OrcaNet: A Neural Network for Killer Whale Classification

Matheus Stolet
University of British Columbia
stolet@cs.ubc.ca

Lawrence Li
University of British Columbia
macknever@gmail.com

Abstract

We propose OrcaNet, a set of two classifiers and an autoencoder used to extract a segmentation mask from images of killer whales and used to predict the ID tag and the type of the whale in the picture. The segmentation mask derived by the autoencoder is used as input to the classifier that the predicts the ID of the whale. For reduced complexity, the classifier that predicts the type of the whale does not use an autoencoder, since it was able to achieve good accuracy without the segmentation mask. The classifier that outputs the type of the whale achieved a 95% accuracy when distinguishing between transient and southern resident orcas. Identifying the exact whale in the picture was more challenging and our classifier for that task only had 20% accuracy when outputting the ID tag of the whale in the picture.

1. Introduction

Killer whales, also known as orcas, are a type of whale belonging to the dolphin family. These whales are found in oceans all over the world, but they take an important role in the Pacific Northwest of North America, where they help control the population of their preys, and also have a significant cultural role in the region. Declining food resources and contaminants in the ocean have caused the population of these animals to dwindle, making them animals of special interest to biologists and marine researchers.

Researchers at UBC gave us access to a dataset containing pictures of 40 different orcas living in the coastal waters of British Columbia. Each picture in the dataset was labeled with the ID tag of the orca, and its type (southern resident or transient). For this project, we set out to create a neural network that is able to identify the type of killer whale in a picture and can also output the ID tag of the specific whale in the picture. Our classifier can be useful to researchers keeping track of these whales for conservation purposes. The laborious process of having an expert manually label the images of hundreds of whales can be automated by a fast and accurate method for labeling the type of the whale or its ID. These labeled images can then be used by marine researchers to keep a status on the population of transient and resident whales. They can also be used to keep track of specific whales being studied by the researchers.

We created two neural networks, one that outputs whether an orca is part of the transient or southern resident population and another that creates a segmentation mask of the image of a whale and uses this segmentation mask to output the ID tag of a whale. For the task of classifying whether an orca is transient or resident, we developed a simple convolutional neural network with max-pooling layers for downsampling. To create a segmentation mask from the image and output the proper ID of the whale we created an autoencoder that downsamples the image to find a latent representation. This latent representation is then decoded by using a series of transpose convolutions that upsample the latent representation into a segmentation mask of the original image. This segmentation mask was then fed as input to a classifier that outputted the ID of the whale in the picture. We achieved 95% classification accuracy in our validation set for identifying the whale population from the image. For the task of identifying the ID of the whale, our classifier was not as successful and achieved 20% accuracy due to the large number of possible output labels (40) and the subtle differences between whales in the same species.

2. Related Work

Many approaches for object detection have been developed. R-CNN approaches [2] [1] [6] are region proposal based deep learning approaches. They try to predict the probabilities of pixels belonging to a certain object or background using the region proposal technique, making cheaper calculation and taking advantage of only the CNN.

By continuously improving the efficiency of the region proposal method, Faster R-CNN [6] and Fast R-CNN [1] optimizes and enhances the strength of region proposal. Faster R-CNN's training time and test time are far less than R-CNN, making real-time detection possible. But in essence, these methods are still training through the region proposal.

R-CNN relies on the region proposal, thus losing overall information coming from the whole image. YOLO [5] di-

Figure 1: Sample images of transient and resident orcas in our dataset coupled with their segmentation masks. The first image is of a resident orca and the second image is the segmentation mask of the first image. The third image is of a transient orca and the fourth image is the segmentation mask of the third image.

vides the picture into $s \times s$ grids, and performs prediction and probability calculations for each grid cell. The YOLO technique is fast. It can process images at 45 frames per second, a lighter version can even bring the speed to 155 frames per second. However, YOLO yields negative results for small objects in the image. Recently, Facebook Research developed a new image detection framework called Detectron2, implementing Mask R-CNN [4] in Pytorch. The framework provides a pre-trained network to train customer datasets, and it achieved positive results in COCO.

3. Design

State-of-the-art architectures like [2, 1, 6] are great models to identify the differences between different objects. The pre-trained model of [6] can perfectly detect the location of the object.

However, telling the differences between two objects is one thing; classifying different IDs from the same species is another. When applying Detectron2 to an image in our dataset, the model can detect the shape of the body, but it struggles to find the shape of the white patch on the back of the whale, which is the key to distinguish between different whales.

R-CNNs and YOLO are nice models to detect the outer shape of the objects, but our task needs to detect not only the shape of the body but also the features shown on a whale's back. The particularity of our task make the pre-train model of [2, 1, 6] hard to apply. Also, to utilize this model, we need to annotate our dataset to fit the requirement of training, which is more expensive than building our own model to train.

We separate our task into two parts, detection and classification. We first try to detect the whale, capture the shape of whale's fin and patch. And then, we train a classifier to find the bond between the detected information and the ID of the whale.

3.1. Autoencoder & ID Classifier

We got our inspiration from Mask R-CNN [6]. Rather than using the pre-trained model, we borrowed the idea from the 'mask'. We built masks for images. The masks have three segmentations, background (black), body (shallow gray), and patch (dark gray). Next, we built an autoencoder and an autodecoder. Training them to find the latent information between the images and the masks. The decoder can reconstruct the mask from latent info which is abstracted from the image and we believe the latent information captures the features of the fin and the patch from the data.

After we got the latent information, the second part of our task was a simple classification task. We use the cross-entropy function as our loss function.

3.2. Type Classifier

For the task of classifying the type of the orca (transient or resident) we chose to use a simpler neural network that did not use a segmentation mask or an autoencoder. This choice was made because we were able to get good accuracy with this simpler network, so there was no benefit in using a more complex network. Our approach to this project was to increase complexity as it was needed. This let us use a smaller network for classification with fast training, which allowed quick prototyping. The network used for this task was composed of two convolutional layers, where each layer was followed by a max-pool filter. After the input image was processed through the convolutional layers and the max pool filters, it went through a fully connected layer, followed by the final output layer. The architecture of the network can be observed in Figure 2

4. Implementation

There were three main efforts in the implementation stage of this project: creating segmentation masks, data cleaning, and developing a neural network architecture. The images given to us did not have the segmentation mask, so

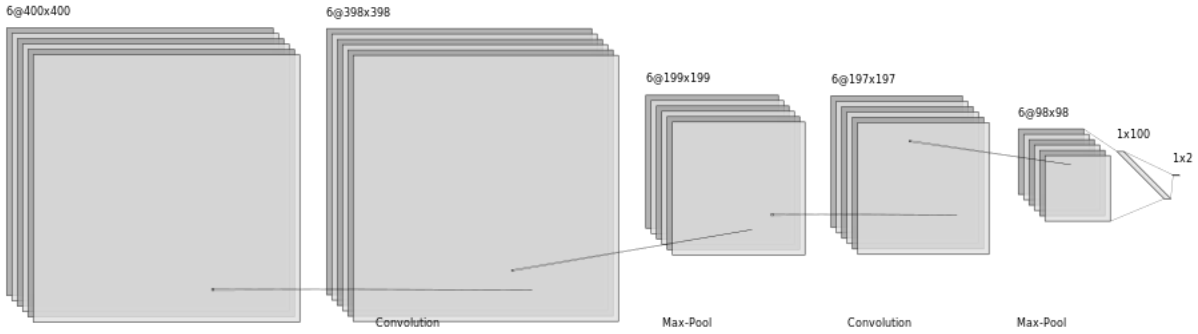

Figure 2: Architecture of the classifier used to identify the type of whale.

we had to manually create a segmentation mask for each image. After getting the segmentation masks, we had clean the data before we could feed it as input for a neural network. The dataset given to us was composed by a collection of images divided into folders, so we wrote a Python script that went through each image and extracted the ID and type of the orca based on the folder the image was located.

We also had to perform a resizing operation on each image to make sure they were all the same size. After each image was resized to be 400 pixels wide and 400 pixels high, we created a dictionary where each element in the dictionary had the image for a whale, the segmentation mask for that image, the ID of the whale, and the type of the whale as attributes. This dictionary was used as input to Pytorch implementations of the autoencoder and classifiers we developed.

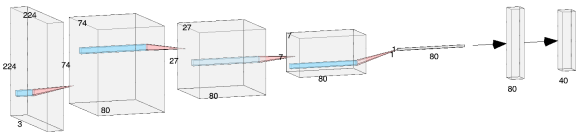

Figure 3: Architecture of the encoder.

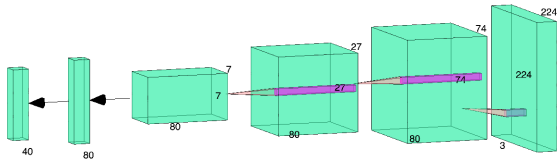

Figure 4: Architecture of the decoder.

4.1. Autoencoder & ID Classifier

The structure of the encoder and the decoder are almost the same. As shown in Figure 3, there are 40 whale IDs, and we try to encode the image into a 40×1 vector.

The complexity of the network should be consistent with the difficulty of the task. So when dealing with net depth, we did not use a very deep network because of our small training set. Furthermore, a complicated network would make the training even more expensive. The encoder is composed of five convolutional layers. Each convolutional operation is followed by a ReLU activation function. Two fully-connected layers make the output of the encoder a vector, which contains the latent information of the fin and the patch.

The decoder contains 4 Transpose Convolutional layers and 2 fully-connected layers. The decoder is reconstructing the mask of the image from the latent information captured by the encoder. Figure 4 shows the structure of the decoder.

4.2. Type Classifier

Our type classifier was composed of two convolutional layers with 3x3 kernels of stride one. Each convolution was followed by a 2x2 max pool filter with stride two. The initial training of this network was challenging to tune properly. Some initializations would not train properly and the gradients would get stuck at zero as the iterations progressed. We suspected we were facing a vanishing gradient problem, so instead of using ReLU as our activation function, we chose to use a Leaky ReLU instead because it is more robust to the vanishing gradient problem.

We divided our dataset into 80% training set and 20% validation set. The network was trained using an Adam optimizer with a batch size of six for a total of ten epochs. We used a cross-entropy loss to evaluate the loss between the predicted values for the type of killer whales and the target type label.

5. Evaluation

5.1. Autoencoder

We evaluate our Autoencoder numerically and visually. When training our Autoencoder, we apply the MSE loss function and select Adam as optimization method. The image sizes in our data are not uniform, and the height and width vary from 500 to 1000. Since we do training by batch, when loading the data we resize the image to obtain uniform measurements. Of course, the larger the size of the image, the longer the training time. We adopt two sizes of 224 $\times$ 224 and 400 $\times$ 400 to train the autoencoder. The quantitative results are shown in Table 1.

Table 1: Numerical result of Autoencoder.

Loss of reconstruction		
Size	224$\times$ 224	400$\times$ 400
Training	0.00342	0.0116
Validation	0.0028	0.00648

The result was unexpectedly good. We trained 500 epochs on the two different sizes. It took about 20 hours in total. The autoencoder achieved good similarity between the original image and the segmentation mask. It is extremely easy to overfit. But from the numerical results, after 500 epochs training, the training results are very good. The validation results are similar to the training results, or even better in some cases.

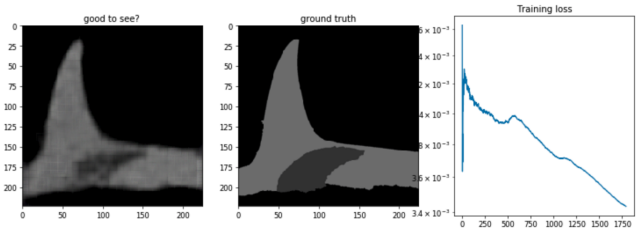

Figure 5: Training results of 224 $\times$ 224 images.

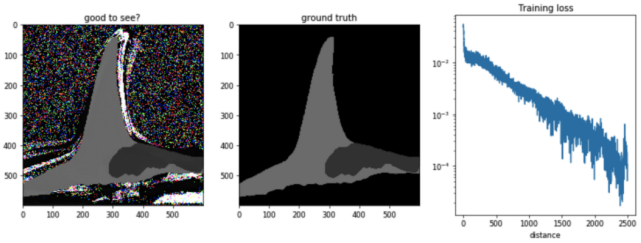

Figure 6: Training results of 400 $\times$ 400 images.

Visually, the encoder's training results are of equally high quality. Figures 5 and 6 show the training results of 224$\times$224 and 400$\times$400 images, respectively. The reconstructed mask can capture the shape of the fin well, and can locate some of the patches. In the validation results, it is hard to tell the difference between them and the training results. If the whale is small in the image, the results are not satisfactory. We can clearly observe the validation results of our Autoencoder from Figure 78910. When the proportion of the whale's body in the image is large, the reconstruction result from the decoder is satisfying. On the other hand, when the whale is small in the image, the result is very fuzzy. Even if the quantitative loss is small, the information captured from such kind of image by our Autoencoder is far from accurate.

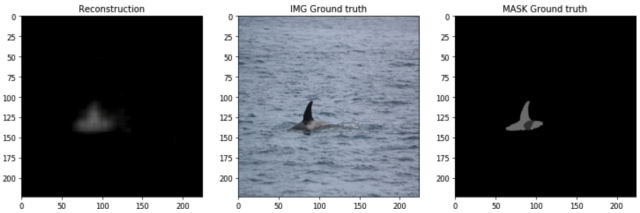

Figure 7: Validation results of 224 $\times$ 224 images (low quality).

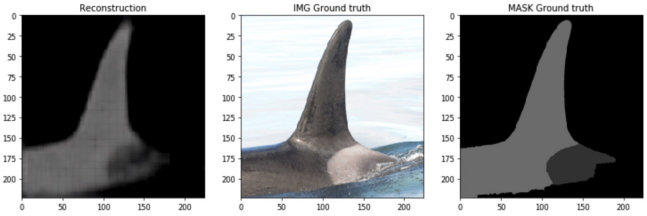

Figure 8: Validation results of 224 $\times$ 224 images (successful).

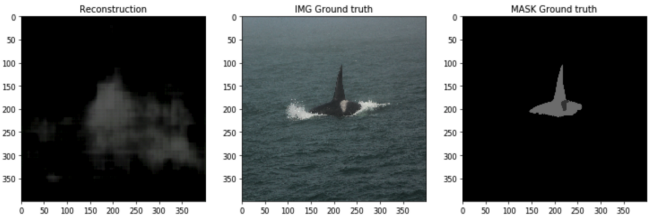

Figure 9: Validation results of 400 $\times$ 400 images (low quality).

These results also show that when the proportion of the whale body in the image is relatively stable, the training results will be better. Therefore, future work can apply some detection models to crop the image, so that the main part of the object in the image is as large as possible. At the same time, when the epochs for the training are the same, the larger the size of the image, the worse the training effect.

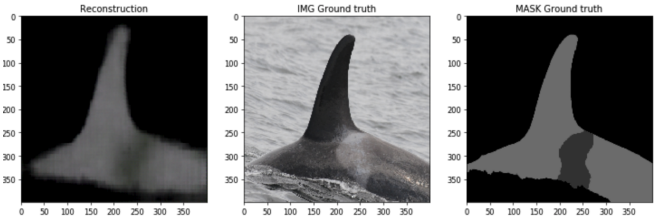

Figure 10: Validation results of 400 $\times$ 400 images (successful).

The complexity of our tasks is not very high and the information that needs to be acquired does not require large-size images to carry. Therefore, reshaping to a relatively small size, is conducive to training.

5.2. ID Classifier

The results of the ID classifier are not great. As shown in Table 2, ID classifier only has a chance of 20 percent to accurately classify orcas based on images. Considering that the sample size is very small and there are many possible output labels, this result is not too bad. We can get 2.5% of accuracy if we just randomly guess the ID of the whale. At the same time, as shown in Table 3, the accuracy of top-k id classifier is barely acceptable.

Table 2: Validation Accuracy for ID Classifier.

Validation Accuracy for ID Classifier	
Latent ID Classifier	20%
Random Guessing	2.5%

We trained our id classifier separately for two different sizes of autoencoder. We also directly trained the classifier based on the image for comparison. The results can be seen in Table 3. Our model of autoencoder did learn the information of the whale's fin and patch. Given the same training time, a small image size was better for learning.

Table 3: Top-K Accuracy of ID Classifier.

Top-K Accuracy of ID Classifier				
Classifiers	Top 1	Top 3	Top 5	Top 10
Latent(224) ID Classifier	20.1%	30.4%	38.6%	42.3%
Latent(400) ID Classifier	6.6%	18.6%	22.0%	28.2%
Directly ID Classifier	0%	0%	7.6%	19.0%

5.3. Type Classifier

To evaluate our classifier we measured the classification accuracy when identifying between southern resident and transient orcas in our validation set. Our classifier from Figure 2 achieved 95% accuracy. The high accuracy for this task was impressive for such a simpler network, but it can be explained by the small number of output labels. If the network randomly chose a label it would already achieve 50% accuracy. Nonetheless, it was still surprising that we achieved such high accuracy, since the differences between resident and transient orcas can be very subtle to the untrained eye.

To motivate our choice for a simpler network. we also compared the validation accuracy of our type classifier to a slightly modified version of the ID classifier. The ID classifier uses an autoencoder to extract a segmentation mask from the original image and then uses this mask to perform classification. This approach achieved better results than using a classifier that made its prediction only from the original image in the ID classification task. Interestingly, when our ID classifier was modified to identify the type of the whale, it did not yield any improvements over our type classifier network that did not use an autoencoder; both approaches reached 95% classification accuracy in the validation set. This shows that for this task we don't need a complex network architecture to achieve good accuracy, so we favoured a simpler one to remove unecessary complexity.

Validation Accuracy for Type Classifier	
Using segmentation mask	95%
Without segmentation mask	95%

Table 4: Table comparing the validation accuracy of the same classifier used for the ID classification task (the one using the segmentation mask) and our redesigned simpler classifier.

6. Discussion

One area open for future work in this project is data augmentation. Unfortunately, we ran out of time, but it would be interesting to test different data augmentation techniques such as mirroring the images, adding noise, cropping, or performing hue and saturation adjustments. These data augmentation steps are important for because of the costly process of having an expert go through the images to label them. This step may take a long time to complete, and there are not many experts that can reliably identify between southern and resident whales, thus making the labeled images we have at hand fairly precious.

Another interesting direction for future work in this project is the exploration of a synthetic data generator. To create more images for our training set, it could be interesting to use a generative adversarial network [3] to generate synthetic images. These images could then be used during

training to increase the size of our training set. One of the challenges with this approach is that our small dataset might lead to a poor synthetic data generator, but it could be interesting to investigate this approach.

Also, in this task, our research on classifier is not deep enough. In the future, we can consider classifying latent information with the combination of decision tree and softmax. Furthermore, because the autoencoder is supporting classifier, the feasibility of training autoencoder and classifier at the same time is also worth studying.

7. Conclusion

In this project, we proposed a set of neural networks that can create a segmentation mask from pictures of killer whales and can also perform two different classification tasks. We increased the complexity of our neural networks according to the difficulty of the task and achieved good results in classifying the type of whale. We were not as successful in classifying the ID of the whale, but we expected this to be a challenging task and we were not hopeful that it would be successful, since we had around seven images for each label in our dataset.

References

[1] R. Girshick. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH*, pages 580–587, 2014.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, 2020.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.