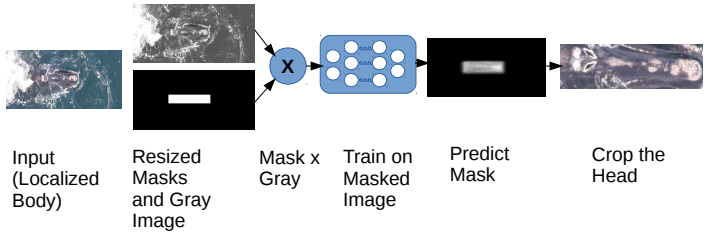

Figure 6. Head Localization Overview: In order to locate the head, the head mask is re-sized to 112×224. Also, the body image is re-sized to 112×224 and converted to gray scale. The mask is multiplied with the gray scale image to produce masked images. The masked image is passed into a network to train it to predict the location of the head. This network is used to predict the location of the head.

of the input image and the mask is different. Therefore, the number of parameters at each layer is different.

As shown in Figure 6, the whale body image is converted into gray scale. The head mask and the gray scale images are multiplied and passed to the model for training. Then, the model is trained to predict the head mask. Finally, the predicted mask is re-sized to have the same size as the whale body image. Then, the predicted mask is thresholded and converted from gray scale image into binary image. The coordinates of the largest rectangle in this binary image are used to crop the head from the body image.

We used multiple thresholding methods to convert the gray scale mask into binary mask. For instance, we thresholded the head masks using Otsu [18]. The image is thresholded as shown in Equation (6.1):

$$I(i,j)_{thresholded} = \begin{cases} 1, & \text{if } I(i,j) \geq threshold_{Otsu} \\ 0, & \text{otherwise,} \end{cases} \tag{6.1}$$

In addition, we use another method to threshold the head mask according to Equation (6.2):

$$I(i,j)_{thresholded} = \begin{cases} 1, & \text{if } I(i,j) \geq \mu_{Otsu} \\ 0, & \text{otherwise,} \end{cases} \tag{6.2}$$

where μ_{Otsu} is the mean of all pixels that are higher than the ostu threshold. In other words, Equation (6.2) produces smaller head crops than Equation (6.1). During each epoch while training the recognition model (in Section 7), we will train on random sample from head crops produced using the Otsu method and the high mean method. We find this to be an effective data augmentation tool to reduce overfitting.

The model extracted crops of the head where all heads are normalized with respect to orientation (east) and rotation (angle = 0), translation, and scale. Figure 11 (in Section 8) shows a sample of head crops produced using the information we introduced in this section. The callosity patterns shown in Figure 11 are unique to each individual whale and can be used to identify a whale. These head images are passed to the recognition model (presented in Section 7).

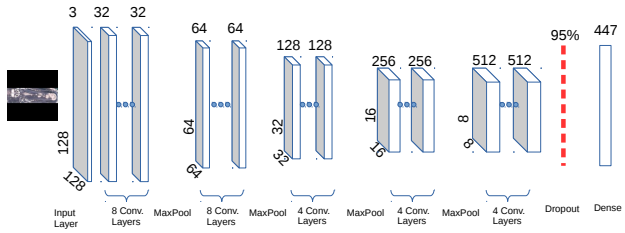

Figure 7. Recognition Architecture: the input to the network is an image showing the callosities of the whale. The output size is 447 corresponding to 447 unique whale IDs.

7. Recognition

Now that the head is extracted from the original image, it is time to train a model to recognize the individual whales. Right whales can be recognized by the callosities on the top of their heads. It is estimated that there are 450-500 north Atlantic whales remaining. However, the dataset only contains 447 individual whales. The network we trained can predict the ID of the whale by examining the callosities. This is very similar to the face recognition problem where the ID (or name) of the person is recognized by examining the facial features.

The network we used for training is described in Figure 7. Driven by the success of the VGG architecture [16], we opted for a similar architecture where the convolutional filter is small (3,3) and the network is very deep. The small convolutional filter helps in regularizing the network because each neuron is connected to a small number of neurons in the previous layer. We did not use any fully connected except for the output layer. Normally, the dropout rate is set at 50%. However, for this problem we set the dropout rate to a relatively high value which is 95%. We noticed that setting this value to lower than 75% leads to overfitting after only 10-20 epochs.

The image is padded by 1 pixel to ensure that the spatial resolution does not decrease except after the pooling layer. We used a (2,2) max pooling to sub-sample the response and detect more abstract features. The activation function we used is the leaky rectification (with leakiness=0.3) [3,4] which ensures that the gradient is not 0 for negative pre-activation. The activation in the output layer is softmax which ensures that the output produced is a probability (between 0 and 1, and all classes sum up to 1). All layers were initialized randomly.

It is worth mentioning that there is some variation in the aspect ratio of the head images. However, the network expects all input images to be of the same size (in our case, it is 256×256). In order to avoid distorting the callosities image when resizing the image, we pad the image so that the image size becomes $width \times width$.

Given the small size of the training set, it is essential to augment the data to alleviate overfitting. Table 1 shows the list of random augmentation we used along with their parameters.

The network was trained for 520 epochs. During each epoch, a random sample of training images is created from different sources. For instance, for each epoch

Table 1. Data augmentation: Random transformations along with parameters. These transformations are applied randomly to each image before sending it to the GPU.

Transformation	Parameters
Rotation	Angle between -20 and +20
Horizontal Flip	Randomness=50%
Vertical Flip	Randomness=50%
Horizontal Shift	Up to 12 pixels
Vertical Shift	Up to 12 pixels
Gaussian Blurring	Up to $\sigma = 1$
Contrast Rescaling	Randomly stretch/shrink intensity

we randomly choose head crops images which were created using the Otsu thresholding method. In addition, we combine them with randomly chosen head crops created using the high mean method (described in Section 6).

The learning rate was initially set at a very low value of 0.003. While training, the validation is continuously being evaluated. If the validation loss does not improve after 10 epochs, the learning rate is automatically reduced by 50%.

8. Implementation and results

In this section, we describe how the model was implemented and the results. The data is hosted on Kaggle [17]. The size of the data is very small with respect to the number of labels. The size of the training set is 4544 images. For the training set, both the images and the labels (individual whales IDs) are provided. On the other hand, for the testing set, only the images are provided without the labels and the size of this set is 6925 images. In order to perform validation locally, we extract a validation set from the original training set. The size of the validation set is 10% of the training set (452 images). Therefore, the training set size is reduced to 4092.

The number of whales in each individual whale varies significantly from whales with 1 training image up to 47 training images. Figure 8 shows a summary of the number of whales with a certain number of images. For instance, there are 24 whales (or classes) with only 1 training image and 29 whales with 2 training images. On the other hand, there is 1 whale with 47 training images. The average number of training images in each class is 10 training images.

We developed our model on a laptop equipped with GTX980M (4GB) graphics card. The code was developed using Theano [5, 12] which is a python library for optimizing and evaluating mathematical expressions in multidimensional arrays. We also used keras [8] which is a highly modular library to train neural networks on GPUs or CPUs.

In order to pre-process the image data and to perform geometric transformations, we used scikit-image [23] and openCV [10]. The networks were trained in batches of size 32 images. This is the largest batch size we could fit in the GPU memory. The CPU performs data augmentation on each batch before sending it to the GPU for training.

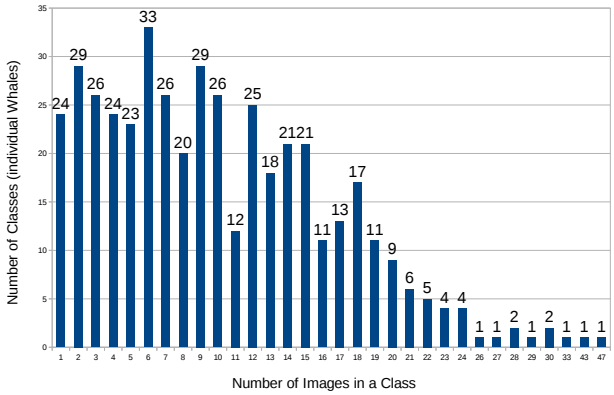

Figure 8. A summary of the number of whales with a certain number of images. There are 24 whales (or classes) with only 1 training image and 29 whales with 2 images. On the other end of the chart, we can see that there are few classes with a relatively high number of training images. For instance, there is one whale with 47 training images.

Training each of the three localization networks took around 9 hours. On the other hand, training the recognition network took around 50 hours. Therefore, the total training time for all the networks is $9 + 9 + 9 + 50 = 77$ hours.

The metric used by the server to evaluate the predictions is the multi-class logarithmic loss (also known as categorical cross-entropy). The equation for this metric is:

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

where N is the number of images in the predictions file (or the number of images in the test set), M is the number of labels (the total number of individual whales). y_{ij} is a mapping from the image i to the true label j (for example, y_{ij} is 1 if the image i belongs to the whale j and 0 if it does not). $log(p_{ij})$ is the natural log of the predicted probability made by the model that the image i belongs to whale j.

While training the recognition network, we minimize this metric directly. Figure 9 shows this loss function as it is being minimized during training. The log loss goes as low as 1.62 at the end of the training.

On the server, the log loss score we achieved is 1.47. The top-1 accuracy of the module on the validation set is 69.7% while the top-5 predictions accuracy is 85.0%.

We used the same loss function to train the localization modules. Figure 9 shows the train and validation log loss progress for both the bonnet localization network and the blow hole network, respectively. In Figure 9, we can see that the loss function of the bonnet localization network decreases from 4.5 to around 2.28 at the end of training.

As shown in Figure 9, the blow hole localization loss decreases from 5 to around 2.52 at the end of the training. The performance of the bonnet network is slightly better than the blow hole localization network as the former has a lower loss than

Table 2. Teams Ranking: this table shows the ranking of the solution we describe in the paper. The solution ranked in the 5^{th} position. The table only shows the top 10 teams while the number of teams that participated in the competition is 364. The full table can be found on the web page of the competition [17].

Ranking	Team Name	Score
1	deepsense.io	0.59600
2	felixlaumon	1.07585
3	SKE	1.14982
4	threedB	1.33648
5	AbdulWahab	1.46909
6	Tsakalis Kostas	1.51900
7	bawdyb .	1.55823
8	Left Whales	1.75764
9	Anil Thomas	1.80178
10	Doug Koch	2.13797

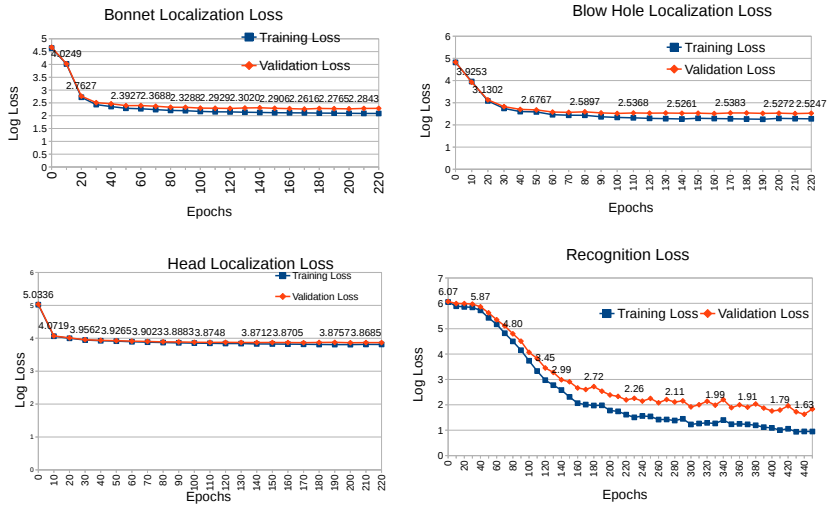

Figure 9. Loss Curve: This figure shows the training and validation loss during the training of the four deep learning models (bonnet localization model, blow hole model, head localization model, and whale recognition model).

the latter. This is likely because for many samples the blow hole is completely covered by water (white pixels) while the bonnet is visible in most of the images.

Figure 9 also shows the performance of loss function of the head localization network. The loss value goes down from 5.03 to around 3.87. Because both the validation and training losses curves are very close to each other, the performance may be improved slightly by using a larger network.

Figure 10. A sample of localized whale bodies.

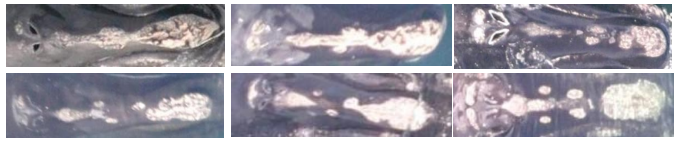

Figure 11. A sample of localized whale heads.

In the context of localization, the log loss function may be difficult to interpret. Of course, the lower the loss function, the better we can expect the localization to be. However, it is also useful to track the quality of the localization visually. Figures 10 and 11 show a random set of images of localized whale bodies and localized whale heads, respectively.

In addition, Figure 12 shows the bonnet location prediction after every 5 epochs (up to epoch 35). The upper row shows the true location of the bonnet while the lower row shows the prediction made by the network. Because we augment the data by flipping the image horizontally and vertically, the true location of the bonnet (in the upper row) changes. At the beginning of the training (epoch 0), the network predicts the bonnet to be in the middle. Then, the prediction starts to improve gradually. At epoch 20, we start to see the network correctly tracking the location of the bonnet. Of course, monitoring one image is not enough so we monitor a small sample of images. Once we were satisfied with the performance of the localization network, we stopped the training. Figure 12 shows one of the monitored images while training the bonnet localization network. We monitor the training performance in the same manner when training the blow hole and head localization networks.

As we mentioned earlier, using a multi-stage approach was the only way to be able to train a deep learning model on this data set. However, when carrying out a multi-stage approach, there is a risk of error propagating from one stage to the next. Figure 13 shows a sample of cases where the whale body crops could not be localized correctly. In the images shown in this Figure, some of them do not show the full callosity pattern on the whale head while others include the whale body oriented in the wrong directory. There are approximately 0.7% of cases where the model could not correctly localize the body.

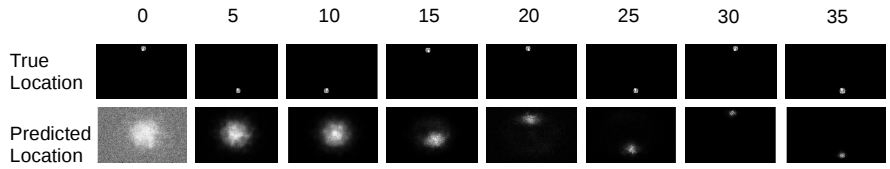

Figure 12. This figure shows how the network performance in tracking the location of the bonnet improves as it is being trained. At epoch 0, the network predicts the location of the bonnet to be in the middle of the image. Later, the network gradually becomes capable at predicting the location of the bonnet.

Figure 13. A sample showing cases where the body of the whale was not localized correctly. In the images shown in this Figure, some of them do not show the full callosity pattern on the whale head while others include the whale body oriented in the wrong directory.

Figure 14. A sample showing cases where the head of the whale was not localized correctly. These images include cases where the callosity pattern is not fully shown in the image.

The head localization error is higher than the body localization error. The head localization error is 2.9%. Figure 14 shows a sample of images where the head was not localized correctly. These images include cases where the callosity pattern is not fully shown in the image.

While the head localization error is relatively low, the union over intersection metric shows that there is a room for improvement. The union over intersection is defined as shown in Equation (8.1):

$$UoI = \frac{M_{true} \cap M_{pred}}{M_{true} \cup M_{pred}} \tag{8.1}$$

where M_{true}, M_{pred} are the true and predicted masks. On the validation set, the average body localization UoI is 0.51. In other words, while the percentage of complete failure in localizing the whale head is relatively low (2.9%), the quality of the head localization can be improved. We strongly believe that using a better model to perform head and body localization is likely to yield a better UoI score. A lower UoI score is likely to lead to a lower whale classification error.

9. Conclusion

We introduced a method to recognize individual whales from the callosities on their heads. This method can help in the conservation efforts of marine biologists. Because the size of the available training images is very low, overfitting is very difficult to avoid. We solved this problem by introducing a model to localize the head of the whale and training the recognition model on it. This helps the recognition model to focus on the callosity features located on the head of the whale. Our model's top-1 and top-5 predictions accuracies are 69.7% and 85%, respectively. We strongly believe that the performance can be boosted by increasing the size of the training set. In addition, improving the body and head localization models is likely to improve the whale classification error.

References

[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, *ImageNet classification with deep convolutional neural networks*, in: Advances in Neural Information Processing Systems, 2012, 1097–1105.

[2] A. Thomas, *whale-2015*, `https://github.com/anlthms/whale-2015` (accessed January 19th, 2016).

[3] A. L. Maas, A. Y. Hannun and A. Y. Ng, *Rectifier nonlinearities improve neural network acoustic models*, in: Proc. ICML **30**, 2013.

[4] B. Graham, *Spatially-sparse convolutional neural networks*, arXiv:1409.6070 (2014).

[5] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard and Y. Bengio, *Theano: New features and speed improvements*, Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[6] C. Khan, P. Duley, A. Henry, J. Gatzke and T. Cole, *North atlantic right whale sighting survey (narwss) and right whale sighting advisory system (rwsas) 2013 results summary*, US Dept Commer, Northeast Fisheries Science Center Reference Document, 2014, 14–11.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, *Going deeper with convolutions*, arXiv:1409.4842 (2014).

[8] F. Chollet, *keras*, `https://github.com/fchollet/keras`, 2015.

[9] F. Lau, *Recognizing and localizing endangered right whales with extremely deep neural networks*, `http://felixlaumon.github.io/2015/01/08/kaggle-right-whale.html` (accessed August 4th, 2016).

[10] G. Bradski, *Opencv*, Dr. Dobb's Journal of Software Tools, 2000.

[11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, arXiv:1207.0580 (2012).

[12] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio, *Theano: A CPU and GPU math expression compiler*, in: Proceedings of the Python for Scientific Computing Conference (SciPy), June 2010, oral presentation.

[13] J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in: Computer Vision and Pattern Recognition, 2009, CVPR 2009, IEEE Conference on, IEEE, 2009, 248–255.
[14] K. Fukushima, *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*, Biological cybernetics **36** (1980), 193–202.
[15] K. He, X. Zhang, S. Ren and J. Sun, *Deep residual learning for image recognition*, arXiv:1512.03385 (2015).
[16] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv:1409.1556 (2014).
[17] Kaggle, *Right whale recognition*, `https://www.kaggle.com/c/noaa-right-whale-recognition` (accessed January 19th, 2016).
[18] N. Otsu, *A threshold selection method from gray-level histograms*, Automatica *11* (1975), 23–27.
[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Dropout: A simple way to prevent neural networks from overfitting*, The Journal of Machine Learning Research **15** (2014), 1929–1958.
[20] Nervana Systems, *Neon*,`https://github.com/NervanaSystems/neon` (accessed August 4th, 2016).
[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein and others, *Imagenet large scale visual recognition challenge*, International Journal of Computer Vision *115* (2015), 211–252.
[22] R. Bogucki, *Which whale is it, anyway? Face recognition for right whales using deep learning*, `http://deepsense.io/deep-learning-right-whale-recognition-kaggle/` (accessed August 4th, 2016).
[23] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu and the scikit-image contributors, *scikit-image: Image processing in Python*, Peer J. **2** (2014), e453.
[24] V. Nair and G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in: Proceedings of the 27th International Conference on Machine Learning (ICML–10), 2010, 807–814.
[25] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks*, in: International conference on artificial intelligence and statistics, 2010, 249–256.
[26] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE **86** (1998), 2278–2324.
[27] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, *Deepface: Closing the gap to human-level performance in face verification*, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, 1701–1708.

AbdulWahab Kabani, Department of Computer Science, University of Western Ontario, Address 1151 Richmond St, London, ON N6A 5B7
e-mail: `akabani5@uwo.ca`

Mahmoud R. El-Sakka, Department of Computer Science, University of Western Ontario, Address 1151 Richmond St, London, ON N6A 5B7
e-mail: `melsakka@uwo.ca`