A STUDY OF CLASSIFICATION AND EMBEDDING METHODS FOR IDENTIFYING HUMPBACK WHALES

Stéphane Junior Nouafo Wanko

Submitted in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

Approved by:
Dr. Charles Stewart, Chair
Dr. Barbara Cutler
Dr. Alexandre Gittens

Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York

[August 2020]
Submitted July 2020

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Many current methods for the identification of individuals of a species do not consider the problem of identifying previously unseen individuals. To be used in a real-world setting, these methods must be able to recognize that all individuals they encounter will not all necessarily be part of the set of individuals the methods were trained to recognize. In this thesis, we explore two different approaches for the identification of new individuals.

The two approaches that we consider are a classification-based approach and an embedding-based one. We were able to achieve a top-1 accuracy of 83.0% for the classifier and of 80.5% when using embeddings. While both approaches showed good results towards identifying novel individuals, there were drawbacks and benefits to using one over the other. Most importantly, we show that a classification-based approach is most appropriate for quickly learning the weights for the used model. It also consistently performs better overall than the embedding-based approach. When using embeddings however, because of the use of an embedding function to acquire the feature vectors that represent our known individuals, there is the possibility to convert new individuals to known individuals without the need for retraining. We achieve a top-1 accuracy of 76.5% with our embedding approach on newly added individuals with no retraining. We also show that a trained classifier can be converted to an embedding model with no or minimal retraining needed.

1. INTRODUCTION

While studies on humans are extremely common, there is still much that is unknown about the migration patterns of birds, the number of species that roam the earth, as well as what lives in the expansive oceans. Being able to correctly identify individuals in animal groups is a crucial task to enable the proper monitoring of populations. Unfortunately, current methods of population monitoring are mostly intrusive. Some of the oldest and most popular methods used include: quadrant count, mark-recapture, and mark-resight [1]. Quadrant count is purely for finding abundance. It involves counting the amount of plants or animals in a known area. Mark-recapture involves capturing of individuals and marking them to later resample what percentage of the population carry marks. And mark-resight is just like mark-recapture except there is no recapturing. The marking method used in this case must allow identification of the mark from a distance [2]. There are various drawbacks to the use of these methods.

Quadrant count does not identify individuals. That makes it inappropriate for anything but monitoring a population's size. Mark-recapture is more versatile than quadrant count as it allows to measure population changes in a wider area. However, it relies on assumptions that are unrealistic for many settings, such as assuming there is no migration. The marking method used for mark-recapture also does not necessarily help keep track of individuals. Mark-resight is less intrusive than mark-recapture but still poses the same problems [1]. There is a better alternative to using these more traditional methods: camera traps. Using camera traps has shown good results in terms of yielding data with less intrusion and a greater efficiency [3]. Ecologists have combined the convenience and versatility of camera-trap data with the models already developed for the traditional mark-recapture and mark-resight methods to get better insights into populations [4-6].

Using images from camera traps and other sources for the re-identification (re-ID) of individuals is not a new idea [7]. The use of camera traps can greatly reduce the workload for ecologists while maintaining or increasing data fidelity. This can make the work less invasive for ecologists but there are also drawbacks to this method. The camera setup can be expensive and properly trained field workers must be available to maintain the equipment [5]. And this is prone to human error and biases. These human errors can lead to unreliable data and questionable results [8, 9]. A big reason for these errors is because humans are not perfect at recognition. Recognizing individuals of a separate species is not easy. The advent of deep learning has brought a new approach to this however.

The increasing availability of large amounts of data, better and cheaper storage options, and more powerful computing power have been heralds for the great potential deep learning has to solve many of our problems. From showing the feasibility of creating Turing machines from combined neurons [10], researchers are now looking into optimizing production schedules using neural network (NN) based algorithms originally developed to play games [11]. The impact of the development and application of neural networks is clear throughout many domains. This is no different for the purposes of population monitoring. To this end, image classification is one of the main tasks worked on by humans where neural networks show great promise.

Image classification has become the cornerstone for the development of many machine learning tasks in the past decade. After first being trained on the ImageNet dataset, image recognition networks can be adapted to other tasks such as object detection, action recognition, and many others. These adaptive networks have produced impressive results [12-14]. The same principle is being applied to the approach being discussed in this work. Image recognition networks have shown excellent results towards the identification of humpback whales [15].

The problem of identification is one that classifiers are designed to solve. However, handling a growing population is hard for a classifier as the whole network needs to be retrained for each set of new individuals it needs to recognize. Being able to remove the need to retrain this network would greatly enhance its potential applications in the real world. If we were to be able to treat each image as a vector and group the vectors representing images of the same individual in an embedding space, that could help address our concerns. Therefore, we want to treat this problem of recognizing humpback whales as an embedding problem and to compare the potential differences to treating it as a classification problem.

A very important type of loss for training embedding networks is the triplet loss function [16]. Triplet losses have helped classification networks achieve much greater accuracies [17, 18]. These types of losses work by trying to maximize the distance between feature vectors representing different classes in a metric space. We use the triplet loss to train both our classifier and embedding model and look at the differences between these approaches. In this thesis, we analyze the impact of the number of training samples on accuracy, the ability of our two approaches to recognize previously unseen individuals, and the benefits and drawbacks of using a classification-based vs an embedding-based approach.

2. RELATED WORK

There has been work done by Weideman et al. [19] on representing the trailing edge of whale flukes and dolphin fins for identification purposes. Their aim was to address problems presented by differential curvature measures by proposing both a dorsal fin integral representation method, that returns stable representations, as well as two different classification algorithms for the identification of individuals. Their proposed use of the local naive Bayes nearest neighbor (LNBNN) algorithm [20] with their integral curvature representation gave impressive results. However, methods like these often require extensive labelling of images and are not appropriate for all applications.

Training an end-to-end classifier can help relieve some of these problems. There has been work done by Körschens et al. to that aim [21]. They discuss a pipeline for the identification of individual elephants from photos taken in the wild. Their approach incorporates a YOLO detector [22] for bounding box detection of an individual's head. They then proceed to use a modified and pretrained ResNet50 [23] for feature extraction. These features are then used with a support vector machine for ranking of potential individuals and the final decision is left to the user. Out of 276 elephants, they could achieve 72% and 85% top-5 accuracy with 1 image and 2 images used for classification respectively. Those are pretty good results, but the problem of identifying new individuals is not at all addressed in this paper.

Various approaches have been taken for the purposes of re-identification of individuals. One such approach is by Schneider et al. [24], where they experiment on the use of Siamese networks for the re-ID of animals. Their goal was to surpass the accuracy achieved by human observer and make better use of data acquired through camera traps. They compare their results

using multiple datasets representing a multitude of species as well as various model types. From their experiments, they concluded that the tested networks were superior to a Siamese-based network when using a triplet loss for all tested species. These experiments further showcase the performance that a triplet loss can bring to the problem of classification but still does not deal with the identification of unseen individuals.

Moskvyak et al. [25] approach the issue of re-identification by looking at natural markings on manta-rays. Their idea is to use embeddings that are pose invariant as an alternative to a classifier. They achieve good top-5 and top-10 accuracies of 95.7% and 97.8%, showing that using an embedding approach can result in a successful representation of individuals without any special work done to actually capture the marking information as is done in [19]. This paper showcases that the use of an embedding as a representation mechanism can be very versatile. We extend this idea here by comparing the performance of an embedding-based approach to a classification-based one on humpback whales.

3. METHODS

3.1 Data and Preparation

The humpback whales (*Megaptera novaeangliae*) dataset used for this thesis was provided by Wild Me and is illustrated in Figure 1. This dataset consists of 6181 images and 2905 individuals. These images have been cropped tightly around detected flukes. The individuals were categorized according to the number of times pictures of them were captured. Each image of an individual, or encounter, was taken during a separate 24-hr period. The distribution of the data can be seen in Figure 2. As illustrated by Figure 2, the number of images/encounters per individual varies greatly. The majority of the data can be found with individuals with a lower amount of encounters; that is why Figure 2 does not contain individuals with 1, 2, or 3 encounters.

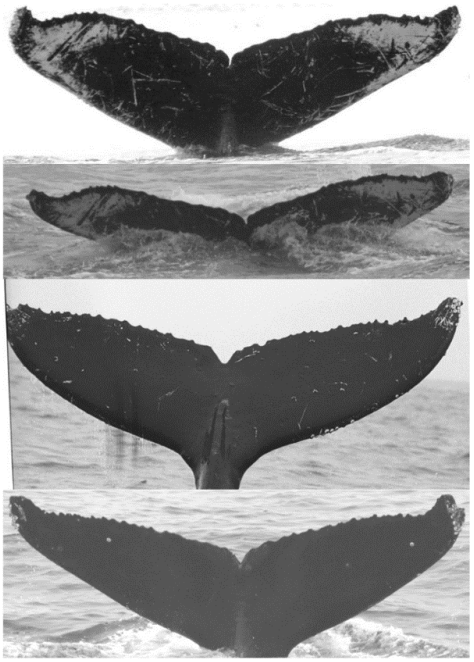

Figure 1: Example images from the Wild Me humpback whale dataset. The top two images represent the same individual while the bottom two are from different individuals.

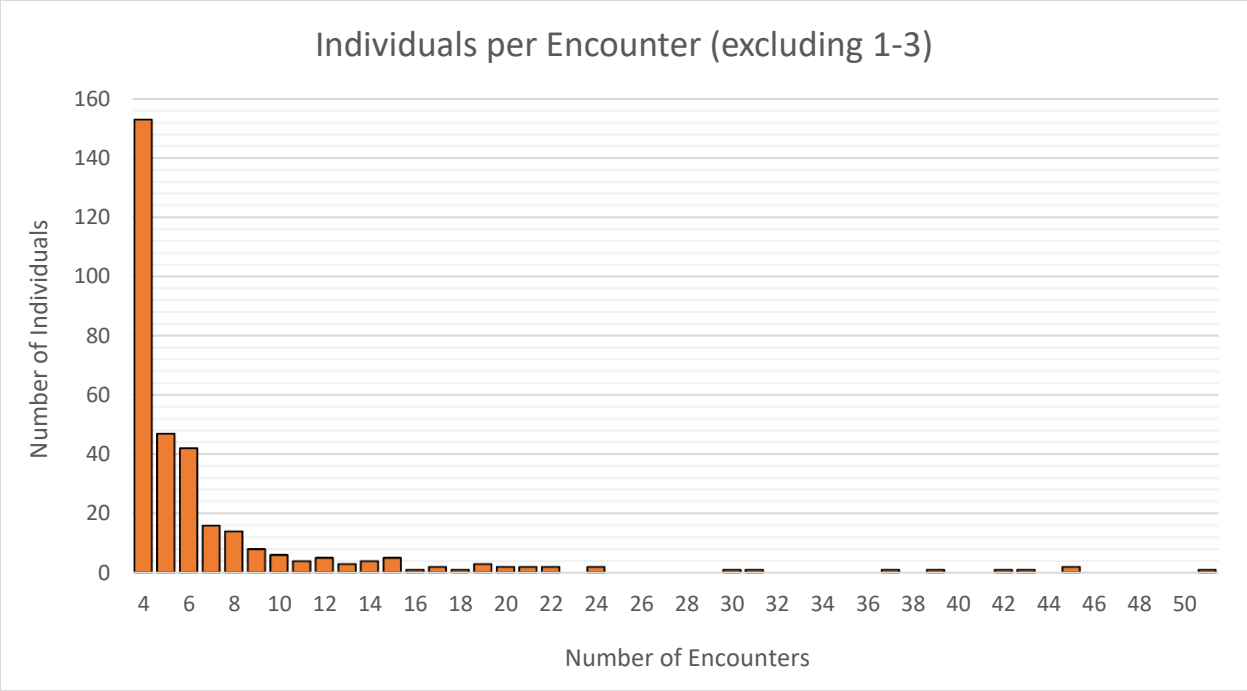

Figure 2: Number of individuals per number of encounters. This graph shows the distributions of the whale dataset. Individuals with 1 (1866), 2 (200), and 3 (508) encounters were excluded from this graph because of the difference in the amount of data available.

All individuals with 3 or more encounters were assigned an ID, resulting in 839 "known" individuals or individuals with an ID. The 200 individuals with 2 encounters were reserved to be used for embedding experiments. The remaining images, of individuals with only 1 encounter, were all grouped into one "class" or ID: 'new_whale'. These 'new_whale's represent individuals that the model must learn to recognize as not previously seen. There is a total of 1866 'new_whale' images and so 1866 individuals all represented as 'new_whale'. For the purposes of this thesis, it was assumed that any specific individual represented as a 'new_whale' could not be present in more than one set, the sets being the training, validation, and testing sets. Overall, there were 1040 individual classes, consisting of 1039 actual whales and 1 "new_whale' class.

The subset of the humpback whale dataset used for validation and testing is comprised of individuals with at least m encounters. The choice of m is critical as it defines how many

encounters per individual are needed for the model to be able to reliably recognize that individual. This was something that needed to be tested and is further explored in the results, in section 4.1. We did know however, that as all individuals in the testing set were also found in the training set, m needed to be at least 2. Since all 1040 individuals had at least 2 encounters, none would need to be excluded if m were equal to 2. However, as 200 out of the 1040 individuals were put aside for embedding tests, only 840 of them were used for training.

When creating the validation and testing sets, it was critical for accurate testing that all individuals present in the validation and testing sets had at least $m - 2$, and preferably more, images in the training set as well. Individuals that had less than m encounters were not split. Instead, all their images were included in the training set to maximize the amount of training data available to our model. If included, individuals with only 4 encounters were split so as to have 2 of their images in training, 1 in validation, and 1 in testing. The rest of the individuals, excluding the 'new_whale' class, were split in order to have 80% of their images in training, 2% in validation, and 18% in testing. Lastly, if included, the 'new_whale' images were split to create a consistent proportion of 'new_whale' vs "known" whales in the training, validation, and testing sets. The consistent proportion applied to the sets was done to help make sure we were testing what we were telling the model to learn. This value was 34% for our experiments to maximize the amount of data used. The choice of this proportion depends on the data available and can be altered.

3.2 Model and Training

The source code for the base model was written by the winners of a 2019 Kaggle "Humpback Whale Identification" competition [15]. Modifications and enhancements following the approach taken by the top contestants of the competition were also tested for possible accuracy

improvements. We further modified the source code to fit our dataset and system requirements. Finally, we also made additions for testing how an embedding-based approach would compare to a classification-based one. There were two different training processes used for training our model: one for approaching this as a classification problem and another for approaching it as an embedding problem.

The model base used by the competitors and chosen for our experiments was the SENet-154 [26] without its classifier head and pretrained on the ImageNet dataset. Given an image, the model was modified by the competitors to return three outputs: "local_features", "global_feature", and classification outputs. The "local_features" were the resulting features of the SENet-154 averaged over its width. They were essentially average-pooled, downsampled, and normalized horizontal stripes of the input data's resulting feature matrix in accordance with [27]. The normalization of the "local_features" converted them all to unit vectors. The "global_feature" were basically the "local_features", except they were averaged across the whole input's resulting feature matrix and they were not downsampled. A dropout layer was also applied to "global_feature". The "global_feature" were then passed through a fully convolutional layer and multiplied by $w = 16$ to create the classification outputs. Model training and testing were performed using Python 3.6.9 and PyTorch 1.1.0 on 32GB NVIDIA Tesla V100 GPUs.

3.2.1 Loss

The loss function L we used was implemented by the competitors for the purpose of training the classifier and has two parts. A sigmoid-based cross entropy loss L_{BCE} and a triplet-based loss L_T.

The cross-entropy loss is used on the classifier outputs. Given a range of predicted probabilities $p_x = p_1, \dots, p_C$ that image x is of class $1, \dots, C$ and a one-hot encoding $h_1, \dots, h_C$ of the correct class, we can calculate the error e_x such that

$$e_x = abs\{h_1 - p_1, h_2 - p_2, \dots, h_C - p_C\}. \quad (1)$$

After getting the error e_x, the target error r_x is then set to be a vector of 0s for each class. The binary cross-entropy loss $loss(e_x, r_x)$ is then

$$loss(e_x, r_x) = mean\{l_1, \dots, l_C\}, \;\; l_c = -[r_x^c \cdot log\, e_x^c + (1 - r_x^c) \cdot log(1 - e_x^c)]. \quad (2)$$

The final calculation is done with a set s_x containing the top q results with the highest errors of e_x. $q = 30$ for our experiments. Let's call the final result of this calculation $loss(s_x, r_x)_A$.

A separate loss value $loss(e_x, r_x)$ is also calculated, where $e_x = p_c$,. p_c here represents the predicted probability for the correct class c image x belongs to. Image x is only included in this loss if it represents a known whale. Let's call this $loss(p_c, r_x')_K$. In this case, the probability p_c is used directly and the target r_x' is set to 1, as the returned probability should be as close to 1 as possible. This loss helped increase the importance of getting the correct predictions for the known individuals as the error values for these predictions might not be included in $loss(s_x, r_x)_A$. The cross-entropy loss used is thus

$$L_{BCE} = loss(s_x, r_x)_A + loss(p_c, r_x')_K. \quad (3)$$

The second part of the loss, the triplet loss, is based on the loss described in Schroff, Kalenichenko, and Philbin's FaceNet paper [16]. An embedding of an image x is represented by $f(x)$, the resulting feature vector returned by the model given an image x. For each individual, an image x_i^a (*anchor*) is randomly chosen to represent all other images x_i^p (*positive*) of the same

individual and an image x_i^n (*negative*) is chosen to represent another individual. What we are then trying to do is to push embeddings of images $\left(x_i^a, x_i^p, \dots\right)$ of an individual close to each other and separate these embeddings from other individuals $(x_i^n, \dots)$ by ensuring

$$\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \; = \; d_2 + \alpha < d_1, \qquad (4)$$

where α is our margin and d_1 and d_2 are the Euclidean distances between the feature vectors of the anchor $f(x_i^a)$ and the chosen hard negative $f(x_i^n)$ and the anchor and the chosen hard positive $f(x_i^p)$ for an anchor x_i^a in the mini-batch. Following the Kaggle competitors' use, $\alpha = 0.3$ was used for the experiments in this thesis. What we are trying to minimize is then

$$loss(d_1, d_2) = max(0, \; -(d_1 - d_2) + \alpha). \qquad (5)$$

We use this loss with the two sets of features returned by our model. When using the returned local_features, we get $loss(d_1, d_2)_L$. Using global_feature results in $loss(d_1, d_2)_G$. Our triplet loss is then

$$L_T \; = \; loss(d_1, d_2)_L \; + \; loss(d_1, d_2)_G, \qquad (6)$$

and the overall loss *L* is then

$$L = L_{BCE} + L_T. \qquad (7)$$

3.2.2 Training and Classification

For each image representing an individual passed through the model, two or three other images are matched with it. The original image of the individual (*anchor*) is matched with another image of the same individual (*positive*) and one or two images of other individuals (*negatives*). Two negative images are passed when the training set includes 'new_whale' individuals. In this

case, one of the negatives is always of a ‘new_whale’ as this is an important, non-centralized class that represents various individuals and so can benefit from more representation.

After acquiring this set of images, each of the three or four images is then randomly horizontally flipped, rotated, shifted, and scaled. If the image is flipped, this new image is given a unique ID. All flipped images of an individual are represented with the same unique ID. Flipping the input image essentially allows us to treat this initial image as if we had two different individuals represented in the same image. It is important to note that the classifier is made to output twice the amounts of classes that actually exist, since each flipped individual gets its own unique ID. Finally, for each image, noise is also added, lighting is altered, and part of the image is erased, all randomly. This altered set of images is normalized, then passed through the classifier. When doing embedding, we added some modifications to this process. The modifications are explained in section 3.2.4.

The classifier results are used to get the loss for backpropagation. Details of the loss function can be found in the previous section. The last step was then accuracy calculation. So far, the model outputs have no indication of whether or not the model believes the images represent a ‘new_whale’. Deciding whether the individual in the image is a ‘new_whale’ is not actually done through including it as a class for the model to predict. What the system does instead is that a probability of 0.5 is added to the classifier output for the ‘new_whale’ class. Note that this ‘new_whale’ probability is only used for calculating accuracy and so does not influence the loss.

A probability of 0.5 for the ‘new_whale’ class is effectively saying that there is a high chance that the input image is a ‘new_whale’. During training, the model should learn to predict that ‘new_whale’ images do not represent any of the other classes and so a probability of 0.5 should

be easily satisfied if the model learns properly. The goal is to leave it to the validation phase to figure out what is the most appropriate threshold to decide whether an image represents a 'new_whale', as discussed in the following section. Picking a threshold for the 'new_whale' ID makes intuitive sense as, in essence, the system only designates an image as representing a 'new_whale' if the image is not similar enough to anything else that is found in the dataset.

3.2.3 Validation

The validation process is mainly the same as the training process. One of the differences is that all input images are horizontally flipped before being passed to the model and no other augmentation is done. The average of the classifier's results for both the original image and its flipped counterpart is then used as the probability vector for the input image. After all the images have been converted to probabilities, during validation is when we need to figure out what is the best threshold to decide whether an image represents a 'new_whale'.

To decide on the best threshold t, we consider different values and see which one gives the highest mean average precision (mAP) when considering the top J predictions. The metric, mAP, was defined as

$$mAP@J(g) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{min(C,J)}\frac{val_n^j}{j}, \quad (8)$$

where g is an NxC matrix containing the model predictions being analyzed, N is the number of images in the mini-batch, C is the number of classes predicted, and val_n^j indicates, by returning a 1 (valid) or a 0 (invalid), whether or not the j^{th} top prediction for g_n is a valid label. A valid label is one that is both correct and the first correct prediction for g_n.

The probabilities for each class prediction range from 0 to 1 and so the possible thresholds were selected from the same range in 0.1 increments. Each possible threshold t_s, as described for the training phase, was added to the initial results of the classifier. These new results g_{t_s} were then used, with $J = 10$, to get a mAP at 10, or $mAP@10$. The smallest threshold t that resulted in the highest $mAP@10$ was then picked as the one to use when testing or

$$t = min\left[\underset{t_s \in [0.1,0.2,\ldots,0.9]}{argmax}\ mAP@10\left(g_{t_s}\right)\right]. \quad (9)$$

3.2.4 Embedding

The main differences we implemented with the embedding-based approach were how the model outputs were acquired and the loss function. To get these new outputs, the 2048D global_feature result, from running the model, is used as the feature vector for a given image x. We did not also use the local_feature result as global_feature is essentially an averaged local_feature. We then acquire the global_feature outputs for all training images with the current model state. These embedded vectors are used as our feature set to find the nearest neighbors of an image b's embedding vector, using their Euclidean distances. These nearest neighbors' distances z are then normalized to result in probabilities z'' in a range of 0 (low) to 1 (high), such that

$$z'' = 1 - \frac{z'}{max(z', z_{max})}\ ,\ z' = [z - min(z, z_{min})] \quad (10)$$

where z_{min} is the lowest minimum distance found in z and z_{max} is the highest maximum distance found in z'. These values are found by comparing various results of the nearest neighbors' calculations for different iterations of the model. For our embedding experiments, we found z_{min} to be 0.8 and z_{max} to be 0.9.

In the same way as for classification, we add a probability of 0.5, that image x represents a ‘new_whale’, to the outputs of every image during the training phase. These outputs are then used for loss calculations. The only difference in the loss function used for embedding is that it does not include the binary cross-entropy loss. The loss used in this case is just L_T, Eq. 6. Finally, the individuals representing the closest, or most probable, image vectors are then chosen, as the system’s predictions, for accuracy calculations.

When approaching this problem as one of high-dimensional embedding, using a chosen probability for the ‘new_whale’ class, instead of it being predicted by the model, makes more intuitive sense. In our embedding space, the goal is that feature vectors representing the same individual should be close to each other. However, a ‘new_whale’ does not represent a single individual but, more specifically, any individual which is not part of the known whales. As the embedded vectors representing these varying, unknown, ‘new_whale’ individuals would probably be located far from each other in our embedding space, it does not make much sense to treat ‘new_whale’ as just another class to predict.

3.2.5 Parameters

The batch size used when loading the training data during classification was 12. It is important to note that a batch size of 12 for this system represents how many anchors were selected for each batch. The size of the mini-batch was actually 48 or 36, depending on whether ‘new_whale’s were included or not. While training using the embedding approach, batch size was doubled to 24 to accelerate the training of the network. This also required the usage of 6 GPUs instead of the 2 used for training the classifier. The model used an Adam optimizer with a learning rate of $3x10^{-4}$ and the model input size was 256x512.

4. EXPERIMENTS AND RESULTS

There are three main findings we want to highlight, each further discussed in their appropriate section. These are: how varying the minimum number of encounters impacted the training process, investigating how adding the ability to recognize 'new_whale's changed the network's performance, and an analysis of the tradeoffs between the use of a classifier vs. an embedding approach as our identification method. The data used for the experiments described in section 4.1 did not yet include 'new_whale' individuals. Unless otherwise stated, these networks were trained with the parameters described in the methods section.

4.1 Number of Encounters per Individual

The first results of interest analyze the impact of the number of samples on the classifier's accuracy on the training set. We found through experimentation that a low amount of training data, when $m = 3$, per individual would lead to a wide gap between the training and validation accuracies. This indicated the model had a possible overfitting issue. m being equal to 3 indicates that there was a minimum of 3 encounters for individuals to be added to the validation and testing set. Because there needed to be at least 1 image in validation and 1 in testing, this only left 1 (u) image to be put in training when $m = 3$. In the case of $m = 3$, u is equal to 1 and represents the minimum number of encounters per individual in the training set, where

$$u \leq m - 2 \tag{11}$$

In the case of $u = 1$, we believe the large accuracy gap happened because the model would memorize the single image of an individual that it would be given and would not learn to properly generalize. To help with this, we experimented with using a greater u, and thus a greater m, with

the assumption that increasing the minimum number of encounters per individual in the training set would help bridge this gap. The results for these experiments can be seen in Table 1.

Table 1: The effects of varying the minimum number of encounters per individual in the training set (u) on the training and validation accuracies. As u increases, the gap between the accuracies decreases, but so do the amounts of images and number of individuals available.

Min. # of enc./ind. in train (u)	Testing top-1 acc.	Validation top-1 acc.	Training top-1 acc.	# of images (train, valid, test)	# of individuals	Time taken to train (hrs)
1	0.712	0.762	0.987	(2171, 905, 839)	839	3.95
2	0.801	0.834	0.987	(1663, 397, 331)	331	1.23
4	0.794	0.863	0.933	(1216, 197, 131)	131	0.88

As expected, increasing u did result in a decreasing gap between the training and validation accuracies. This, however, did come at a cost. Increasing u also meant that there was less overall data that could be included. A higher u resulted in a decrease of both the amount of training and validation images available but also the total number of individuals represented by these images. Therefore, higher choices for u and m were limited by the data we had available. In the end, a minimum m of 4 and a u of 2 were picked for the rest of the experiments as this resulted in a better accuracy gap while still allowing for a good amount of data.

4.2 The Effects of ‘new_whale’s and More Data

Our network’s ability to recognize ‘new_whale’s was a critical part of this thesis. We looked at how the validation accuracy was impacted by the type and amount of data we included in the training set. We separate the results from before and after including the ‘new_whale’ individuals to get an idea of how this class affects the network’s performance. We also decided to experiment on the effect of including all individuals in the training set, even if some of those individuals were not found in the validation and testing set, as it increased the amount of training data available. These additional individuals included in the training set were those that would have otherwise been excluded from all sets because they had less than m encounters.

Table 2: The effects of adding ‘new_whale’s and additional training data on validation accuracy. Adding ‘new_whale’ individuals results in a drop in validation accuracy but this was negated by the addition of extra data to the training set.

Version # and description	Testing known top-1 acc.	Validation top-1 acc.	Training top-1 acc.	# of images (train, valid, test)	# of individuals (train, valid/test)	Time taken to train (hrs)
1. Without ‘new_whale’s	0.801	0.834	0.987	(1663, 397, 331)	(331, 331)	1.23
2. With ‘new_whale’s	0.683	0.803	0.957	(3522, 604, 331)	(332, 332)	0.75
3. With ‘new_whale’s and extra data	0.760	0.838	0.990	(4774, 500, 707)	(840, 332)	3.02

Detailed in Table 2 are the results for this set of experiments. The result labeled version 1 was the one chosen from the previous set of experiments in section 4.1, the one with $u = 2$. Version 2 results are for when we included the 'new_whale' ID and images in the training and validation sets. This addition resulted in a drop in performance but greatly increased the applicability of this network. The network could now reliably identify which individuals it had not previously seen instead of trying to classify them all as something it knew.

As for version 3, including all the data that does not satisfy the requirements of m in the training set could potentially lead to a u of 1, while m would still be 4. This did not happen in our case because there were no individuals with 1 encounter as they were all converted to the 'new_whale' ID. Version 3 results show that including more data resulted in a better top-1 testing and validation accuracy when compared to version 2, as shown in Table 2, but it also required significantly more time to train with this new data. The configuration described for version 3 was used for the rest of the experiments.

4.3 Classification vs. Embedding

Our analysis of the tradeoff between classification and embedding focuses on the testing accuracies achieved as well as the time taken to train the corresponding models. As seen in Table 3, training the classifier took about 3 hours and resulted in a top-1 testing accuracy of 83.0%. This matches what was expected from the validation accuracy. Training the model for embedding was a bit trickier.

Table 3: Testing accuracies of classification vs. embedding. While the embedding approach resulted in a big decrease in the achieved validation accuracy, it performed remarkably well during testing; with comparable results to the classification approach.

Approach	Testing top-1 acc. (known, 'new_whale')	Testing top-5 acc. (known, 'new_whale')	Testing top-10 acc. (known, 'new_whale')	Validation top-1 acc.	Training top-1 acc.	Time taken to train (hrs)
Classification	0.830 (0.760, 0.960)	0.895 (0.839, 1.000)	0.902 (0.850, 1.000)	0.838	0.990	3.02
Embedding	0.805 (0.756, 0.895)	0.880 (0.821, 0.988)	0.907 (0.856, 1.000)	0.704	0.917	3.02

Training for embedding from scratch was very slow. This was because, for each iteration, we have to generate the embeddings for all the training data to get the updated distance measures for our loss, training accuracy and validation accuracy. Training from scratch also did not increase validation accuracy significantly between iterations. Therefore, to help jumpstart the model's learning, we decided to use the model weights pretrained for classification as a starting point when training for embedding.

It was very helpful for embedding to have the pretrained model be used when changing to the embedding task, as seen in Table 3. The model started off with a pretty high accuracy and its performance did not significantly change with further training. In other words, the model weights and embedded features acquired while training the classifier performed the best for embedding without any further training. With this model state, the model did comparably to the classifier on the testing set. The embedding approach scored a top-1 accuracy of 80.5%. When looking at the accuracies split by known vs 'new_whale', we can see that the embedding approach did a little

worse than the classifier on the ‘new_whale’ category. Both approaches, however, scored significantly better on the ‘new_whale’ class compared to the known individuals.

For the results shown in Table 3, the thresholds t used by the classifier and when using the embedding approach were both 0.3. The lower the threshold used, the better the networks have learned to recognize individuals. This was a good sign that both approaches learned to effectively represent the individuals in our humpback dataset.

4.4 Embedding and ‘new_whale’s

While the previous experiments showed that the embedding model could give comparable results to the classifier, one more experiment was necessary to prove one of the biggest potential benefits of the embedding approach. Using an embedding approach gives us a guarantee in terms of our embedding function. If the embedding function works well and can properly represent a given individual in an embedding space, there should be no restriction on this individual to be one that has been previously seen by the model. Testing the validity of this statement is of great importance. To this end, we explore in Table 4 the ability of our embedding model to recognize whales that were added to the dataset outside of the training set.

Table 4: Testing accuracies of embedding on old vs. new individuals. The embedding approach resulted in testing accuracies on newly added individuals comparable to those on individuals the model was trained on.

Individuals trained on?	Testing known top-1 acc.	Testing known top-5 acc.	Testing known top-10 acc.	Individuals tested	Time taken to train (hrs)
Yes	0.817	0.858	0.880	331	3.97
No	0.765	0.865	0.885	200	3.97

With no additional training, new individuals were successfully added to the feature set of known individuals for our embedding model. Furthermore, for each of these individuals, only 1 encounter was used to add them to our set of known individuals. This is compared to the minimum u of 2 used during training. These new individuals' embedding vectors were acquired as results from the model and they were added to our 2D matrix of existing, known embeddings. Testing on just these newly added individuals resulted in a 76.5% top-1 accuracy. This is comparable to the 81.7% top-1 accuracy achieved by the same model on individuals it had already encountered in the training set. When comparing the top-5 and top-10 results, this gap is even smaller.

5. CONCLUSION

In this thesis, we compared two different approaches for the identification of humpback whales. We showed that our embedding approach performed well compared to the classification approach while allowing for the addition of new individuals without the need for retraining. We also experimented on deciding on an appropriate minimum number of encounters per individual in the testing set and investigated how the network's performance was affected by attempting to recognize 'new_whale's. The ability to recognize novel individuals is important and we show that, with enough data, achieving this is not a concern.

As can be seen in Table 3, recognizing 'new_whale's is something that both the classification and embedding approaches have shown they can do well. The fact that the models do better on recognizing 'new_whale's vs. known whales could be used to an ecologist's or researcher's advantage. A variant of the model could more reliably be used to just recognize whether an individual is already part of the training set. The model could also be used to recognize 'new_whale's while giving a list of probable individuals that an ecologist would then have to make the final decision on. If used this way, both the classifier's and the embedding's top 5, top 10, or top 20 results can be used by a trained ecologist for achieving higher accuracies in a reasonable timeframe than either could separately. Both the classification and embedding approaches resulted in a 90% top-10 accuracy as shown in Table 3. Making the top 10 possible individuals available and pairing them with the model's prediction of the image's probability to belong to those classes can greatly reduce the effort needed by ecologists for identification while still retaining the benefits the model provides.

Approaching the problem of identification as an embedding problem changes the nature of the problem itself. Instead of treating identification as a problem of reducing an image to a set of probabilities, extracting an embedding is a problem of accurately representing an image in a metric space. We show that an embedding-based approach can lead to comparable results to a classification-based one for the identification of humpback whales. Switching to embedding resulted in a lower validation accuracy score on the same data. However, the model did about as well as the classifier on the testing set. This shows that choosing an embedding-based approach could be advantageous if it were to bring other benefits as well.

There is an important benefit to using an embedding approach for identification that is not present with a classification approach. If there is a need to expand the dataset, an embedding approach allows the model to learn of new entries immediately, no retraining needed, as shown in Table 4. This can really be helpful in cases where training takes many hours and there are often new entries. It is also worth considering, if attempting multi-way classification, that not retraining the model helps limit the size of the output as the number of classes increases. So, in theory, there is no worry about tackling datasets with thousands of classes when using an embedding model, as the dimensionality of the embedding vector would not need to change. The validity of this, however, would need to be validated with appropriate data and experiments. But to further emphasize the benefits of this point, let's consider what the choice of m implies for adding new individuals to the set of known individuals.

m was chosen to be 4 for this thesis. To maximize the accuracy of the system in predicting an image's class, it is best to require every individual in the validation and testing sets to have at least $m - 2$ images in the training set. This requirement can be a concern for adding new individuals to the set of known individuals when using a classification-based approach. To retrain

the classifier with the goal of expanding the dataset, for every individual to be added, one would need to have at least $u = m - 2$ images for training and 2 more images if it were to be validated and tested as well. In the case of embedding however, this requirement would not be necessary as using the embedding function on a new individual does not require any retraining.

When using embeddings, whatever the value of m might be, individuals with only 1 encounter could just be added to the set of known individuals as all that matters is that we know that the embedding function works. This advantage of the embedding-based approach is also showcased in Table 4. Since $u = 2$ for our training process, at least 2 images would be needed when training the classifier to meet this requirement. However, the embedding-based approach does not have this requirement and so just one new feature vector, of an individual that was not trained on, can be used to recognize another individual with the same ID.

Lastly, it is worth discussing the choice of some parameters used throughout these experiments. In terms of the applying the approaches discussed in this thesis to other datasets, there are some modifications that can be done to the parameters mentioned in the methods and results sections. Particularly, the value of m can significantly vary. Increasing m will likely lead to higher accuracies, if there is enough training data, but at the cost of training data, as shown in Table 1. This decision can be made according to the size of the dataset one has access to and to the number of encounters per individuals found within the dataset and expected in the setting the system will be used in. It might also be a good idea to change the proportion of ‘new_whale’ to known images in the training and validation set. The proportion used in this thesis was about 1:2 but this might greatly differ, in practice, depending on the setting, data available, and expected use. If one does not expect to see such a high percentage of 'new_whale’s when the network is used, the proportion could certainly be decreased for a potential boost to accuracy.

REFERENCES

[1] C. J. Krebs, *Ecological Methodology*, 2nd ed. Menlo Park, CA, USA: Benjamin/Cummings, 1999.

[2] R. S. Alonso, B. T. McClintock, L. M. Lyren, E. E. Boydston, and K. R. Crooks, "Mark-recapture and mark-resight methods for estimating abundance with remote cameras: a carnivore case study," *PLOS ONE*, vol. 10, no. 3, Mar. 2015, Art no. e0123032.

[3] L. Silveira, A. T. Jacomo, and J. A. F. Diniz-Filho, "Camera trap, line transect census and track surveys: a comparative evaluation," *Biol. Conservation*, vol. 114, no. 3, pp. 351-355, Dec. 2003.

[4] L. N. Rich *et al.*, "Comparing capture-recapture, mark-resight, and spatial mark-resight models for estimating puma densities via camera traps," *J. of Mammalogy*, vol. 95, no. 2, pp. 382-391, Apr. 2014.

[5] S. C. Silver *et al.*, "The use of camera traps for estimating jaguar Panthera onca abundance and density using capture/recapture analysis," *Oryx*, vol. 38, no. 2, pp. 148-154, Apr. 2004.

[6] A. F. O'Connell, J. D. Nichols, and K. U. Karanth, *Camera Traps in Animal Ecology: Methods and Analyses*. Tokyo, Japan: Springer Japan, 2010.

[7] K. U. Karanth and J. D. Nichols, "Estimation of tiger densities in India using photographic captures and recaptures," *Ecology*, vol. 79, no. 8, pp. 2852-2862, Dec. 1998.

[8] R. J. Foster and B. J. Harmsen, "A critique of density estimation from camera-trap data," *J. Wildlife Manage.*, vol. 76, no. 2, pp. 224-236, Feb. 2012, doi: 10.1002/jwmg.275.

[9] P. D. Meek, K. Vernes, and G. Falzon, "On the reliability of expert identification of small-medium sized mammals from camera trap photos," *Wildlife Biol. in Pract.*, vol. 9, no. 2, pp. 1-19, Dec. 2013, doi: 10.2461/wbp.2013.9.4.

[10] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bull. of Math. Biophys.*, vol. 5, no. 4, pp. 115-133, Dec. 1943.

[11] A. Rinciog, C. Mieth, P. M. Scheikl, and A. Meyer, "Sheet-metal production scheduling using AlphaGo Zero," in *Proc. 1st Conf. Prod. Syst. and Logistics (CPSL)*, 2020, pp. 342-352, doi: 10.15488/9676.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2014, pp. 580-587.

[13] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances Neural Inf. Process. Syst. 27 (NIPS)*, 2014, pp. 568-576.

[14] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," presented at the Eur. Conf. Comput. Vision (ECCV), Amsterdam, The Netherlands, Oct. 8-16, 2016.

[15] *Kaggle Humpback Whale Identification Challenge 1st place code*. (2019). Accessed: October 24, 2019. [Online]. Available: https://github.com/earhian/Humpback-Whale-Identification-1st-

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2015, pp. 815-823.

[17] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2016, pp. 1335-1344.

[18] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, arXiv:1703.07737.

[19] H. J. Weideman *et al.*, "Integral curvature representation and matching algorithms for identification of dolphins and whales," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2831-2839.

[20] S. McCann and D. G. Lowe, "Local naive bayes nearest neighbor for image classification," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2012, pp. 3650-3656.

[21] M. Körschens, B. Barz, and J. Denzler, "Towards automatic identification of elephants in the wild," 2018, arXiv:1812.04418.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2016, pp. 779-788.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.

[24] S. Schneider, G. W. Taylor, and S. C. Kremer, "Similarity learning networks for animal individual re-identification-beyond the capabilities of a human observer," in *Proc. IEEE/CVF Winter Conf. Appl. of Comput. Vision (WACV) Workshops*, 2020, pp. 44-52.

[25] O. Moskvyak, F. Maire, A. O. Armstrong, F. Dayoub, and M. Baktashmotlagh, "Robust re-identification of manta rays from natural markings by learning pose invariant embeddings," 2019, arXiv:1902.10847.

[26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2018, pp. 7132-7141.

[27] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, 2018, pp. 480-496.