

# **Towards Automatic Photo-Identification of Cetaceans: A Fine-Grained, Few-Shot Problem in Marine Ecology**



**Cameron Trotter**

School of Electrical and Electronic Engineering  
Newcastle University

In Partial Fulfilment of the Requirements for the Degree of  
*Doctor of Philosophy*

January 2023



## Acknowledgements

Thank you to my supervisors Nick Wright, Stephen McGough, and Per Berggren for their continued support and expert advice throughout my PhD.

Thank you to Paul Watson and Darren Wilkinson for giving me the opportunity to pursue a PhD through the Cloud Computing for Big Data CDT.

Thank you to Matthew Sharpe, Georgia Atkinson, and Kirsten Crane for your aid during fieldwork and understanding the frustrations of working with messy ecology data.

Thank you to Barbara Cheney (University of Aberdeen), Mònica Arso Civil (University of St Andrews), and Reny Tyson Moore and Jason Allen (Chicago Zoological Society's Sarasota Dolphin Research Program) for providing additional photo-id data for use in this thesis.

Thank you to the citizen scientists of both the Newbiggin-by-the-Sea Dolphin Watch and the North East Cetacean Project. Your dedication was vital to the success of my fieldwork.

Thank you to Jennifer Wood and Andrew Turnbull for your indispensable aid in navigating the endless minefield that is University administration.

Thank you to those who have provided engaging discussions (and distractions) throughout my PhD, including but not limited to: Tom Owen, Tom Lowe, Kieran Peel, Jack Aiston, Dale Whinham, Phillip Darke, and Carlos Vladimiro González Zelaya.

Finally, thank you to my parents, Paula and Norman Trotter. I would not have achieved what I have without you.

*This work was supported by the Engineering and Physical Sciences Research Council  
Centre for Doctoral Training in Cloud Computing for Big Data  
[EP/L015358/1]*



## Abstract

Understanding the health of Earth's ecosystems is imperative for the future safeguarding of our planet and its inhabitants. One of the most common tools utilised by researchers to develop their understanding of an area's health is indicator species, organisms whose abundance or absence in a system reflects overall environmental health. Cetaceans such as dolphins, porpoises, and odontocetes (toothed whales) are excellent indicator species given their status as top predators, allowing for the monitoring of risks to marine environments, such as offshore wind farm development or commercial fishing activity.

Cetacean monitoring is frequently performed using capture-recapture surveys, through which researchers record the presence of individual animals to produce population estimates. Photo-identification (photo-id) is one of the main non-invasive capture-recapture methods, whereby image data containing the animals' individually identifiable prominent markings are captured. Upon survey completion these data are curated to produce a photo-id catalogue, allowing for an abundance estimate to be generated and ecosystem health to be determined. Catalogues are updated over time as more surveys are undertaken, new individuals are encountered, and prominent markings change. Photo-id catalogue curation is traditionally performed manually and can be extremely labour and cost intensive, especially for large resident populations.

This thesis details a framework for automatic photo-id catalogue matching based on unprocessed field imagery via a pipeline of computer vision models. The development of a coarse-grained dorsal fin detector and the use of post-processing techniques to aid downstream identification is first examined. Next, the creation of a photo-id catalogue for cetaceans resident in the waters of Northumberland, UK, is outlined. This catalogue is then utilised to facilitate the development of a model capable of fine-grained, few-shot catalogue matching via latent space similarity, allowing for the flagging of potentially uncatalogued individuals. At all stages, the developed techniques' robustness to spatio-temporal changes is evaluated, including their generalisability to multiple cetacean species. The automation of photo-id data curation outlined in this thesis affords researchers more time to work on application of their data, for example to inform mitigation and policy change, rather than administration.



# Table of contents

<b>List of figures</b>	<b>xi</b>
<b>List of tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Aim and Contributions . . . . .	3
1.2 Thesis Structure . . . . .	4
1.3 Related Publications . . . . .	6
<b>2 Background</b>	<b>7</b>
2.1 A Brief Introduction to Photo-Identification . . . . .	7
2.2 Machine Learning: Supervised vs Unsupervised Approaches . . . . .	11
2.3 A Brief Introduction to Deep Learning . . . . .	12
2.3.1 Optimising Deep Learning Networks . . . . .	13
2.3.2 Backpropagation . . . . .	15
2.4 Deep Learning for Computer Vision . . . . .	16
2.4.1 Convolutional Neural Networks . . . . .	16
2.4.2 Object Detection . . . . .	21
2.4.3 Semantic Segmentation . . . . .	24
2.4.4 Part Segmentation . . . . .	26
2.4.5 Instance Segmentation . . . . .	27
2.4.6 Fine-Grained Visual Categorisation . . . . .	32
2.4.7 Extreme Fine-Grained Visual Categorisation . . . . .	32
2.5 Computer Vision for Ecology . . . . .	33
2.5.1 Utilising Photo-id Aids in Cetacean Research . . . . .	35
2.6 Summary . . . . .	43

<b>3 Initial Dataset Creation</b>	<b>45</b>
3.1 The Zanzibar Dataset . . . . .	46
3.2 The Northumberland Dolphin Dataset 2020 . . . . .	47
3.2.1 Data Collection . . . . .	48
3.2.2 The Survey Area . . . . .	48
3.2.3 Survey Effort . . . . .	48
3.2.4 Field Season Summary . . . . .	51
3.3 Creation of NDD20 . . . . .	51
3.3.1 Above Water Data . . . . .	51
3.3.2 Below Water Data . . . . .	52
3.3.3 NDD20 Summary . . . . .	54
3.4 Summary . . . . .	56
<b>4 Cetacean Detection Using Deep Learning</b>	<b>59</b>
4.1 Requirements of a Cetacean Detector . . . . .	59
4.1.1 Environmental Requirements . . . . .	59
4.1.2 Technical Requirements . . . . .	62
4.2 Deciding on Approach . . . . .	63
4.2.1 An Investigation into Bounding Boxes . . . . .	64
4.2.2 Instance Segmentation Architectures . . . . .	67
4.3 Initial Testing of Mask R-CNN . . . . .	68
4.3.1 Transfer Learning . . . . .	69
4.3.2 Utilising Transfer Learning to Train the Mask R-CNN . . . . .	70
4.3.3 Data Augmentation . . . . .	71
4.4 Mask R-CNN Model Selection . . . . .	73
4.4.1 Detection Hyperparameters . . . . .	73
4.4.2 Training Hyperparameters . . . . .	73
4.4.3 Hyperparameter Tuning via Grid Search . . . . .	76
4.4.4 Model Selection Based on Grid Search . . . . .	78
4.4.5 An Evaluation of Optimal Model Hyperparameters . . . . .	80
4.4.6 Limitations of the Model . . . . .	80
4.5 Model Evaluation Using NDD20 . . . . .	82
4.5.1 Evaluating the Effect of Geography, Time, and Species Change . .	82
4.5.2 Below Water Detection Baseline . . . . .	83
4.6 Summary . . . . .	84

<b>5 Mask Post-Processing Techniques</b>	<b>87</b>
5.1 Handling Multiple Detections . . . . .	87
5.2 Morphological Transformations . . . . .	89
5.3 Background Subtraction . . . . .	89
5.4 Colour Thresholding Mask Components . . . . .	90
5.5 Cropping . . . . .	93
5.6 Post-Processing NDD20 . . . . .	96
5.6.1 Additional External Data . . . . .	97
5.7 Summary . . . . .	99
<b>6 Individual Cetacean ID via Automatic Most Likely Catalogue Matching</b>	<b>101</b>
6.1 Most Likely Catalogue Matching System Requirements . . . . .	101
6.2 Siamese Neural Networks . . . . .	103
6.2.1 Clustering Embeddings in a Latent Space . . . . .	104
6.2.2 Meeting the Outlined Requirements . . . . .	105
6.2.3 Pairwise vs Triplet Ranking Loss . . . . .	106
6.2.4 Semi-Hard Triplet Mining . . . . .	109
6.2.5 Class Prototyping . . . . .	110
6.2.6 Top-N Accuracy . . . . .	112
6.3 Siamese Neural Network Development . . . . .	113
6.3.1 Hyperparameter Tuning Via Bayesian Optimisation . . . . .	114
6.3.2 Data Augmentation Strategy . . . . .	115
6.4 Siamese Neural Network Model Selection . . . . .	116
6.4.1 An Evaluation of Optimal Model Hyperparameters . . . . .	117
6.4.2 NDD AU SMRU Uncatalogued Individual Thresholding . . . . .	118
6.4.3 Limitations of the Model . . . . .	122
6.5 Per-Side Identification . . . . .	123
6.5.1 Individual Only Top-N Accuracy . . . . .	125
6.6 Summary . . . . .	126
<b>7 Evaluating Siamese Neural Network Catalogue Matching Robustness</b>	<b>129</b>
7.1 Effect of AU SMRU Data on Model Performance . . . . .	129
7.2 Effect of Background on Embedding Generation . . . . .	130
7.3 Examining Siamese Neural Network Catalogue Matching Generalisability	132
7.3.1 The SDRP Dataset . . . . .	132
7.3.2 Evaluation Using the SDRP Dataset . . . . .	134
7.3.3 SDRP Uncatalogued Individual Thresholding . . . . .	136

7.3.4	Evaluating the NDD AU SMRU SNN Using the SDRP Dataset . . . . .	139
7.3.5	Effect of Training Set Size on Model Backbone Selection . . . . .	139
7.4	Considering Catalogue Matching as a Standard Classification Task . . . . .	141
7.5	Summary . . . . .	144
<b>8</b>	<b>Conclusion</b>	<b>145</b>
8.1	Thesis Summary . . . . .	145
8.2	Evaluation Against Thesis Aims . . . . .	148
8.3	Future Research Directions . . . . .	149
8.3.1	Effect of Initial Catalogue Size on Most Likely Matching . . . . .	149
8.3.2	Effect of Background Over Large Temporal Scales . . . . .	150
8.3.3	Further Per-Side Experimentation . . . . .	151
8.3.4	Creating a Generalised Model for Catalogue Matching . . . . .	151
8.3.5	Verification of Dataset Ground Truth Labels . . . . .	152
8.3.6	Expansion to Other Data Subjects . . . . .	152
8.3.7	Transforming the Model Pipeline . . . . .	153
8.3.8	The Creation of a Working Computer Program . . . . .	154
<b>Appendix</b>		<b>155</b>
A	mAP@IOU[0.5:0.95] Scores for Mask R-CNN Grid Search Models . . . . .	155
B	Data Collection Camera Settings . . . . .	156
C	NDD20 Above Water Example Images Class Labels . . . . .	157
D	NDD20 Below Water Example Images Class Labels . . . . .	158
E	Optimal Hyperparameters for Per-Side NDD AU SMRU Identification . . . . .	159
F	Optimal Hyperparameters for Standard Classification Task Architectures . . . . .	159
<b>Bibliography</b>		<b>161</b>

# List of figures

1.1	A high level overview of data flow through the proposed system. . . . .	3
2.1	Examples of the main body parts utilised in cetacean photo-id. . . . .	9
2.2	Examples of prominent markings used during dolphin photo-identification surveys. . . . .	9
2.3	A visual representation of convolution. . . . .	17
2.4	A kernel, represented by the grey squares, operates over a padded input matrix, blue, with a stride of 2 to produce an output feature map, green. . .	18
2.5	A visual representation of the final stages of a classification network. . . .	20
2.6	The R-CNN pipeline. . . . .	22
2.7	An example of RoIs generated on an image by an RPN, showing 10 random proposals. . . . .	22
2.8	Left: an example input image to a semantic segmentation model. Right: the resultant semantic segmentation mask for the given input image. . . . .	24
2.9	Generated anchors. . . . .	25
2.10	An example of refined anchors. . . . .	26
2.11	Left: an example of some input image. Right: the corresponding dolphin ground truth instance segmentation masks. . . . .	27
2.12	The Faster R-CNN architecture. . . . .	31
2.13	A visual representation of the changes made to stage two of Faster R-CNN to create Mask R-CNN. . . . .	31
2.14	An example image from a photo-id catalogue with one individual present. .	33
3.1	An example of the labelling process using VIA. . . . .	47
3.2	Map of the survey area, with the Coquet to St. Mary's MCZ highlighted. .	49
3.3	Example above water images from NDD20 with filenames displayed. . .	53
3.4	Example below water images from NDD20 with filenames displayed. . .	54
3.5	The ID class label distribution for the above water set of NDD20. . . . .	55

3.6	The ID class label distribution for the below water set of NDD20. . . . .	55
4.1	Some cetaceans, such as bottlenose dolphins, travel in pods. The developed detection system must be capable of splitting this pod into individual animals to be passed to the identifier. . . . .	60
4.2	Two images of the same individual taken from different angles of approach, directions of travel, and distances from the vessel. . . . .	61
4.3	Left: an example input image. Right: the corresponding manual crop utilised in bounding box suitability testing. . . . .	64
4.4	Left: an example input image. Right: the corresponding manual crop with the result of SURF feature extraction overlaid. . . . .	65
4.5	Left: an example input image. Right: the corresponding manual crop with the result of SURF feature extraction when thresholded based on RGB colour values overlaid. . . . .	66
4.6	Left: an example input image. Right: the corresponding manual crop with the result of GrabCut background removal overlaid. . . . .	67
4.7	The high level pipeline overview, shown in Figure 1.1, with the Mask R-CNN component highlighted. . . . .	68
4.8	Examples of data augmentations found in the literature. . . . .	71
4.9	An example image showing the effect of changing the Mask R-CNN’s minimum detection confidence. . . . .	74
4.10	Left: the input image passed to the cetacean detector. Right: the detection masks produced by the model overlaid onto the image. . . . .	81
4.11	Left: the image passed to the cetacean detector. Right: the detection masks produced by the model overlaid onto the image. . . . .	81
4.12	Left: the image passed to the cetacean detector. Right: the detection masks produced by the model overlaid onto the image. . . . .	82
5.1	The high level pipeline overview, shown in Figure 1.1, with the post-processing techniques highlighted. . . . .	87
5.2	A visualisation of the three dolphin detections for an input image produced by the Mask R-CNN detector. . . . .	88
5.3	Left: a detection mask before closing has been applied. Right: the same detection mask after closing. . . . .	89
5.4	A visualisation of the three dolphin detections from Figure 5.2 before and after background subtraction. . . . .	90

5.5	Top Left: a visualisation of the dolphin detection for an input image produced by the Mask R-CNN detector, alongside the confidence score. Top Right: the resultant detection mask after morphological transformation. Bottom: the resultant output image after <i>bitwise and</i> operations performed between the input image and the cleaned detection mask. . . . .	91
5.6	The global range of pixel intensities for each RGB colour channel, split by pixel classification. . . . .	92
5.7	Workflow detailing colour thresholding to remove an area of disjoint splash which has been detected as part of a dolphin object. . . . .	94
5.8	Workflow showing how utilising the 90% global threshold may lead to over-exposed detections being erroneously discarded. . . . .	95
5.9	Left: an input image and overlaid detection mask. Right: the corresponding cropped output image after post-processing. . . . .	95
5.10	Example images removed when post-processing NDD20 as their identity could not be verified manually. . . . .	97
5.11	Example images from Segmented NDD20. . . . .	98
6.1	Left: An example input image. Right: The corresponding post-processed crop which contains some misclassified noise. . . . .	103
6.2	An example two-branch SNN architecture for signature verification. . . .	104
6.3	A 2-dimensional visualisation of a multi-dimensional latent space produced by an SNN trained on the MNIST dataset for 100 epochs. . . . .	105
6.4	SNN optimisation using Pairwise Ranking Loss. . . . .	107
6.5	SNN optimisation using Triplet Ranking Loss. . . . .	108
6.6	A visualisation of the areas in the latent space where Easy, Hard, and Semi-Hard triplets can occur. . . . .	110
6.7	An example latent space with two classes (cross and square) alongside a triangle which represents the embedding location of an unclassified inference image. The two class examples selected for distance measurement to classify the triangle using the naive approach are circled. . . . .	111
6.8	An example latent space with two classes (cross and square) alongside a triangle which represents the embedding location of an unclassified inference image. The two class prototypes used to classify the triangle, $P_x$ and $P_{\blacksquare}$ respectively, are circled. . . . .	112
6.9	The high level pipeline overview, shown in Figure 1.1, with the SNN component highlighted. . . . .	113
6.10	Example uncatalogued individual thresholding. . . . .	120

6.11	Example noise thresholding.	121
7.1	Example data used to examine the effect of retained background on most likely catalogue matching.	131
7.2	Example images from the SDRP dataset with filenames displayed.	133
7.3	Left: an image of individual 13 from the original SDRP catalogue set. Right: example masks generated for the Left image.	134
7.4	The class distribution for the SDRP dataset, split by set.	135
7.5	Example uncatalogued individual thresholding for the SDRP dataset using an individual not present during training.	137
7.6	Example uncatalogued individual thresholding for the SDRP dataset using an individual present during training.	138
7.7	Importance scores for the hyperparameter tuning of the ResNet50 model.	143
7.8	Importance scores for the hyperparameter tuning of the VGG16 model.	143

# List of tables

2.1	A comparison of available photo-id aids. . . . .	36
3.1	A comparison of computer vision datasets capable of training models for individual animal identification, sorted by number of individuals. . . . .	56
4.1	Hyperparameter values and mAP@IOU[0.5, 0.75, 0.85] scores used for each Mask R-CNN grid search model trained on the Zanzibar data. . . . .	77
4.2	Hyperparameters of the best performing Mask R-CNN models on the Zanzibar dataset. . . . .	79
4.3	mAP@IOU[0.5:0.95] scores for model 20190902T0946, the best performing Mask R-CNN dorsal fin detector. . . . .	83
4.4	mAP@IOU[0.5:0.95] scores for the best performing Mask R-CNN model trained using the below water set of NDD20. . . . .	84
5.1	A comparison between the NDD20, Segmented NDD20, and NDD AU SMRU datasets. . . . .	99
6.1	Results of SNN training for the task of most likely catalogue matching on the NDD AU SMRU dataset. . . . .	117
6.2	Optimal SNN hyperparameters for each architecture-augmentation combination located using Bayesian optimisation over 100 iterations. . . . .	118
6.3	Results of SNN training for the task of most likely catalogue matching on the NDD AU SMRU dataset after per-side splitting. . . . .	124
6.4	Comparison between the best performing SNN models for the task of most likely catalogue matching trained on the combined-side and per-side NDD AU SMRU datasets. . . . .	124
6.5	Results of SNN training for the task of most likely catalogue matching on the NDD AU SMRU dataset after per-side splitting, utilising the modified Top- $N$ accuracy measurement to only take into account the individual classification.	125

6.6	Comparison between the best performing SNN models for the task of most likely catalogue matching trained on the combined-side and per-side NDD AU SMRU datasets, evaluated against the modified top- $N$ accuracy metric. . . . .	126
7.1	Comparison between two SNNs trained for most likely catalogue matching, one using the Segmented NDD20 17 dataset, and the other using the NDD AU SMRU 17 dataset. . . . .	130
7.2	Comparison of the Top-1, Top-5, and Top-10 accuracies between training an SNN on bounding boxed or masked photo-id data. . . . .	132
7.3	Results of SNN training for the task of most likely catalogue matching on the SDRP dataset. . . . .	135
7.4	Results of SNN training for the task of most likely catalogue matching on the reversed SDRP dataset. . . . .	140
7.5	Top- $N$ accuracies of the best performing image classification models on the NDD AU SMRU dataset. . . . .	142
A	mAP@IOU[0.5:0.95] scores for each Mask R-CNN model trained in the Zanzibar dataset grid search. . . . .	155
B	Camera settings used during data collection as outlined in Section 3.2.3. . . . .	156
C	Class labels for the images shown in Figure 3.3. . . . .	157
D	Class labels for the images shown in Figure 3.4. . . . .	158
E	Optimal SNN hyperparameters for the best performing SNN trained on the per-side NDD AU SMRU dataset located using Bayesian optimisation over 100 iterations. . . . .	159
F	Optimal SNN hyperparameters for the best performing models trained on the NDD AU SMRU dataset for the task of image classification. . . . .	159

# Nomenclature

## Acronyms/Abbreviations

CNN	Convolutional Neural Network .....	16
CPU	Central Processing Unit .....	13
FCN	Fully Convolutional Network .....	25
GPU	Graphical Processing Unit .....	13
GUI	Graphical User Interface .....	154
IOU	Intersection Over Union .....	24
KNN	K-Nearest Neighbours .....	119
MCZ	Marine Conservation Zone.....	48
MLOps	Machine Learning Operations .....	154
NDD20	Northumberland Dolphin Dataset 2020 .....	4
NLP	Natural Language Processing .....	153
PCA	Principal Component Analysis .....	37
ReLU	Rectified Linear Unit.....	21
RIB	Rigid Inflatable Boat .....	50
RoI	Region of Interest.....	4
RPN	Region Proposal Network.....	21
SDRP	Sarasota Dolphin Research Program.....	132

SGDR	Stochastic Gradient Descent with Restarts .....	14
SGD	Stochastic Gradient Descent .....	14
SIFT	Scale-Invariant Feature Transform .....	41
SMRU	Sea Mammal Research Unit .....	98
SNN	Siamese Neural Network .....	2
SURF	Speeded-Up Robust Features .....	39
TPE	Tree-structured Parzen Estimator .....	114
UAV	Unmanned Aerial Vehicle .....	35
VIA	VGG Image Annotator .....	46
ViT	Vision Transformer .....	153
VM	Virtual Machine .....	76

# Chapter 1

## Introduction

In recent years there has been a concerted effort to apply computer vision techniques to challenging problems which can have a positive societal impact [1–6]. A highly important area where computer vision can help is ecology [7]. One of the main goals of ecological research is to monitor animal populations in their distribution area, undertaking abundance estimates to inform policy change. This is most commonly performed using capture-recapture surveys where researchers identify the presence of individuals and estimate abundance of animals in an area to produce population estimates [8–13]. These surveys can be classified as invasive where animals are physically trapped, tagged, and released [14–16], or non-invasive where monitoring is performed passively such as via the collection of images – referred to as photo-id [17–21].

Photo-id is one of the main non-invasive capture-recapture methods utilised by cetacean (dolphin, porpoises, and whales) researchers [22, 23]. Surveys are usually undertaken from vessels at sea, although monitoring from coastlines or aircraft may also be utilised [24–26]. The methodology is employed for the monitoring of multiple cetacean species, with a range of studies demonstrating its efficacy [9, 10, 27, 28]. Outside of cetaceans, photo-id has found further use in the study of other marine life [19, 21] and terrestrial species [29, 30].

All non-invasive capture-recapture methodologies rely on the target species having some form of individually identifiable markings. Depending on the species, different parts of the body are the primary identifying feature; for dolphins this is usually the dorsal fin as this body part is most likely to be visible above the waterline [10, 31]. During photo-id surveys, researchers often focus on long lasting stable markers such as dorsal fin shape, notches, scarring, and pigmentation [32–34]. These markings can be difficult to capture in detail due to the free roaming nature of the animals causing high variances in angles of approach, direction of travel, distance from camera, and surfacing elevation. This is

exacerbated when dealing with cetacean species that travel in pods, making it difficult to distinguish the individuals present.

Marine photo-id, which necessitates the use of human-operated cameras generally on-board a vessel, can be extremely labour and cost intensive compared to on-land surveys, which rely on the use of camera traps placed in stationary locations to capture images when they detect movement. This setup is not possible at sea however, as movement inherent to these environments such as waves may cause the triggering of motion detecting camera traps, resulting in a high false positive capture rate. For camera trap devices where this rapid environmental change may not cause false triggers, the lack of stationary objects within the environment to attach these devices to can also be a barrier to their usage.

Upon survey completion, photo-id data must be analysed and individuals identified to produce a catalogue. Images collected during surveys are large in size and contain significant amounts of background noise, defined as any non-animal related nuisance such as splash, waves, and other vessels. Historically, curation of this data has been a manual process that often takes longer than the entire data collection period [35], further increasing labour and costs. As such, any techniques to speed up the cataloguing process would be welcomed both by researchers and their funding bodies, affording more time to work on the application of data, for example to inform mitigation and policy change, rather than curation.

This thesis details a system for fully automatic catalogue matching based on unprocessed photo-id imagery. This is achieved through a pipeline of trained computer vision models and robust post-processing techniques capable of automatic fin detection and most likely catalogue matching based on latent space similarity, a visualisation of which can be seen in Figure 1.1.

As photo-id surveys are not guaranteed to capture all individuals in a given geographic area, naive approaches such as training a simple image classifier on existing catalogue examples would not suffice as they are incapable of flagging previously uncatalogued individuals. First, the raw images are passed through a Mask R-CNN [36] dorsal fin detector, removing the need for manual data pre-processing. Detections are then post-processed ready for fine-grained, few-shot catalogue matching utilising a Siamese Neural Network (SNN) [37] trained using triplet loss [38] and online semi-hard triplet mining to create a latent space based on the provided catalogue. Matches are obtained using the Euclidean distances between an input and class prototypes, which represent generalised embeddings generated from the examples for each class. This also allows for the flagging of potentially uncatalogued individuals to the researcher, identified as samples in the embedding space far from the known classes.

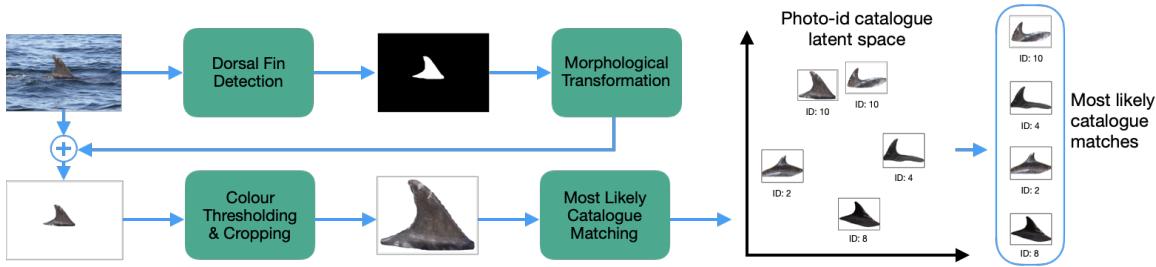


Figure 1.1 A high level overview of data flow through the proposed system. Input images (top left) are passed through a dorsal fin detection model, producing a binary dolphin output mask, morphologically transformed to remove erroneous holes. The transformed mask is then combined with the input image using a *bitwise and* operation to remove background. The resultant image is then colour thresholded and cropped, before being passed to a most likely catalogue matching model which reduces the image down to a lower dimensional embedding, which can be plotted into a latent space and compared to other previously embedded images to produce a list of most likely catalogue matches.

## 1.1 Research Aim and Contributions

The high level aim of this thesis is to:

**Design, implement, and evaluate a system for fully automatic catalogue matching based on unprocessed photo-id fieldwork imagery.**

Implicit in this aim are a number of research questions:

1. Is it possible to remove or greatly reduce the need for manual pre-processing of photo-id data in a fully automated way through the use of coarse-grained detection?
2. Can post-processing of detections be performed in such a way as to both reject likely false positives and remove noise whilst retaining identifiable markings?
3. Is it possible to perform highly accurate most likely photo-id catalogue matching based on extreme fine-grained information, even when operating on few-shot data?
4. How generalisable are the above concepts to changes in species of interest and spatio-temporal shifts?

To answer these questions and thereby address its aim, this thesis presents the following contributions:

1. A detailed survey on the current state of the art in applied computer vision for cetacean ecology. Existing photo-id aides are described and evaluated, providing a framework to understand their shortcomings. This is presented in Chapter 2.
2. An open-source dataset, called The Northumberland Dolphin Dataset 2020 (NDD20), for use in the evaluation of photo-id aids. Curated from imagery collected during fieldwork undertaken off the coast of Northumberland, UK, this dataset is released publicly to aid in the development of future photo-id aides, providing open-source data to a research area where this is not yet commonplace. This is presented in Chapter 3.
3. A highly accurate Mask R-CNN [36] based coarse-grained cetacean detector, capable of generating pixel-wise mask predictions from above water photo-id survey imagery. The detector is invariant to both the extreme variation in Region of Interest (RoI) shape inherently present due to the free roaming nature of cetaceans, and the large amount of noise generated as a result of operating in marine environments. This is presented in Chapter 4.
4. A framework for the robust post-processing of predicted RoIs produced by the cetacean detector with the aim of preventing false-positives from moving downstream. Predictions are automatically cleaned, allowing for the removal of noise and the enhancement of the prominence of individually identifying markings, thus removing the need for manual data pre-processing. This is presented in Chapter 5.
5. An SNN based computer vision model capable of fine-grained, few-shot catalogue matching using the post-processed detections passed downstream by previous system components. Inputs to the model are embedded into a latent space to generate a list of most likely catalogue matches using class prototypes and Euclidean distance measurements, allowing the model to flag potentially previously uncatalogued individuals to the user. Implementation work is presented in Chapter 6, with generalisability evaluation using a second real-life photo-id catalogue presented in Chapter 7.

## 1.2 Thesis Structure

The structure of this thesis is as follows:

**Chapter 1** provides the motivation for the work undertaken in this thesis, and highlights its main contributions. An overview of the peer-reviewed publications produced as a result of work undertaken in fulfilment of this thesis is also presented.

**Chapter 2** presents the required background knowledge for understanding the work presented in future chapters. An introduction to both photo-id and deep learning is provided, alongside a discussion of key computer vision concepts and their recent use in ecology. Existing photo-id aids are also examined.

**Chapter 3** outlines the creation of two key datasets developed to allow for work presented in later chapters of this thesis to be completed. The first is a coarse-grain dataset for cetacean segmentation, created using data collected from a previous study in 2015 around the waters of Zanzibar, Tanzania. The second is a dataset containing photo-id survey data collected as part of this thesis during 2019 around the waters of Northumberland, UK. This dataset contains labels allowing for both coarse-grained instance segmentation, as well as fine-grained species and individual level classification.

**Chapter 4** discusses the approach adopted to create a computer vision model capable of coarse-grained cetacean detection in noisy, above water photo-id fieldwork imagery. Initial model testing is discussed, including the use of transfer learning, data augmentation, and hyperparameter tuning via a grid search. Evaluation of the developed model is performed using both datasets created in Chapter 3.

**Chapter 5** provides details of the methodology developed to post-process the outputs of the model created in Chapter 4. Techniques outlined here allow for a reduction in the computational expense of downstream operations and ensure that no potentially important information which will assist in an identification is lost.

**Chapter 6** outlines the approach utilised for the task of automated fine-grained, few-shot most likely catalogue matching. The implementation of the chosen approach is examined in detail, including model training, evaluation metrics, and hyperparameter tuning via Bayesian optimisation. The use of uncatalogued individual thresholding is examined using generated class prototypes, Euclidean distance measurement, and the K-Nearest Neighbours algorithm over the model's latent space.

**Chapter 7** evaluates the chosen approach to most likely catalogue matching outlined in Chapter 6. Various data perturbations and their effect on model performance are examined, as well as a robustness evaluation using a second real-life photo-id catalogue

to explore the generalisability of the model, the previously determined uncatalogued individual thresholds, and the effect of species of interest and spatio-temporal changes.

**Chapter 8** summarises the conclusions of the work presented in this thesis and explores avenues for future work in the area.

## 1.3 Related Publications

The work outlined in this thesis has led to the publication of the following peer-reviewed papers:

- [39] Trotter, C., Atkinson, G., Sharpe, M., McGough, A.S., Wright, N. and Berggren, P., 2019. The Northumberland Dolphin Dataset: A Multimedia Individual Cetacean Dataset for Fine-Grained Categorisation. In *The 6<sup>th</sup> Workshop on Fine-Grained Visual Categorization, CVPR 2019*. Available: [doi.org/10.48550/arXiv.1908.02669](https://doi.org/10.48550/arXiv.1908.02669).
- [40] Trotter, C., Atkinson, G., Sharpe, M., Richardson, K., McGough, A.S., Wright, N., Burville, B. and Berggren, P., 2020. NDD20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation. In *The 7<sup>th</sup> Workshop on Fine-Grained Visual Categorization, CVPR 2020*. Available: [doi.org/10.48550/arXiv.2005.13359](https://doi.org/10.48550/arXiv.2005.13359).
- [41] Trotter, C., Wright, N., McGough, A.S., Sharpe, M., Cheney, B., Arso Civil, M., Tyson Moore, R., Allen, J., and Berggren, P., 2022. Towards Automatic Cetacean Photo-Identification: A Framework for Fine-Grain, Few-Shot Learning in Marine Ecology. In *2022 IEEE International Conference on Big Data (Big Data)* (pp. 1942–1949). IEEE. Available: [doi.org/10.1109/BigData55660.2022.10020942](https://doi.org/10.1109/BigData55660.2022.10020942).

The following work into the application of computer vision to ecology was produced during, but does not form a part of, this thesis:

- [42] Curry, R., Trotter, C. and McGough, A.S., 2021, December. Application of deep learning to camera trap data for ecologists in planning/engineering – Can captivity imagery train a model which generalises to the wild?. In *2021 IEEE International Conference on Big Data (Big Data)* (pp. 4011–4020). IEEE. Available: [doi.org/10.1109/BigData52589.2021.9671661](https://doi.org/10.1109/BigData52589.2021.9671661).

# Chapter 2

## Background

In recent years, deep learning models, large neural networks capable of exploiting abstract patterns in data to solve a task, have become a widely used technique to tackle problems faced in an ever increasing range of areas [43–46]. As an extremely fast paced and ever growing field it would not be possible to explore the entirety of deep learning, and as such this chapter will focus primarily on deep learning in a computer vision context, exploring and understanding image data.

One novel area where computer vision and deep learning can play an important role is in the world of marine ecology, helping to automate a currently labour intensive discipline. This project focusses on the automation of cetacean photo-identification, a process utilised by marine ecologists for tasks such as population estimates and health assessments [11, 13, 21, 33]. Before the research undertaken in this project is discussed, this chapter will seek to provide an introduction to both photo-identification and deep learning, before expanding into how this has been applied to computer vision. Literature focussing on computer vision in a cetacean ecology space is explored, as well as the current state of fine-grained recognition – utilising computer vision algorithms to differentiate between visually similar classes.

### 2.1 A Brief Introduction to Photo-Identification

One of the main goals of ecology research is to assess and monitor the status of both resident and migratory animal populations within a given geographic area. This is most commonly performed using mark-recapture analysis surveys in which researchers identify the number of unique individuals in an area at a given time, later returning to the same area and again identifying the number of individuals present [8–10, 12, 13, 47]. These values allow for an

estimate of the total population size to be obtained, with the accuracy of this value increasing proportionally to the number of recaptures.

Photo-identification, often abbreviated to photo-id, is one of the main non-invasive mark-recapture methods usually undertaken over large geographic areas at sea through the use of a small boat although monitoring from coastlines or aircraft may also be utilised [22–26]. More recently, the use of citizen science has started to play a role within photo-id surveys to monitor resident populations containing small numbers of individuals which traverse large geographic areas, making traditional monitoring infeasible [48, 49].

Initially utilised for the tracking of individual distinctive animals within a species [50, 51], the methodology was quickly adapted to large-scale monitoring of whole pods [52, 53]. Photo-id has been utilised for the monitoring of multiple cetacean species, with proven use cases in a range of studies such as those focussing on Indian Ocean humpback dolphins (*Sousa plumbea*) [10], Risso's dolphins (*Grampus griseus*) [27], Northern bottlenose whales (*Hyperoodon ampullatus*) [28], and killer whales (*Orcinus orca*) [9]. Outside of cetaceans, photo-id has further found use studying other marine life such as whale sharks (*Rhincodon typus*) [21], sea turtles (both *Chelonia mydas* and *Eretmochelys imbricata*) [19], giant sunfish (*Mola alexandrini*) [54], and Florida manatees (*Trichechus manatus latirostris*) [20]. Land based photo-identification studies are also possible, with Goswami *et al.* [29] utilising photographic data to estimate demographic parameters of Asian elephants (*Elephas maximus*).

Utilising photo-id for mark-recapture relies on the species having some form of individually identifiable markings, similar to human fingerprints. Typically for marine animals, this identifying information is located on a part of the body which is likely to breach the water at some point during an encounter – examples of underwater photo-id do exist however the practice is not yet commonplace [17, 55]. Depending on the species of animal, different parts of the body are the primary identifying location; for dolphins this is usually the dorsal fin whilst for whales this is primarily the fluke, or callosities if present [8, 10, 31, 56, 57]. See Figure 2.1 for examples.

During photo-id surveys, researchers will often focus on long lasting markers such as body-part shape, nicks, notches, and pigmentation which have been shown to be stable throughout the life of the animal [32–34]. In some cases, secondary markers, those which may heal and are thus not stable such as scarring, may also be utilised for identification. Markers may be anthropogenically caused, for example from collision with a vessel, or natural, for example from encounters with prey. Examples of prominent markings used in dolphin photo-identification surveys can be seen in Figure 2.2.

Scarring is of particular use when identifying Risso's dolphins, a species well known for the persistent nature of their scars caused by tooth rakes from other Risso's [60]. The use of



Figure 2.1 Examples of the main body parts utilised in cetacean photo-id. Left: callosities present on the head of a northern right whale (*Eubalaena glacialis*) [58]. Right: fluke of a humpback whale (*Megaptera novaeangliae*) [59]. Bottom: dorsal fin of a common bottlenose dolphin (*Tursiops truncatus*) [40].



Figure 2.2 Examples of prominent markings used during dolphin photo-identification surveys. Top Left: nicks. Top Right: scarring. Bottom Left: scratches. Bottom Right: pigmentation.

pigmentation for photo-id is also present in the literature for species such as striped dolphins (*Stenella coeruleoalba*) where it can be considered a primary marker [61].

Regardless of the species being analysed or the body-parts used during photo-id it is imperative that the process is standardised, allowing for work to be compared over spatial and temporal scales. This process began in 1988 through workshops held by the International Whaling Commission [22], with further recommendations published in 2015 by Urian *et al.* [18].

This standardisation process requires some assumptions to be universally made. For one, all of the markers must be considered stable, that is, they must not fade over the years. Even if a photo-id study only occurs over a few years, the markers utilised must be stable enough so that if another survey is conducted in the same area in later years, individuals from the first study must still be identifiable – providing useful information to health assessments, population estimates, and residency surveys. This stability reduces false positives, where one individual is recorded as multiple over time. Second, the markers must be considered individually unique. Those chosen to identify an individual must not overlap with other individuals in the survey area. This reduces the chance of false negatives, where multiple individuals are recorded as one. Chosen markers must also allow for a consistent re-sighting probability over time. This is critical for abundance estimates, ensuring that an individual's chosen markers provide it with the same chance of being spotted in one year as another. As such, it is extremely important that photo-id methodologies are standardised, both at an international level and between researchers in the same organisations. For a full review of marine photo-id methodologies, efforts to date, and the assumptions made during data collection, see Hammond *et al.* [62].

Because of the assumptions which must be adhered to, as well as the manual nature of the photo-id process, there are many downsides to the process. Being able to identify individuals relies on high quality photographs. Thanks to the advent of digital photography and the relative inexpensiveness of cameras capable of capturing large megapixel images, this is less of an issue than before, although it must still be considered. Surveys can also only be undertaken in good weather conditions in terms of sea state and lighting, both of which can affect the chance of an accurate match. These conditions are harder to meet in some areas of the world, reducing the suitability of photo-id for some geographic areas. Conditions, as well as the nature of the animal itself, may make photographing both sides of the individual impossible. Markings are rarely duplicated on both sides of an individual, and thus not having both sides may make matching difficult. For example, an individual may have an extremely distinctive marking on the left side of their dorsal fin, however if only the right side of the individual has been captured, when the left side is also eventually photographed

it may be labelled as a new individual as no previous examples of the individual's left side exist in the catalogue. If the individual has a very distinct fin shape then this issue can be overcome, although this may not always be possible. As such, individuals may not be added to a catalogue without both sides of the fin available for later comparison.

Furthermore, photo-id as a whole is extremely labour intensive. Unlike land-based camera trap systems, marine-based photo-id surveys require a large human effort. Staff are needed not just for photographic purposes, but also for piloting of vessels. As the surveyed animals are free roaming, large spatial areas must be covered, and there is no guarantee of encountering them during a given day. Back on land, the captured photographs must then be manually analysed and the individuals in them identified. This can often take longer than the entire data collection period. Due to the labour intensiveness of the photo-id process, it can also be extremely costly to undertake. Staff need to be paid, vehicles need to be fuelled, and equipment must be maintained. Because of this, any solutions which may speed up the photo-id process, provided they are robust, would be welcomed both by researchers and their funding bodies.

## 2.2 Machine Learning: Supervised vs Unsupervised Approaches

Before it is possible to understand how deep learning and computer vision can be utilised to aid in the photo-id process, it is important to discuss the differences between supervised and unsupervised machine learning.

**Supervised learning** tasks are those where the model can be trained using an input and expected output value pair, known as a ground truth. This technique lends itself well to tasks such as classification, where an input can be mapped to a set of defined output classes, or in regression where an input can be mapped to a continuous output space.

Training is performed by splitting the available data into training and test sets, with the former being used to train the model and the latter being used to test the model's performance on previously unseen data. Both the training and test set contain ground truth data, but only the training set's influences the generalisation of the model. For example, in the case of a dog verses cat classifier a dataset may contain one thousand images, some labelled as dog and some as cat (the ground truths). This data will then be split randomly into a training and a test set; for example 80% of the data used for training with 20% used for testing. The classifier will then iterate through the training set, using the ground truth values to train the network's parameters in a way to best generalise the model. After training has been completed the model will then be evaluated against the test set. Each data point will be

processed by the model and a prediction outputted, which is then compared to the unseen ground truth to provide an evaluation of the model’s performance.

**Unsupervised learning** tasks are, in contrast, those where prior ground truths for the data are not available. This approach lends itself well to clustering and aiding in the understanding of the underlying data structure. These unsupervised algorithms, such as K-Means clustering [63], are not provided human guidance on how to group the data given, but are rather left to discover interesting structure patterns on their own.

Taking the dog and cat labelled data again as an example, this data could be clustered in an unsupervised manner. Asking the clustering algorithm to provide two clusters for the data (e.g.  $K = 2$ ), a model could be trained to split the data with all dogs in one cluster and all cats in the other, without having to be told which images are dogs and which are cats. However, due to the unsupervised nature of the learning process, the model is equally as likely to cluster the data based on whether the animal is, for example, sitting or standing.

## 2.3 A Brief Introduction to Deep Learning

Deep learning, a subfield of machine learning, aims to create artificial networks to complete tasks through a learning process. These computational models are made up of neurons and are often multiple layers deep. Earlier layers represent basic abstractions building up from this as you go deeper into the network. Layers at the deepest points can, based on information passed to them from higher layers, begin to provide estimations of answers to a given problem. The ability to learn directly from the data provided is the key difference between deep learning and more classical machine learning techniques, which often require considerable domain expertise to design a feature extractor allowing for raw data values, such as pixels, to be transformed into a feature vector suitable for a model to learn.

Deep learning models in contrast are capable of learning to perform tasks such as classification on raw data values through multiple layers of simple non-linear transformations. For example in the case of computer vision, earlier layers of neurons are optimised by the network itself to learn lines and basic shapes, middle layers may be optimised to learn more complex ideas such as how these lines and shapes fit together, with the final layers providing an output of object label (e.g. dolphin). It should be stressed however that the features these layers are looking for are not specified by humans, but rather learned from the data by updating their parameters predominantly through optimisations such as stochastic gradient descent (see Section 2.3.1 for further details).

This ambition to create artificial networks similar to how the brain operates stems mainly from work undertaken in 1943 by McCulloch and Pitts [64] in an attempt to understand how

neurons in the brain allow for the understanding of complex patterns. This model formed the basis of future work into machine learning, and thus deep learning. This work continued at small scale for many years. It has only been recently, thanks to advances in availability of large scale datasets needed to train these networks and power of the computing resources available, that deep learning research has accelerated. The transition to training on clusters of high-powered Graphical Processing Units (GPUs) has allowed for a significant speed-up in both model training and inference time compared to traditional Central Processing Units (CPUs), allowing for a higher amount of prototyping in a smaller time frame [65]. Further to this, the advent of cloud computing has allowed for much more cost-effective model development.

Development of deep learning models has been helped greatly through the creation of standard programming frameworks. Google's Tensorflow [66] and Meta's PyTorch [67] allow for researchers to develop models much faster than previously due to their reduction in the amount of boiler-plate code needed, with these frameworks often doing a lot of the heavy lifting in the background. Further advances have been made through the availability of large scale coarse-grained datasets such as MNIST [68], Caltech-256 [69], and ImageNet [70] allowing for common baselines to be adopted by the computer vision community and for the introduction of transfer learning, allowing for the reuse of models trained on one task to be utilised for another [71]. Furthermore, additional regularisation techniques have provided improvements to model accuracy. Notable examples of this in the literature which are now commonplace in deep learning models include dropout [72], batch normalisation [73], stochastic gradient descent with warm restarts [74], and mixup [75]. The use of data augmentation is also commonplace. This is where existing data is perturbed randomly to create new, artificial data. Examples of image data augmentation include simple perturbations such as random horizontal and vertical flipping, and cropping, as well as more complicated techniques like Gaussian blurring, perspective shifting, and colour jitter.

### 2.3.1 Optimising Deep Learning Networks

In order to generalise deep learning models, there exists a need to optimise the parameters within each individual neuron. These parameters can either be weights that control the strength of the connection between two neurons, or biases – some constant additional input. Most commonly, this is performed using gradient descent to minimise some loss function (a measure of distance between ground truth and model prediction) with respect to a dataset. If this function were to be visualised on a graph each parameter would be represented by an axes, resulting in a hyper-surface with millions of dimensions in the case of deep learning

models. The goal of network optimisation is to find the minimum point on the hyper-surface, as this would give the parameter values which produce the smallest loss.

This hyper-surface is non-convex however, which can result in multiple local minimums. In order to find the (hopefully global) minimum point on the hyper-surface, during training the model's weights must be updated iteratively in the opposite direction of the gradient of the loss function's hyper-surface. As such, we follow the direction of the slope of the hyper-surface downhill until we reach a minima, an area where the loss is lowest [76].

Before the advent of deep learning and big data, it was common for the whole training set to be used to compute the gradient at each iteration; however due to the size of modern day datasets this is no longer possible due to the computational cost this would impose on the system. As such, batches, a small random subset of the larger dataset, are often used to give an estimation of the overall loss gradient.

In order to achieve this, a process known as mini-batch stochastic gradient descent (SGD) is commonly used. At each iteration of SGD, a mini-batch which contains some subset of the dataset is utilised rather than the entirety. The loss for this batch is calculated and used to step down the gradient slope, rather than the sum of the loss' gradient over all training examples. As we only take some data subset per iteration, the path taken down the slope to the minima is far noisier and more random than the path obtained from using all examples. Whilst this results in a longer convergence time to the minima compared to non-batch gradient descent, this is outweighed by the fact that the entire dataset does not need to be loaded into memory at once, allowing for computation to be performed using machines which are, relatively speaking, more resource constrained. The use of SGD often leads to a good set of model weights quickly compared to other, more elaborate techniques [77].

In recent years there have been efforts to modify SGD in an attempt to improve model optimisation. The most commonly seen optimisers within production code include SGD with warm restarts (SGDR) [74], Momentum [78], RMSProp [79], Adam [80], and AMSGrad [81]. Work in this thesis utilises SGDR and Adam primarily – for a more in-depth discussion of these, see Section 4.4.2. All of these optimisers attempt to stop the problem of getting stuck in local minima rather than the global minimum of the overall loss function (although it should be noted that it is very unlikely the global minimum will ever be reached during the training process). However, recent studies show that the problem of local minima is not as big as first thought and, regardless of initial conditions, vanilla SGD rarely gets stuck in local minima [82, 83].

### 2.3.2 Backpropagation

As discussed in Section 2.3.1, the weights and biases in each neuron can be learnt and optimised using optimisers such as SGD. However, it is also important to understand how a weight's or bias' effect on the gradient is computed. This can be performed relatively quickly using backpropagation, or the backward propagation of errors algorithm. Before delving too deep into deep learning, it is of imperative importance to understand backpropagation; it is after all often cited as one of the cornerstones of deep learning [84].

Backpropagation was originally described in Linnainmaa's Masters thesis from 1970 [85], although its effect was not fully realised until 1986, when Rumelhart *et al.* discussed the advantages to using backpropagation over other learning approaches [86]. In recent years multiple works have provided updates and improvements to the original backpropagation algorithm, however none have seen wide-scale adoption [87–91].

During each training step, the output for each example input is predicted using the weights and biases of each node in the network. Any error in the output prediction can be represented as a weighted sum of the error in the node's weights and biases. As such, this error can be corrected by changing these weights and biases on a per node basis. However, as nodes between layers can be connected together and influence each other (network architecture depending), it can be computationally expensive to determine the change required for each node in isolation.

The backpropagation algorithm provides a computationally efficient way of calculating the change required with respect to the gradient of the loss function. Operating on a per node basis, the algorithm works backwards through the network determining how each node's weights and biases should be updated. For each node, the calculated gradients of the loss function are passed backwards through the network to the preceding directly connected nodes to be used in their calculations, up until the input layer. As such, the standard backpropagation algorithm is, in essence, the chain rule (a formula for computing the derivative of compound functions), and provides a far more computationally efficient way of calculating the overall gradient loss compared to calculating each layer's gradient loss in isolation. Further efficiencies have been made thanks to deep learning frameworks implementing backpropagation in a way that takes advantage of GPUs, leading to extremely efficient computations when performing deep learning tasks.

## 2.4 Deep Learning for Computer Vision

The field of computer vision, allowing computers to gain and interpret knowledge from image and video data, is one area where deep learning has excelled [92]. Generalisable concepts such as Convolutional Neural Networks (CNNs) have quickly become commonplace for solving computer vision tasks, in most cases replacing the need for hand-crafted pipelines specialised to the task at hand, thanks to their ability to learn complex patterns in data where there is a strong spatial and temporal dependency between the values. This ability is essential for processing image data which is, at its most basic level, a tensor of pixel values. These tensors are three dimensional, representing an image's height, width, and depth, where depth is dependant on the colour model used to represent the image. The most common of these models is RGB which has channels representing the red, green, and blue colours present in an image; a matrix representing an RGB image will have a depth of three. Other colour models have varying depth values, a greyscale image would have a depth of one for example, whilst a CMYK image which has cyan, magenta, yellow, and black colour channels would have a depth of four. As can be imagined, these matrices can very quickly reach unworkable sizes. An RGB image at 1080p resolution for example would require a tensor of size 1920x1080x3, or 6,220,800 values. Thankfully, CNNs utilise a set of standard operations which are capable of reducing image sizes down to a more workable form whilst still retaining key features which allow for inference to occur.

### 2.4.1 Convolutional Neural Networks

Modern CNNs are composed mostly of three main layer types; convolutional layers, pooling layers, and fully connected layers. Each of these layers will perform some operation on the input matrix passed to it, and provide a transformed output to the subsequent layer(s). These layers can be stacked in various orientations to build different CNN architectures.

#### Convolutional Layers

The convolutional layer is the workhorse of the CNN, performing the vast majority of the operations required. Convolutional layers utilise what is known as a kernel in order to efficiently extract features from an input image matrix. A kernel is a matrix, most commonly 3x3 or 5x5 in size, which represents some weighting refined through training. This kernel is slid over the input data computing the dot product between the weights and the section of the input matrix it is currently over, before summing these into a single value and passing them through an activation function. This gives an output matrix of features represented

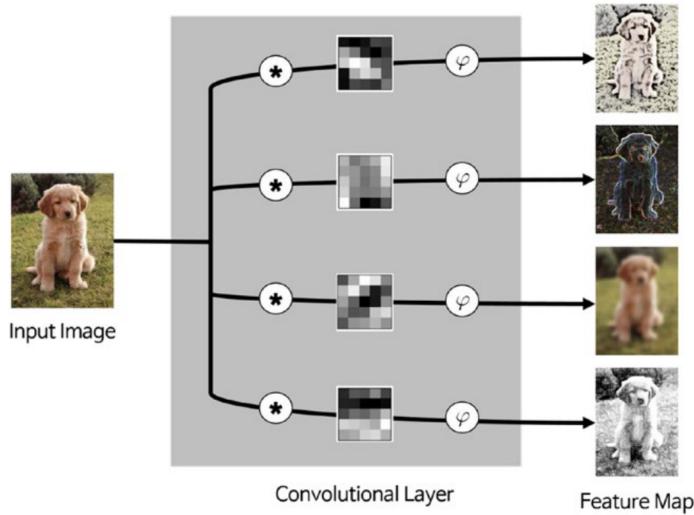


Figure 2.3 A visual representation of convolution. An image, left, is fed into a convolutional layer. The input is passed through a convolution operation,  $*$ . The greyscale blocks in the centre of the convolutional layer represent the kernels passed over the image.  $\varphi$  represents the activation function the kernel output is fed to. This process results in multiple outputted feature maps. Image from [93].

by the weighted sum of the input, known as a feature map. These feature maps generally represent basic shapes at shallow levels, which are then built on as the model gets deeper [93]. The kernel process can be performed multiple times over the same input using different weightings, giving multiple output feature maps. A visual representation of a convolutional layer can be seen in Figure 2.3.

In the case where there are multiple input dimensions (such as an RGB image), kernels are required to operate over all dimensions. As such, the resulting feature maps are summed element-wise, along with some bias term, to produce a single output map. The size of the kernel determines the number of input features which are combined to give the new output feature map, although the size of the resulting map is determined also by two other properties; stride and padding. Stride refers to the distance in pixels the kernel will move when performing the next input mapping. For example, a stride of 1 would result in the kernel sliding along one pixel value each time, resulting in an output feature map of equal size to the input, whereas a stride of 2 would skip every other pixel, reducing the output feature map by half. Figure 2.4 shows a visualisation of the kernel process with a stride of 2.

A problem can arise during convolution when the kernel reaches the edges of the input matrix. As there is nothing past the edge values, these must be trimmed as they can never be in the centre of the kernel. This is known as ‘valid’ padding, and can cause a reduction in the size of the matrix which may be detrimental when an output feature map of equal or greater

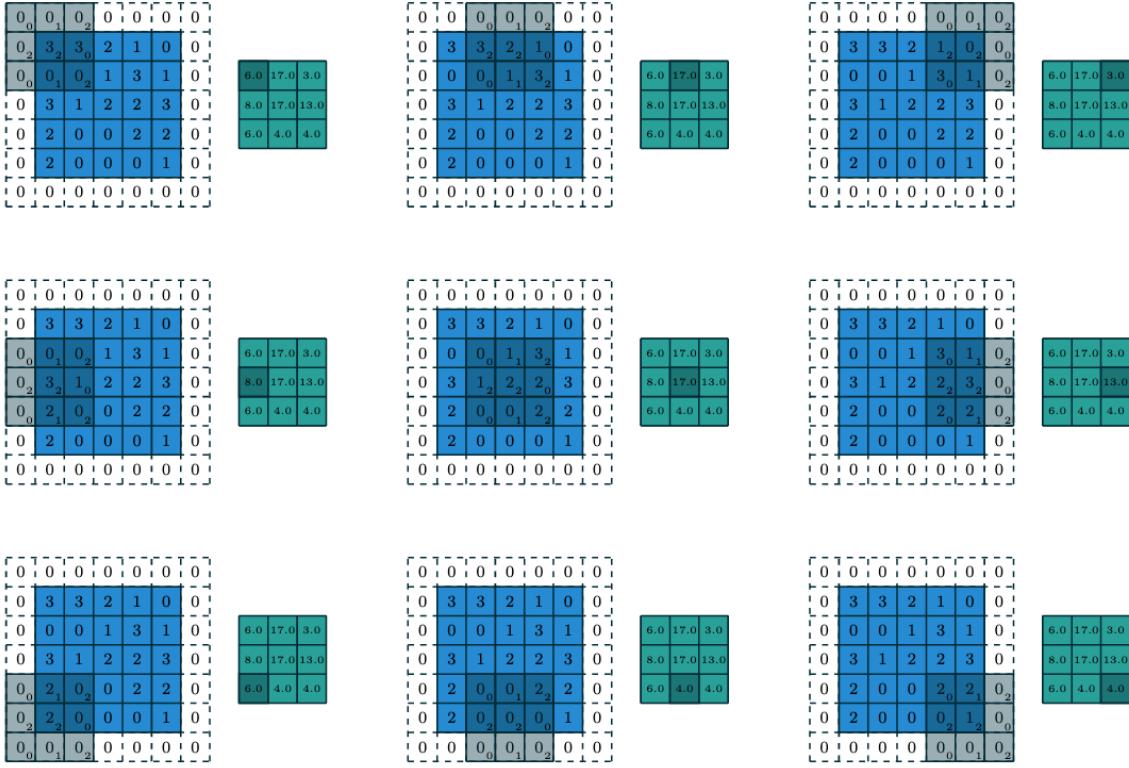


Figure 2.4 A kernel, represented by the grey squares, operates over a padded input matrix, blue, with a stride of 2 to produce an output feature map, green. Note that the kernel’s weights are denoted by the number in the lower right of each box and the input pixel value is denoted by the number in the middle of each box. Image from [94].

size than that of the input is required. To avoid this, ‘same’ or zero padding can be utilised whereby zeros are added to the edges of the matrix, allowing the input and output feature maps to be of the same size. ‘Full’ padding can also be utilised, whereby multiple rows and columns of zeros are added such that all non-padding values are visited equal amounts of times by the kernel. This has the effect of increasing the size of the output feature map.

## Pooling Layers

Pooling layers help reduce the computational complexity of the convolutions performed by the CNN. This is achieved by reducing the spatial dimensions of the input ready for the next convolutional layer through the use of some function applied over batches of the input pixel values, similar to how a kernel operates in a convolution layer by sliding over the image.

Pooling only affects the width and height of the input, not the depth, as all depth channels are required to keep the colour mapping of the image intact.

Pooling as such inevitably leads to a reduction in the amount of information available to subsequent layers; this is advantageous however as it leads to less computational complexity, aiding in the minimisation of overfitting in the model.

A number of different pooling layer architectures exist in the literature, such as max pooling which only keeps the maximum pixel value in the batch, average pooling [95] which outputs the mean of all pixel values in the batch, and stochastic pooling [96] which selects an output pixel value from each batch based on a probability distribution. For a review of current pooling methods, see Gholamalinezhad *et al.* [97].

## Fully Connected Layers

Fully connected layers take feature maps produced by the preceding convolutional and pooling layers and reduce these down to a single  $N$ -dimensional vector. In the case of the last layer of a network geared towards a classification problem,  $N$  represents the total number of possible classes the network was trained to identify, with the values in the vector representing the input's class probabilities.

An activation function is responsible for deciding which neuron in the layer passes its value to the layer below by computing the weighted sum of the inputs and passing the result through some non-linear function. This has the effect of scaling the layer's output to within some defined limits. Multiple common activation functions exist in the literature, such as the tanh function which scales values to between -1 and 1. The use of the softmax activation function is commonly used in the last layer of a classification network in order to determine the class probabilities mentioned previously. Softmax takes the exponents of each input and normalises them by the sum of all inputs, giving values between 0 and 1. Outputed  $N$ -dimensional vectors can then be considered a feature map in their own right for further processing [98] or as a category for classification as the last layer of the network [99].

Figure 2.5 shows the final layers of a network whose aim is to classify an image into one of four classes. At the end of this network is a fully connected layer which takes a feature map from the preceding hidden layer, and produces as output four values between 0 and 1. Each of these values is required to be outputted by its own neuron, resulting in a fully connected layer with four neurons where each neuron represents a possible class for the input image. The image's classification is provided by whichever neuron outputs the highest value.

## Layer Architectures

Using the three layer types described previously it is possible to create an infinite number of CNN architectures. There is no guarantee that every possible architecture will perform well however (indeed, one possible combination would be a single fully connected layer, which would not perform well at all). Whilst it may be advantageous for certain areas of research to create their own custom CNN architecture, either through trial and error or the more recent approach of Neural Architecture Search [100], due to the computational expense of identifying good networks many choose not to perform this work. For the vast majority of cases, there exists in the literature well-defined generalised CNN architectures which have demonstrated that they produce good results, and it is often these architectures which are utilised for computer vision tasks.

LeNet [68] was the first well defined CNN architecture. LeNet was only seven layers deep, but performed well enough to be applied by some banks for automatic recognition of numbers on cheques. It was not until around 2012 that more attention was paid to these defined architectures however, thanks to AlexNet [98]. Utilising a similar but deeper architecture to LeNet, with more filters and a larger number of stacked convolutional layers, AlexNet also included now commonplace deep learning building blocks such as dropout

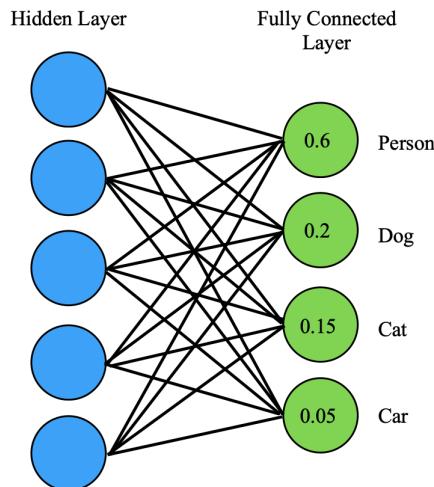


Figure 2.5 A visual representation of the final stages of a classification network. The blue circles represent the final hidden layer in the network, with the green circles representing the fully connected layer. Each neuron in the hidden layer is connected to all neurons in the fully connected layer. The value in the hidden layer's neurons represents the activations after softmaxing, with the label denoting the class represented by the neuron. The network has predicted the input to be of class Person. Note that additional CNN layers would be present before the hidden layer.

whereby nodes in the model are intentionally not updated during a training step with some probability to aid model generalisability [72], max pooling [95], and Rectified Linear Unit (ReLU) activation functions; the most popular non-linear activation function currently in deep learning, especially in computer vision [101].

In 2014, Google introduced GoogleNet, also known as an Inception architecture, to the ILSVRC14 competition [102]. This architecture achieved a top-5 error rate of 6.67%, very close to what untrained humans could achieve on the competition dataset, ImageNet [98]. This was achieved through a 22 layer deep CNN utilising several small convolutions, reducing the number of parameters from 60 million in AlexNet to 4 million in GoogleNet.

Finally ResNet [103] was introduced a year later at ILSVRC15. This architecture can be up to 152 layers deep, and achieved a human-beating top-5 error rate of 3.57%. Shallower versions of ResNet exist, such as ResNet50 and ResNet101, which are 50 and 101 layers deep respectively.

## 2.4.2 Object Detection

Thanks to advancements in deep learning technology and the creation of standardised architectures, CNNs are now utilised en masse to perform tasks such as object detection, attempting to identify distinct regions containing task-specific classified objects in images and video. Whilst this is often performed in one of two ways, it is important to note that all object detection is still in essence a series of tasks performed via network layers.

### Region Proposal Networks

The first, known as a Region Proposal Network (RPN), attempts to find image regions likely to contain objects of given classes. Training data is usually provided in the form of bounding boxes drawn around objects of interest and labelled with the corresponding class. One of the most common and widely used RPN architectures is derived from the R-CNN, or Regions with CNN features, architecture [99]. R-CNN utilises a selection search [104] to generate 2000 Regions of Interest (RoIs) representing the most likely areas of the input image to contain a class example. By limiting the number of RoIs generated this allows for fast computation compared to operating on every possible region in the image. The proposed RoIs are then fed through a CNN to be classified. See Figure 2.6 for a visual representation of the R-CNN pipeline. Example proposed RoIs can be seen in Figure 2.7.

Utilising selection search leads to a high recall rate thanks to the large amount of proposals, as there is a high probability that some of these proposals will contain RoIs with objects being searched for. However, this can be time consuming and computationally expensive

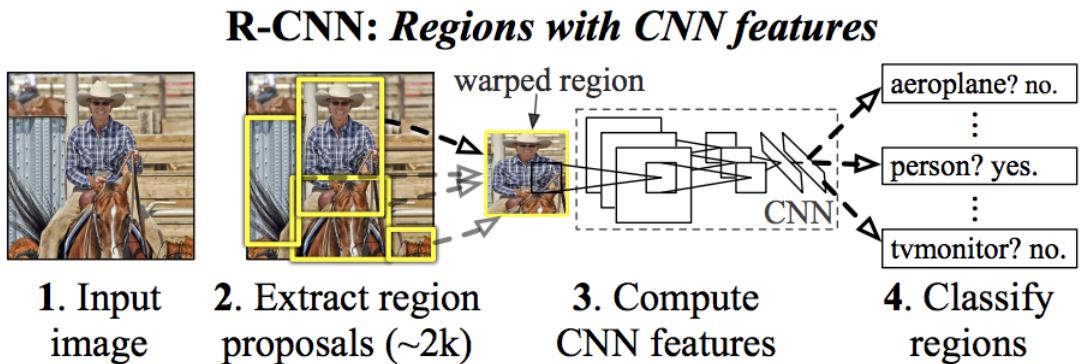


Figure 2.6 The R-CNN pipeline. (1) Input images are passed to (2) a region proposal network, extracting around 2000 RoIs. (3) Each ROI is then resized and passed through some CNN backbone architecture. (4) The output of the CNN corresponds to the ROI's classification and confidence score. Any ROIs with confidences above some defined threshold are outputted by the model. Image from [99].

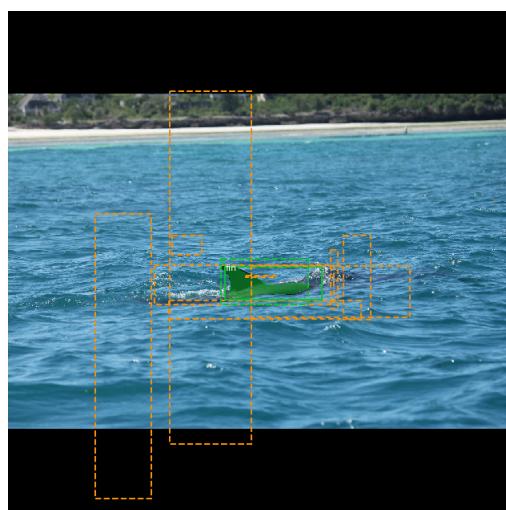


Figure 2.7 An example of ROIs generated on an image by an RPN, showing 10 random proposals.

(although less computationally expensive than just sliding a window over the full image) as the network needs to classify the 2000 region proposals generated. Detection can also be slow using an R-CNN and, with the selection search being fixed, no adaptive learning takes place here which may lead to bad region proposal generation.

Some of these time drawbacks were fixed in a later version of R-CNN, known as Fast R-CNN [105]. Rather than feeding the region proposals generated to the CNN, this algorithm instead feeds the input image to the CNN and generates a feature map. RoIs can then be taken from the feature map using selection search and warped into a shape suitable for the pooling layer, before being reshaped again into a fixed size for the fully connected layer. This is advantageous as it allows us to reuse some computations and allows for backpropagation to occur throughout the network, greatly improving run times. This also means however that the runtime is determined by how fast RoIs can be generated.

To fix this issue, Faster R-CNN was developed [106]. Now instead of utilising selection search to generate the RoIs, a separate network can be utilised to predict RoIs which are then used to classify images within the regions. As such, training now occurs with four losses:

1. An object/not object classification from the RPN; 2. The RoI shift; 3. The object classification; 4. Final bounding box co-ordinates.

Building upon Faster R-CNN, Cascade R-CNN [107] aims to reduce the overfitting during training and quality issues at inference through the use of a proposal sub-network to produce preliminary detection hypotheses which are then processed by an RoI sub-network.

## Detection Without Proposals

One issue with all RPNs is that they generally take a significant amount of time in order to classify objects in images, with the bottleneck being the region proposal generation. Because of this, there are algorithms which attempt to remove the region proposals altogether and instead look at the whole image. The input image is divided into an equal size grid. Within each square of the grid a set number of bounding boxes are generated, which the CNN provides classification confidences for. Any above a set threshold are used to locate the object within the image. These algorithms are essentially one large CNN rather than splitting into a CNN and an RPN and are thus much faster although are not as accurate, especially on smaller objects due to the spatial constraints of the algorithm. Examples of detection without proposal systems include YOLO [108] (including the more recent state-of-the-art versions [109–114]), SSD [115], DINO [116], RetinaNet [117], and EfficientDet [118].



Figure 2.8 Left: an example input image to a semantic segmentation model. Right: the resultant semantic segmentation mask for the given input image. All pixels which make up any cow object have been classified as one cow mask. Image from [119].

### 2.4.3 Semantic Segmentation

Along with object detection, semantic segmentation is one of the key research areas in computer vision. Rather than provide ROI bounding boxes as output, semantic segmenters instead output a class label for each pixel in the image. A group of connected pixels of the same class is known as a mask. An example semantic segmentation mask can be seen in Figure 2.8.

In general, semantic segmenters can be thought of as having two main components; an encoder, usually a pre-trained classifier built with a standard detection architecture such as ResNet [103], and a decoder whose job is to project the coarse-grained features learnt by the encoder to a fine-grained pixel space. There are two main ways to approach this decoding step.

The first is to use an RPN to perform region based semantic segmentation, extracting the regions from an image and then describing them. Each pixel of the image is then given a classification based on which highest scoring region it is contained in. Note that any pixels not in a region are given the class label of background. This is achieved through the use of a lightweight binary classifier operating over multiple proposal boxes, known as anchors, covering the image at different scales. Each anchor is given an object score denoted by Intersection Over Union (IOU), a measure of how much overlap there is between a model's predicted bounding box and the ground truth. This is taken at a set confidence threshold, usually 50%, as the model will predict potentially hundreds of boxes for an image, all with different confidence levels. The vast majority of these predictions will be wrong, but will also (hopefully) have very low confidence scores and so they can be safely ignored and thus

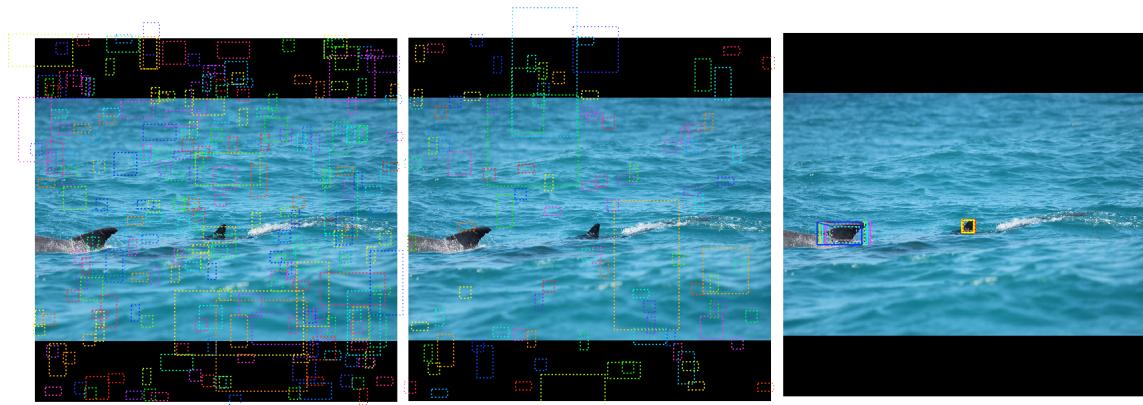


Figure 2.9 Generated anchors. Left: negative anchors. Middle: neutral anchors. Right: positive anchors.

not counted in evaluation metrics. Taking a predicted bounding box  $B_p$  and a ground truth box  $B_g$ , the IOU between the two can be defined as:

$$IOU = \frac{\text{Area of overlap}(B_p, B_g)}{\text{Area of union}(B_p, B_g)}. \quad (2.1)$$

Anchors with an  $IOU \geq 0.7$  with any ground truth are denoted as positive anchors and are passed on for classification. Those with an  $IOU < 0.3$  are considered negative anchors, and are utilised during the training process as negative class examples. Those where  $0.3 \leq IOU < 0.7$  are denoted as neutral anchors and are not used for training. An example of generated negative, neutral, and positive anchors for an image can be seen in Figure 2.9.

In some cases, positive anchors may not fully cover the ground truth object. Because of this, the RPN regresses a refinement applied to the anchors, shifting and resizing them to correct their encasement of the ground truth object. An example of this can be seen in Figure 2.10.

Utilising RPNs does have disadvantages however. Generating the segmentations from the regions takes a significant amount of time, and the features generated by RPNs generally do not contain enough feature information to generate well defined masks. Recent research has attempted to fix these issues, such as SDS [120] or Mask R-CNN [36]. Work undertaken in this thesis makes use of Mask R-CNN, see Section 4.2 for discussion of the reasoning behind this.

Fully Convolutional Networks (FCN) can also be utilised for semantic segmentation [121]. An FCN learns pixel to pixel mappings without the need for region proposals and are built using only convolutional and pooling layers, allowing for an input image of arbitrary

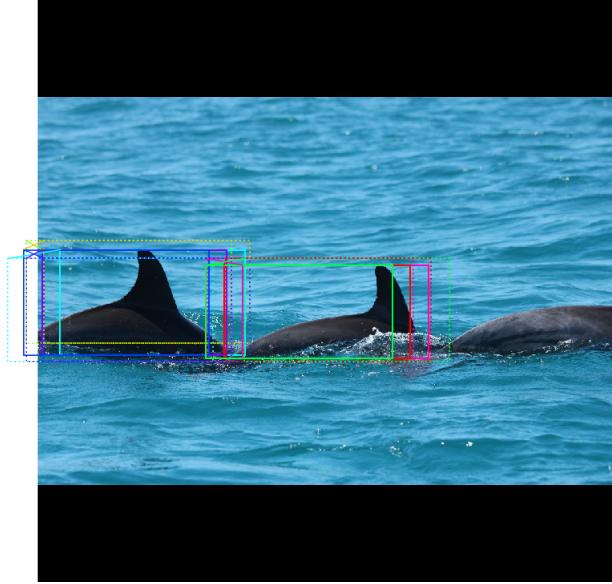


Figure 2.10 An example of refined anchors. Positive anchors before refinement are dotted, after refinement are solid.

size (compared to classical CNNs which are generally constrained by a preset image size). This does lead to the disadvantage of down sampling the resolution of the outputted feature maps, leading to sometimes ill-defined segmented boundaries. This issue has been tackled through the development of more advanced network architectures such as SegNet [122], DeepLab [123, 124], ConvNeXt [125], and PSPNet [126]. More recent approaches to the task of semantic segmentation also make use of Transformer [127] architectures, such as Segment Anything [128] or Swin [129].

Semantic segmentation can be aided through forms of supervised learning. Providing training images which have been given pixel by pixel segmentation masks can greatly improve segmentation class accuracy. Creating these masks can be extremely time consuming for researchers, and is often farmed out to external companies such as Amazon’s Mechanical Turk [130]. However, doing this can lead to a wide variance in the quality of ground truth masks generated due to the financial incentive for those creating the masks to work as quickly as possible, necessitating the need to develop bespoke systems for quality control [131].

#### 2.4.4 Part Segmentation

In part segmentation, a coarse-grained classification is broken down into sub-components which are then analysed to provide a fine-grained identification [132]. This is still an active area of research, with some approaches focusing on a form of hierarchical part matching



Figure 2.11 Left: an example of some input image. Right: the corresponding dolphin ground truth instance segmentation masks. Each mask is highlighted a different colour, with background denoted black.

[133], some on alignment of objects to define a super-class shape [134], some utilising deformable part descriptors [135], and others using part localisation [136].

At first glance it may seem as though part segmentation would be useful for this thesis' work into automatic photo-id. However, when analysing the target data it becomes apparent there would be little benefit to this approach over other segmentation techniques. As this work focusses on above water photo-id the vast majority of images observed by the developed system only contain one part of the animal – the dorsal fin as it breaches the waterline. As such there would be little advantage to the use of part segmentation here, as only one part is visible. If this work shifted focus to include underwater photo-id, part segmentation would be extremely useful. In this case there would be multiple parts of the animal visible such as the dorsal, tail stock, head, ventral, or sides. Each of these parts would have their own prominent markings useful for identification. However even if all parts of the animal are visible, it may be the case due to water conditions that not all prominent markings are. Algal bloom and light refraction can obscure an individual's identifying information, and so it would be very useful to be able to break down an individual using part segmentation, allowing for identification to be performed on each part rather than the whole animal, potentially increasing the chance of identification.

#### 2.4.5 Instance Segmentation

Building on the concept of semantic segmentation, instance segmentation can be performed when further detail about an image is required by a developed system. Whilst many of the underlying processes are similar between the two segmentation types, instance segmentation allows for the model to distinguish between multiple objects which are of the same class; an example of this can be seen in Figure 2.11.

As such, instance segmentation provides a far more detailed explanation of the input image. This information can be invaluable if the developed system is required not only to understand what pixel classes are present in the input image, but also how many of these class instances there are. These systems are often expensive to develop, however, due to the increased workload of data labelling required. Compared to data labelled for object detection, which only requires a bounding box ground truth, instance segmented data is required to be labelled on a pixel basis in a similar manner as semantically segmented data. However instance segmented data requires each pixel be assigned a specific object if there are multiple objects of the same class in the image which is not the case for semantically segmented data.

In a sense, instance segmentation can be seen as combining both object detection and semantic segmentation into one task. Traditionally, however, in order to achieve the goal of instance segmentation, proposed systems have kept the two tasks divided. These traditionalist methods take one of two approaches.

The first, known as top-down, begins by detecting objects of interest via an RPN to create bounding boxes. These detections are then fed to the mask predictor to determine which pixels inside of the boxes belong to either the target or background class. Examples of top-down approaches to instance segmentation include Mask R-CNN [36], built upon Faster R-CNN [106], and HCT [137], built upon Cascade R-CNN [107].

In contrast, bottom-up systems first segment then detect, such as SpatialEmbedding [138] which attempts to tackle instance segmentation through the use of a Gaussian function to produce a probability for a pixel being part of the background or foreground, and then performing object detection on the foreground pixels. The major similarity between both top-down and bottom-up approaches is that they are both sequential in nature, requiring one stage to happen before the other. As such, these systems are very hard to speed up. However two stage systems often perform the best in terms of accuracy, and thus are still extremely common backbones of systems requiring the use of instance segmentation [139].

In recent years research into the development of fast instance segmentation has shifted to utilising a one stage approach. These one stage systems are often able to achieve faster performance than their two stage counterparts, although often struggle to reach the same levels of segmentation accuracy [139]. ExtremeNet [140] works to extract four “extreme points” and one “center point” of potential objects in the input image through the use of a keypoint estimation network, creating a coarse mask. ESE-Seg [141] utilises the concept of Chebyshev polynomials to fit a radius around each object inside of the detected bounding box. Similarly, PolarMask [142] also represents masks through the use of a contour around the object, modelling this through the use of polar coordinates. FourierNet [143] builds on this radius concept further through the use of a Fourier transform to smooth the contour. This

contouring of the object is extremely fast; however, the generated masks are very imprecise. Further, any objects which contain spaces or holes, such as doughnuts, would not be able to be accurately represented.

YOLOACT [144] builds on the well known YOLO object detection architecture, specifically YOLOv3 [145], adding a branch for mask prediction, but performing this through the use of two parallel tasks. The first utilises an FCN to generate prototype masks, whilst the second predicts instance coefficients. These can be combined into one mask through matrix multiplication operations with the detected bounding box. BlendMask [146] works in a similar way to YOLOACT, however, it predicts an attention map rather than instance coefficients and utilises FCOS [147] as a backbone, a completely anchor and proposal free object detection architecture resulting in reduced complexity when compared to YOLO [108] and SSD [115].

Whilst the majority of one stage approaches to instance segmentation rely on bounding boxes, this is not always the case. SOLO [148] introduces the concept of instance categories, assigning categories to each pixel according to the size and location of the instance. SOLOv2 [149] builds on SOLO through the implementation of a novel non-maximum suppression algorithm. SOLOv2 often depicts higher quality masks than more often used two-stage systems such as Mask R-CNN and is able to perform real-time inference. Further, the use of Transformer [127] architectures such as Mask DINO [150] and EVA [151] has allowed for state-of-the-art instance segmentation results to be obtained on a variety of benchmark datasets. It should be noted however that these architectures are recent additions to the instance segmentation arsenal, with SOLO and SOLOv2 released in 2020 whilst Mask DINO and EVA were published in 2022.

## Mask R-CNN

As discussed in previous sections, there are multiple standardised architectures utilised for segmentation tasks. As such, when developing a system that utilises segmentation, developers of these systems will, more often than not, use one of the many architectures from the literature rather than developing their own custom architecture. Utilising one of the standard architectures has many advantages; for one, researchers do not need to spend time creating a model architecture for their task, allowing for development in other, novel areas. Further to this, utilising a standard architecture allows for research to be more easily understood and reproduced. As this thesis focusses on the automation of photo-identification systems rather than on the development of new novel architectures, it makes sense to make use of an architecture that is well known, has a track record of performing well when trained on non-benchmark or custom datasets, and is easily reproducible. As such, parts of this

project’s automation pipeline make use of Mask R-CNN [36]. Because of this, it is important to understand Mask R-CNN in more detail compared to the other architectures discussed previously in this chapter.

As seen previously, it is often the case that new architectures either extend or borrow features from older ones. This is also the case with Mask R-CNN, developed in 2017 by He *et al.* on top of the existing 2016 Faster R-CNN architecture from Ren *et al.* [106] (itself an extension of Fast R-CNN developed in 2015 [105]).

Faster R-CNN is a two stage architecture. The first stage utilises a standard backbone network such as ResNet [103], VGG [152], or Inception [102], to convert an input image into a set of feature maps which are passed to an RPN for analysis (see Section 2.4.2 for a breakdown of RPNs). This RPN generates region proposals which are passed to the second stage of Faster R-CNN, along with the previously generated feature maps, and fed to an RoI pooling layer. Here, each proposed region and corresponding feature map is utilised to predict bounding boxes, classifications, and confidence scores. A visual representation of Faster R-CNN’s architecture can be seen in Figure 2.12.

Mask R-CNN extends Faster R-CNN, allowing for instance segmentation through some relatively simple changes and additions to stage two of the architecture. First, the RoI pooling layer is replaced with an RoI align layer. This replacement layer removes the “harsh quantisation” which is present in RoI pooling, and properly aligns the extracted features with the input image. Second, an additional branch is added to the end of stage two. This branch receives the output of the new RoI align layer and processes it using a mask head, consisting of additional convolutional layers which generate pixel predictions and instance mask outputs. See Figure 2.13 for a visual representation of the changes made by Mask R-CNN.

Thanks to these additions, Mask R-CNN is able to perform extremely accurate instance segmentation with a relatively small drop in inference speed, even when predicting on custom datasets. This speed is greatly valuable when a large number of images are required to be processed in a batch manner, such as overnight in between photo-id surveys. Indeed the use of Mask R-CNN for instance segmentation in the literature is far ranging, being utilised in medical [153–156], agricultural [157–160], sports [161–163], astronomical [164], and nautical [165, 166] fields. Alongside being well known, Mask R-CNN is also extremely reproducible. An official PyTorch implementation is available [167], though Matterport’s implementation is most commonly utilised when working with Tensorflow (including in this thesis) [168].

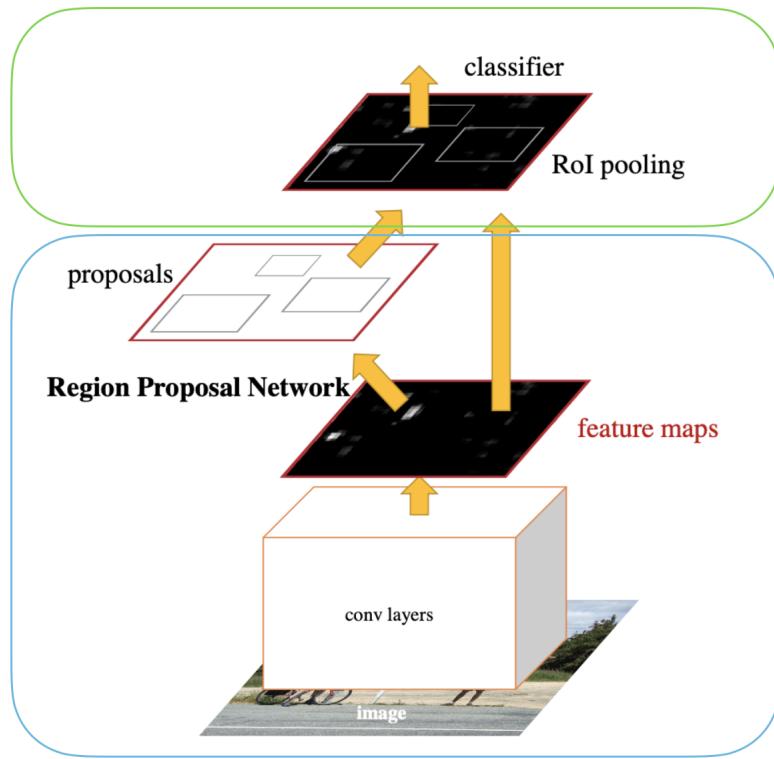


Figure 2.12 The Faster R-CNN architecture [106]. The blue box represents operations in stage one, which includes a standard backbone CNN architecture and RPN. The green box represents operations in stage two, performing RoI pooling and classification.

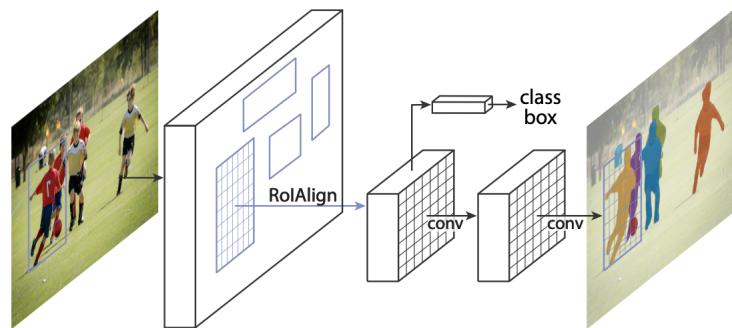


Figure 2.13 A visual representation of the changes made to stage two of Faster R-CNN to create Mask R-CNN. The ROI Pooling layer has been replaced with an ROI Align layer, along with the addition of a mask head. Image from [36].

### 2.4.6 Fine-Grained Visual Categorisation

Categorisation of objects through the use of machine learning may at first glance look like a solved problem. Indeed, it is now possible to achieve better than human levels of accuracy on a wide variety of tasks; at the time of writing the current state of the art for ImageNet [70] and CIFAR-10 [169], two of the most commonly used classification benchmark datasets, are both held by Foret *et al.* utilising EfficientNet with SAM [170] at 88.61% and 99.70% top-1 accuracy respectively. However, it is important to note that all of these tasks are coarse-grained in nature. Benchmark datasets usually contain classes which are relatively distinct, for example `cat`, `dog`, and `ship` classes in CIFAR-10, which all have large inter-class variation.

In contrast to the coarse-grained nature of the datasets above, fine-grained datasets are those with a small inter-class variation. Whilst CIFAR-10 has one class covering all different types of dog, the fine-grained dataset Stanford Dogs [171] is made up of 120 classes each containing examples of only one dog breed each (`chihuahua`, `beagle`, etc.). Other common fine-grained benchmark datasets often focus on wildlife or vehicles, including Caltech-UCSD Birds 200 [172] and the updated Caltech-UCSD Birds 200-2011 [173], and FGVC Aircraft [174].

Whilst fine-grained datasets may contain small inter-class variation, their intra-class variation can be relatively large (although not larger than the inter-class variation). Class examples may contain a wide variety of orientations, poses, colour, and sizes. This allows for trained models to generalise and be capable of detecting class examples in a wider variety of cases. It is also important to note here that models which perform well on coarse-grained data are not guaranteed to do so on fine-grained data. For example EfficientNet with SAM which, as previously stated, is state of the art in multiple coarse-grained tasks however ranks 31<sup>st</sup> in the FGVC Aircraft benchmark ranking on Papers With Code<sup>1</sup> at the time of writing.

### 2.4.7 Extreme Fine-Grained Visual Categorisation

Residing at the end of the scale of classification granularity is the task of extreme fine-grained classification, where the differences between dataset classes are minute. Developing extreme fine-grained datasets is extremely difficult, often requiring the input of domain experts to label class examples. Taking photo-id data as an example, at a coarse-grained level it would be enough to label all classes in a photo-id catalogue as `dolphin`. At a fine-grained level multiple classes may start to exist, such as splitting based on species. Building a useable

---

<sup>1</sup>Papers With Code – FGVC Aircraft Rankings: [paperswithcode.com/sota/fine-grained-image-classification-on-fgvc](https://paperswithcode.com/sota/fine-grained-image-classification-on-fgvc)



Figure 2.14 An example image from a photo-id catalogue with one individual present. At a coarse-grained level, this individual is classed as *dolphin*, at a fine-grained level *WBD* (white-beaked dolphin), and at an extreme fine-grained level 32, the individual's catalogue ID. Image and class labels from [40].

photo-id catalogue however is an extreme fine-grained task, where classes are split based on the individual.

Often there will be very few prominent markings which will allow for an individual to be classified, and these markings are often very small such as a notch in the fin or a scar; the vast majority of pixels in the classes will be very similar. Because of this, photo-id catalogues can only be accurately produced by local marine ecologists who have studied the resident cetacean population for many years. Even with this expertise however, creating an extreme fine-grained dataset such as a photo-id catalogue can be a large undertaking, often requiring many months of work to ensure all classifications are correct. An example highlighting the differences between coarse, fine, and extreme fine-grained recognition can be seen in Figure 2.14.

## 2.5 Computer Vision for Ecology

Thanks to the large advances in computer vision and deep learning, and the increasing prevalence of these systems in areas such as manufacturing and healthcare, researchers have, in recent years, begun exploring other areas of society which could benefit from AI systems. One of the more niche, but arguably highly important areas where computer vision can make an impact, is ecology [7].

Work into applying object detection and segmentation to ecology data mostly focusses on camera trap systems due to the large amount of data readily available. For example, the Snapshot Serengeti project<sup>2</sup>, developed by Swanson *et al.* has utilised camera traps in Tanzania’s Serengeti National Park to develop a fully labelled camera trap image dataset capable of training machine learning systems. The camera traps used have been in continuous operation since 2010 and cover an area of 1125km<sup>2</sup> [175]. The iWildcam dataset provides further camera trap training data from across the South-western United States [176].

Camera traps capture a photo every time movement in the frame is detected, and as such a large proportion of the images a camera trap captures either do not contain any animals at all (e.g. wind has caused the surrounding vegetation to move), or contain animals which are not the primary species of investigation. This provides a key driver for the development of machine learning camera trap systems which could, for example, filter out erroneous captures automatically. Whilst these images may simply be discarded by the researchers, they have a use in the development of machine learning camera trap based systems, allowing them to be trained on a wide variety of examples. As such, machine learning systems developed for camera traps have found quick adoption in the research community with many systems now capable of performing fine-grained species classification with extremely high accuracy [177–181]. Recent work by Clapham *et al.* [30] has moved further to the extreme of fine-grained classification with BearID, a project which adapts human facial recognition systems for use with brown bears (*Ursus arctos*) via metric embeddings, achieving an “individual classification accuracy” of 83.9%. Here, BearID is not classifying the species *Ursus arctos* but rather individuals within the survey area, a significant achievement given the challenge of identifying individuals within a species which lack unique markings.

As camera traps work through movement, taking a burst of images whenever the environment they observe changes – be this due to an animal walking through the scene or wind moving foliage – they must be kept stationary, usually mounted to grounded objects such as trees. These requirements make camera traps unsuitable for above-water marine environments, as the trap could not be attached to a stationary object at sea. Should it be possible to provide a stationary mount for the camera trap, the environment would still be unsuitable due to the rapid changing of the observed scene, due to factors such as waves, causing the camera to constantly produce image bursts. As a result, applying computer vision to marine environments is a greater challenge than on-land camera traps. This is also in part due to the relative lack of available data to train systems. Marine datasets such as FathomNet [182] are available, although its focus is on underwater detection and classification.

---

<sup>2</sup>Snapshot Serengeti: [snapshotserengeti.org](http://snapshotserengeti.org)

Marine ecologists traditionally rely on identification from photographs taken either from the coastline, a vessel at sea, an aircraft, or aerial drone. As this requires a human operator, the size of datasets available is relatively small. Furthermore, given the high cost of data collection, marine ecology groups often keep a tight grip on their data. This has led to a lack of available open-source datasets for those who wish to train machine learning systems for use in marine ecology. Thanks to advances in Unmanned Aerial Vehicle (UAV) technology and their current inexpensiveness, some research groups have shifted focus to the use of UAVs for image capture. This new approach has seen success in areas such as photo-identification [183, 184], microbial sampling [185], and human-interaction response monitoring [186]. However, some recently published work highlights the need to better understand how UAVs affect the behaviour and health of marine species [187–190].

### 2.5.1 Utilising Photo-id Aids in Cetacean Research

Performing manual analysis of photo-id surveys with the goal of creating a catalogue can be extremely time consuming and labour intensive. Images collected during fieldwork are brought back to the lab and filtered, removing images which do not contain cetaceans. The remaining images are then often pre-processed through a combination of manual orientation shifts and crops to centre and isolate the dorsal fin – any with more than one fin are split into multiple images at this stage. Next, the fins present in the images are identified by multiple researchers, either using an existing catalogue to compare against or building this using the data if one does not currently exist. These researchers are often experts in their field, having performed photo-id catalogue matching for many years, with multiple experts utilised to reduce the risk of misidentifications.

Due to the lengthy and expensive nature of manual photo-id, multiple aids have been developed over the years to ease the process. These have advanced at pace, gradually reducing the amount of human interaction and manual pre-processing of the data seen by these systems. A description of existing programs and academic literature is now provided, with an overview in Table 2.1.

#### Catalogue Management Systems

Due to the potential size of photo-id catalogues, especially in geographic locations with large populations, many aids focus on the management of these catalogues using databases. The first of these, FinScan, was developed in 2000 [191]. This is a semi-automated photo-id

---

<sup>3</sup>Photo-ID Ninja: [photoid.ninja](http://photoid.ninja)

<sup>4</sup>FlukeBook: [flukebook.org](http://flukebook.org)

Table 2.1 A comparison of available photo-id aids.

System	Catalogue Size (If Known)	Requires Data Pre-processing	Dorsal Fin Detection	Full Background Removal	Individual Photo-ID	Can Flag Individuals Not Currently In Catalogue	Uses All Information Found on Dorsal
FinScan [191]	190	✓	✗	✗	✓	✓	—
FinBase [192]	409	✓	✗	✗	✓	✓	—
DARWIN [193]	200	✓	✗	✗	✓	✓	—
catRlog [194]	—	✓	✗	✗	✓	✓	—
CurvRank [195]	3973	✓*	✗*	✗	✓	?	✗
Karnowski <i>et al.</i> [196]	—	✗	✓	✓	✗	—	—
Photo-ID Ninja <sup>3</sup>	—	✗	✓	✗	✗	—	—
Quiñónez <i>et al.</i> [197]	—	✗	✓	✗	✗	—	—
Morteo <i>et al.</i> [198]	533	✓	✗	✗	✓	✗	✗
Bouma <i>et al.</i> [199]	185	✓	✓†	✗	✓	✓	✓
Lee <i>et al</i> [200]	25	✗	✓	✓	✓‡	?	—
finFindR [201]	271\149*	✗	✓	✗	✓	✓	✗
Georgetown University & Google [202]	>1800 <sup>▽</sup>	✓	✗	✗	✓	✗	✗
DolFin [203]	60	✗	✓	✓	✓§	✓	✓
Ours	43\23 <sup>▷</sup>	✗	✓	✓	✓	✓	✓

\* Flukebook<sup>4</sup> can perform automatic data pre-processing and fin detection before passing the output to CurvRank, but the algorithm itself does not facilitate this.

? It is unclear whether the system is capable of flagging previously uncatalogued individuals.

† Utilises Photo-ID Ninja<sup>3</sup> for detection.

‡ Sparse on detail regarding photo-id performance.

§ Evaluation utilises “full” (n = 271) and “reduced” (n = 149) datasets.

|| Individuals not present within the top-50 most likely are considered be previously uncatalogued.

▽ Exact number not given.

▷ Utilises SURF for photo-id, thus unsuitable for cetacean species without well-defined markings.

▷ Most likely catalogue matching evaluated using the NDD AU SMRU (n = 43 plus noise), see Section 6.4, and the SDRP datasets (n = 23), see Section 7.3.

assistant whereby the user imports images taken during fieldwork. FinScan then attempts to create a trace of the fin in the image. Users may manually edit this trace however if it is not exact (this feature was developed due to frustration with barnacles attaching to fins in the area where FinScan was developed). The trace of the fin is then checked against a local Microsoft Access database to determine close matches which are presented to the user. Before images are imported into FinScan they must be manually cropped, sharpened, and rotated by the user. Rotation of the image is especially important, as the FinScan algorithm is not rotation invariant. Whilst FinScan is freeware it is no longer readily available. Anyone who wishes to use it must procure a copy from someone else, there is no central repository for downloading. Issues with running the software on newer systems may also be present.

Similarly to FinScan, FinBase is a photo-id database management system developed by NOAA Fisheries [192]. However, unlike other systems, FinBase provides no matching based on automatically generated fin properties; instead, FinBase facilitates matching through user defined attributes. These could be physical descriptors such as ‘top notch’ or ‘skin disorder’ but may also be non-physical attributes if the user wishes. Fins are partially matched based on querying the backend database for entries which also have the attributes of those inputted by the user for the query fin.

Alongside both of these, DARWIN [193] provides automated identification of new images based on those already in the attached database. Like FinScan, users of DARWIN trace around the leading and trailing edges of the fin they wish to identify. These edges are stored in a database as a set of evenly spaced points approximating the outline of the fin which is then used for identification.

CatRlog [194] is an R Shiny based catalogue management system which allows for photo-identification of individual cetaceans. Researchers enter descriptions manually for unique markings located on either the dorsal fin or fluke of the individual. One of catRlog's main advantages is the ability to be hosted locally on machines via RStudio whilst also being capable of handling catalogues hosted on cloud-based storage solutions. This allows for efficient id verification, as multiple experts can id images at the same time each uploading results to the central catalogue. If any discrepancies arise, such as an individual given a different id by multiple experts, this centralised approach allows for the catalogue manager to easily determine the final id and disseminate this to other catalogue users. CatRlog is also capable of automated print-friendly photo-id catalogue creation.

### **Classical Machine Learning Systems**

Multiple systems which utilise classical machine learning are also available to aid cetacean researchers. Karnowski *et al.* propose using Robust Principal Component Analysis (PCA) to subtract background from underwater images to help detect captive bottlenose dolphins and track their movements through multiple distinct areas [196]. CurvRank is an algorithm developed by Weideman *et al.* [195] which automatically identifies the trailing edge of the fin and represents this as a set of ordered coordinates. Each coordinate point then has a circle of radius  $r$  placed upon it, before being transformed horizontally. The curvature at this point for a given  $r$  value is then defined as the ratio of the area under the curve against the area of a square around the curve of length  $2r$ . This allows for the definition of the trailing edge of the fin to be rotation invariant. Whilst initially developed for cetaceans, CurvRank has also found use in land-based photo-id surveys due to its robustness. For example, Kulits *et al.* [204] utilise CurvRank and SEEK [205] as the basis for a human-in-the-loop system for African elephant (*Loxodonta africana*) photo-id.

### **Deep Learning Systems**

In recent years the use of deep learning has been explored, inching forward towards a fully automatic photo-id system. One of the main labour costs in the creation of photo-id catalogues is the processing of survey data. This is usually cropping the images collected down to just the fins in the image, removing unneeded background. Photo-ID Ninja<sup>3</sup> is

designed to speed up this processing. Users provide a batch of images taken directly from the field to the system, which utilises an object detection model to output bounding boxed dorsal fins which can then be manually identified.

Quiñonez *et al.* propose a CNN based system capable of cetacean object detection, detecting four distinct classes: `dolphin`, `dolphin_pod`, `open_sea`, and `seabirds` [197]. Morteo *et al.* [198] perform semi-automatic fin measuring using fin shape in order to aid in population monitoring. This technique, based on work by Weller [206], does not require the user to trace the fin; instead lines are projected out from the base of the leading edge of the fin, with the user cutting these lines where they intersect with a point on the trailing edge.

Bouma *et al.* provide a system focusing on learned metric embeddings in order to photo-id individual New Zealand common dolphins (*Delphinus spp*). This research utilises Photo-ID Ninja to detect and crop fins before they are passed for identification, focussing on common dolphins as data subjects. The system is capable of achieving top-5 accuracy scores of around 93% [199], although it should be noted here that the data utilised to achieve this score is not currently publicly available. An attempt to obtain this data was undertaken, but no response was received. As such, it is not possible to determine how distinct each individual is in the dataset. Based on the figures presented in the paper, it seems no segmentation of the cropped fins is performed before they are embedded. As such, some noise in the embeddings will be present.

FinFindR is an algorithm developed by Thompson *et al.* [201] which allows for inputted images containing bottlenose dolphins to be identified. FinFindR works with uncropped images and automatically detects any dolphin present and crops it out, saving a new image. Cropping can be performed on either the whole body or on the dorsal fin only. This cropped image is then passed to a Canny edge detector which creates an embedding of the trailing edge of the fin. This embedding is mapped into a high dimensional space based on work presented in FaceNet [38], with clustering of individuals achieved using Ward's variance minimising clustering [207]. Reported accuracies for finFindR currently stand at a top-1 accuracy of 88%, top-5 accuracy of 94%, and top-50 accuracy of 97%. It should also be noted that finFindR has currently only been tested on bottlenose dolphins and work is still ongoing.

Work undertaken by Georgetown University and Google in the area of cetacean photo-id has also provided promising results [202]. The system, which utilises Google's Cloud Auto ML framework, can quickly identify bottlenose dolphins from Australia's Shark Bay. This system shows users the top-200 closest matches along with their confidence scores utilising both the leading and trailing edges of the fin. It is reported that this system saves Georgetown University's cetacean team around 4500 hours per year, highlighting the need for systems

such as these to researchers in this field. However, this system does not link to a backend database to log matches found – this must be done manually by the researchers. Further, any new individuals that need to be added to the system, or indeed if the system was to be redeployed to a new area, then all training of the underlying model must be performed by Google engineers rather than locally by the researchers who wish to utilise the system. Further, fins to be identified must be inputted one by one, no batch input function exists.

Recent work undertaken by Lee *et al.* proposes a novel architecture for cetacean identification [200]. The proposed system is capable of detecting small objects in large images, and utilises this for fin detection. Next, segmentation is performed using U-Net [208]. The resultant output is then passed to a post-processor which re-aligns and normalises the fin. The most significant features of the fin are then extracted and passed to a VGG based system [152] combined with a novel V2BC component for identification, although results achieved for identification in the paper are sparse.

Maglietta *et al.* propose DolFin [203], a Speeded-Up Robust Features (SURF) based identification system [209] built upon in work undertaken by Renò *et al.* [210] for identifying individual Risso’s dolphins. As mentioned in Section 2.1, this species is susceptible to prominent long-term scarring, and thus is well suited to feature detection algorithms such as SURF, with published results showing a much greater identification accuracy can be achieved compared to utilising common photo-id aids such as DARWIN [193]. DolFin also makes use of a “fin mask extraction” unit capable of detecting, segmenting, and post-processing fins before they are identified using SURF, although details on this unit are unclear. This system is truly fully autonomous, although due to the SURF based system would only be capable of working with Risso’s dolphins. Other cetacean species do not develop as prominent scarring, which would greatly reduce the effectiveness of SURF-based feature extraction systems.

## Online Tracking Systems

Whilst the use of photo-id catalogues for resident populations is beneficial for researchers in the animal’s local area, some cetacean species traverse large geographical distances, sometimes travelling between continents. In these cases it would not be feasible for one research group to record sightings. This is where online tracking systems play an important role. These websites started by allowing users to enter their own sighting data into an online database, allowing for tracking of individuals over large areas. In recent years however the focus has shifted to performing detection and identification online from unlabelled sighting images. This allows groups of users such as citizen scientists to contribute to the database, as no knowledge of the existing catalogue or training in photo-id is required.

HappyWhale<sup>5</sup> is a CNN based photo-id system focussing on humpback whales (*Megaptera novaeangliae*). The underlying CNN for this system was developed through a Kaggle competition<sup>6</sup> by user Jinmo Park to identify patterns present on the tailstock of the humpbacks [211], utilising elements of ArcFace [212] and DeepFace [213] to do so. Users interact with HappyWhale through their website, uploading images of the tailstocks encountered. The HappyWhale system then attempts to identify the individual before presenting back to the user. If the user provides location data, HappyWhale also keeps track of this to produce travel maps for the individuals, as humpback whales are known to travel vast distances in their lives. The success rate for HappyWhale varies greatly, from 99% for “good to high quality” images to 50% for full fins at 50x50px. HappyWhale struggles with partially obscured tail stocks however, and work is currently ongoing in this area [59].

FlukeBook<sup>4</sup> is a fully automatic photo-id system capable of identification of multiple cetacean species. This system is part of a wider network of animal identification tools based on Wildbook, an open source software framework developed by non-profit organisation WildMe to facilitate the introduction of artificial intelligence into the ecology space [214]. FlukeBook makes use of both CurvRank and finFindR when working with bottlenose and spotted dolphins (*Stenella frontalis*). Whilst FlukeBook is capable of detecting and identifying individual cetaceans, the fact that it is hosted online with all submissions freely searchable may be a disadvantage for some cetacean researchers. Local photo-id catalogues are closely guarded due to the effort and expense required to collect the data, and as such some groups may prefer a local automated photo-id solution over one which requires them to hand over their data to a third party.

### Summary of Available Photo-id Aids

As can be seen in Table 2.1, there is a wide variety of photo-id aids currently available to researchers – each fulfilling a different need depending on the researcher and how comfortable they are with automating parts of the photo-id process. If they prefer a fully manual approach to analysis but require storage solutions, FinScan and FinBase provide excellent catalogue management. DARWIN can then extend this through automating existing catalogue databases.

Manual analysis is extremely time consuming however, leading to modern systems allowing for some automation of processing. For fast cropping, Photo-ID Ninja can be utilised, greatly reducing the sizes of images and removing unneeded background. Algorithms like CurvRank can then be utilised to aid in identification.

---

<sup>5</sup>HappyWhale: [happywhale.com](http://happywhale.com)

<sup>6</sup>Kaggle competition: [kaggle.com](https://www.kaggle.com)

Whilst all of these systems perform their intended task well, they are mostly stand-alone. This would require researchers to make their own data pipelines should they wish to host locally, such as Bouma *et al.* and their use of Photo-ID Ninja for detection before embedding using deep learning to provide catalogue matching [199]. Should researchers feel comfortable with open-sourcing their catalogues and matching process, whilst also working with a supported cetacean species, then systems like HappyWhale or FlukeBook are most appropriate, requiring little data pre-processing whilst not requiring researchers to host multiple algorithms or solutions locally, reducing the need to create their own custom data pipelines.

It is the work undertaken by the aforementioned solutions that motivates this thesis. A fully automated photo-id system must be capable of the following:

1. **Operating with no data pre-processing:** Researchers should not need to process data from fieldwork cameras before it can be handled by the system.
2. **Detection:** The system must be capable of detecting dorsal fins present in input images. This allows researchers to pass the system all images taken in the field to reduce human workload and allow the system to operate on data which has not been pre-processed. The system must be capable of detecting multiple fins in the same image.
3. **Segmentation:** Once fins have been detected, it must be possible to segment them from the image. This aids the photo-id process by removing other fins, which may contain prominent markings, as well as background such as the sea, which may be feature heavy due to waves.
4. **Photo-ID:** The system must be capable of identifying individuals in a species agnostic manner.
5. **Flagging uncatalogued individuals:** The system must be capable of flagging individuals that have not yet been catalogued to the user.
6. **Locally deployable:** It must be possible to deploy the system on local hardware, allowing researchers to keep full ownership of their catalogue.

Of the works highlighted in Table 2.1, only DolFin [203] proposes a fully automated photo-id aid which can be locally run, requires no data pre-processing, and is capable of handling uncatalogued individuals. However as previously stated this solution utilises SURF [209]. This algorithm, like its predecessor Scale-Invariant Feature Transform (SIFT) [215], excels at detecting well defined features in images. This is appropriate for DolFin's data

subjects, Risso's dolphins, due to the well defined scarring patterns present on their dorsal fins [60]. The algorithm would be expected to fail when used on other cetacean species however due to a lack of well defined features on the fin, where even prominent markings used for photo-id fail to be extracted. In contrast, this thesis makes use of a robust embedding generation technique through the use of a trained Siamese Neural Network (SNN), allowing for the extraction of features in a species agnostic way. See Section 4.2.1 for an investigation into the use of feature extraction algorithms on bottlenose dolphin dorsal fins, as well as Sections 6.3 and 7.3 for discussion of the use of SNNs to extract identifiable features from catalogues containing different species of interest collected over a range of spatio-temporal environments.

According to Tyson Moore *et al.* [35], finFindR is the most commonly used photo-id aid. However, as previously mentioned, this system has only been evaluated in a peer-reviewed context using data containing bottlenose dolphins and makes use of identifying information found only on the trailing edge of the dorsal fin. Further, flagging potentially uncatalogued individuals is performed implicitly, with input images considered previously uncatalogued if a match is not obtained with any of the top-50 most likely matches – assuming a similar data distribution to the catalogue used to evaluate system performance [201]. This may cause issues when working with catalogues containing significantly more than 50 individuals. In contrast, this thesis evaluates performance utilising multiple different dolphin species, makes use of all available individually identifiable markings, and explicitly states to researchers when a given input is potentially uncatalogued, allowing for a more informed decision irrespective of catalogue size.

Work by Bouma *et al.* [199] also makes use of embedding generation to perform most likely catalogue matching of New Zealand common dolphins from a single geographic location only. Further this work is not stand alone, requiring users to make use of dorsal fin detectors such as Photo-ID Ninja beforehand. These detections may require manual post-processing to be utilised. As such, work presented in this thesis extends the literature by providing a full data pipeline which is shown to be robust to changes in species of interest and spatio-temporal changes. Alongside this, work outlined in Section 7.2 provides evidence to suggest that most likely catalogue matching can be influenced greatly by retained background features. As images utilised by Bouma *et al.* retain their background, it may be susceptible to inflated evaluation accuracy.

## 2.6 Summary

This chapter presents the key ideas required to understand and appreciate the novel work proposed in this thesis. An overview of deep learning and computer vision is provided, as well as an introduction to photo-id, a key methodology for mark-recapture surveys utilised by marine ecologists. A summary of previous research combining ecology and deep learning has also been provided, with a focus on marine environments and cetaceans.

Before development on the system components required by an automated photo-id aid can be undertaken, datasets suitable for the training and evaluation of these components must be created. This is outlined next in Chapter 3.



# Chapter 3

## Initial Dataset Creation

Before the development of an automated photo-id aid could begin, the procurement of large scale data with a distribution similar to that seen during photo-id surveys was required. Exploration of available open-source datasets to find images of cetaceans in similar conditions to those seen in photo-id survey data proved unfruitful. Many standard benchmarking datasets contain animal classes, and thus an exploration of these was conducted. Of the more generalised benchmark datasets, those such as ImageNet [70] which contain a large corpus of varied classes, only CIFAR-100 [169] contains a dolphin class. However, images in CIFAR-100 are only 32x32 pixels in size, too small to be useful for the task at hand.

Moving the search away from generalised datasets and towards those which are targeted at conservation efforts or the natural environment also proved fruitless. A large portion of these datasets focus on camera traps or land-based fauna, such as iWildCam [176], for reasons discussed in more detail in Section 2.5. Some images included in the iNaturalist dataset [216] are of cetaceans, such as a class for the short-beaked common dolphin (*Delphinus delphis*), however most focus on other aquatic animals such as the Florida manatee (*Trichechus manatus*), various amphibians, and molluscs.

As no open-source datasets containing imagery required for the work in this thesis were available, this chapter highlights the work undertaken to develop these. The creation of a coarse-grained instance segmentation dataset consisting of readily available survey data collected around the waters of Zanzibar, Tanzania is first discussed. Next, the creation of a dataset consisting of imagery collected around the waters of Northumberland, UK, known as the Northumberland Dolphin Dataset 2020 (NDD20), is outlined. This dataset contains coarse-grained instance segmentations, as well as fine-grained species and individual level classifications.

### 3.1 The Zanzibar Dataset

Due to the lack of open-source datasets to aid in the development of the cetacean detector, a photo-id catalogue curated during a 2015 research effort undertaken in Zanzibar, Tanzania by Newcastle University’s Marine MEGAfauna Lab was obtained. This catalogue was utilised to determine the status of Indo-Pacific bottlenose dolphins (*Tursiops aduncus*) in the area [10], and consisted of 1021 images of size 5184x3456 – supplied in a format suitable for manual photo-identification rather than for the training of a neural network. Work was then undertaken to convert this catalogue into a machine learning dataset.

In order to perform this conversion, the provided images were first labelled. This was achieved using the VGG Image Annotator software (VIA) [217]. Other labelling software such as LabelImg [218] were examined, however VIA was deemed the best choice for the task at hand. This software was chosen for multiple reasons; first, the software is noticeably easy to use and allows for efficient labelling on a per-pixel basis as required by Mask R-CNN. Second, the tutorial data provided by the Mask R-CNN Github repository was labelled in VIA format. Furthermore, use cases of VIA being utilised for labelling of marine-oriented data are available in the literature [219], providing evidence of suitability of the labelling software for research purposes and data representing similar conditions.

Before labelling the Zanzibar data, some curation was performed. Each image labelled by VIA is required to contain at least one non-background class. As such, any images provided which did not contain an example of a `dolphin` class were discarded. Other images where the class examples were unsuitable for training a Mask R-CNN model, such as those which contained only an extremely small section of the photographed dolphin or were deemed too blurry, were also removed. This left 312 images which were suitable for the Mask R-CNN.

The process for labelling the data with VIA is rather straightforward. The software runs locally through a web browser, with each image labelled sequentially. Figure 3.1 shows an example image labelled using VIA. Each image is shown on-screen to the user who is then able to trace around class examples by selecting multiple points on the image. Once a full trace has been performed, any pixels inside of the trace are treated as one class. This class is labelled through the use of a class attribute, in the case of the Zanzibar data this was the class label `dolphin`, denoting the class example as an animal breaching the waterline. These labels are stored in a corresponding JSON file, which is fed to the Mask R-CNN model along with the images during training. This labelling allows the model to learn per-pixel class examples during training. This tracing method also allows for each distinct individual in a group to be labelled individually, even if overlapping, which would be much harder

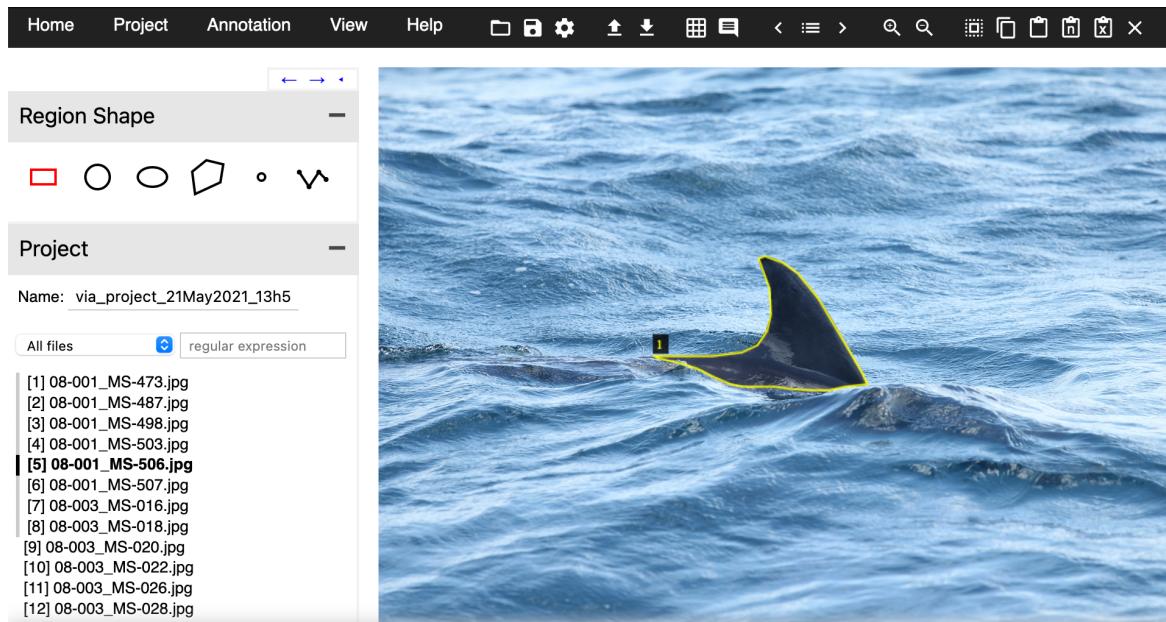


Figure 3.1 An example of labelling process using VIA. The dorsal fin in the image selected has been highlighted by the yellow outline.

to perform with bounding box labelling and allows the model to learn how to differentiate between group members.

## 3.2 The Northumberland Dolphin Dataset 2020

Whilst the creation of the Zanzibar dataset allows for the development of a model capable of coarse-grained automated detection of cetaceans from photo-id imagery, as no fine-grained individual IDs are present it does not contain the information required to allow for the training of a model capable of individual identification. As a result, this section discusses the collection of abundance estimate data off the coast of Northumberland, UK. The collected photo-identification data was then curated and transformed from an ecological catalogue into a fine-grained computer vision dataset known as The Northumberland Dolphin Dataset 2020 (NDD20), to allow for the development of an automated catalogue matching system to occur.

Surveys undertaken and described within this section also provided data for two other theses at Newcastle University, one within the School of Electrical and Electronic Engineering focussing on the automated identification of cetaceans via signature whistles and the other within the School of Natural and Environmental Sciences focussing on the creation of abundance and health assessments of the cetaceans resident in the survey area.

### 3.2.1 Data Collection

The following section provides context for the data collection survey, beginning by briefly outlining the geographic area in which the data was collected and discussion of why the area was chosen. Next the survey effort is discussed in detail, including a run-down of the methodology used, for the purposes of reproducibility.

### 3.2.2 The Survey Area

Whilst the Marine MEGAfauna Lab conducts research heavily in the Indian Ocean around Zanzibar [220–225], this is not the only place they operate [226–228]. In recent years, their work has begun to include more local waters such as the North Sea off the coast of Northumberland, UK [11, 229]. These waters are known to host a wide variety of marine mammals, with the Marine MEGAfauna Lab focussing efforts specifically on the common bottlenose dolphin (*Tursiops truncatus*) and white-beaked dolphin (*Lagenorhynchus albirostris*) populations.

Data collection was conducted in and around the Coquet to St. Mary's Marine Conservation Zone (MCZ), located off the coast of Northumberland, UK. The MCZ, established in January 2016 through powers granted by the Marine and Coastal Access Act 2009 [230], covers approximately 40km of coastline from Alnmouth in the north to Whitley Bay in the south, extending outwards 7.5km at its greatest to cover an area around 192km<sup>2</sup>. A map of the survey area and MCZ can be seen in Figure 3.2.

The area is of high ecological importance, supporting a wide variety of marine life thanks to sections of intertidal and sub-tidal rock and sediment, making it fertile feeding grounds for the bottlenose and white-beaked dolphins which make use of the area. As a result of this fertility, as well as waters up to 30m deep in some places, the MCZ sees high levels of fishing activity – typically for crustaceans using pots [231]. Whilst these fishing vessels operate from a number of small ports throughout the North East of England, the MCZ itself lies close to the large Port of Blyth. As a result, the MCZ boundary provides a 250m buffer zone around the limits of the port in order to reduce economic damage. This survey region was selected as no previous surveying had been undertaken in the area for the purposes of cetacean abundance estimates and health assessment.

### 3.2.3 Survey Effort

Dedicated bottlenose and white-beaked dolphin photo-identification surveys were conducted in the MCZ between 19/07/2019 and 10/10/2019, with a total of 27 surveys undertaken.

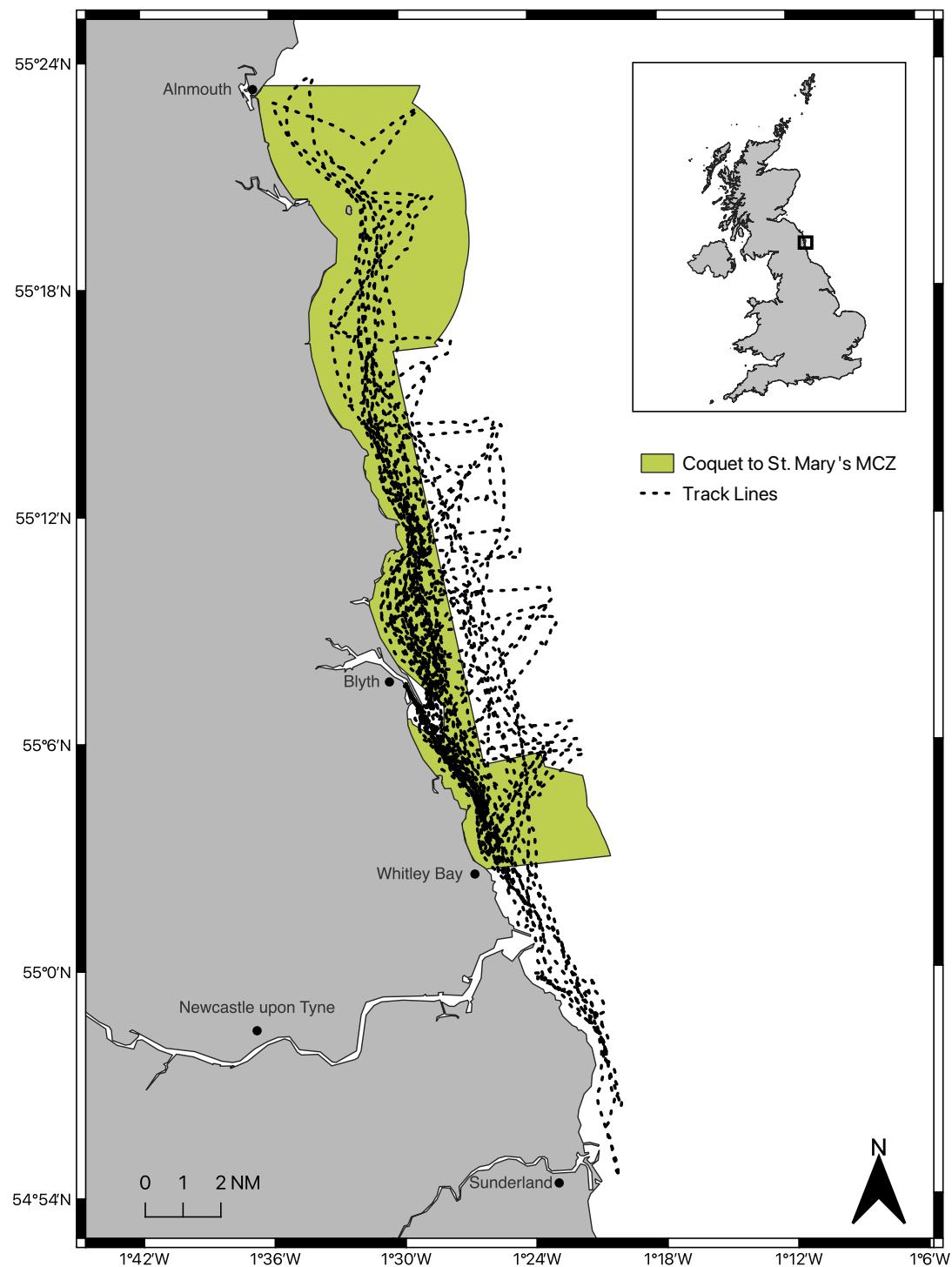


Figure 3.2 Map of the survey area, with the Coquet to St. Mary's MCZ highlighted. Track lines for all survey days are overlaid.

These were performed using a 5.6m rigid inflatable boat (RIB) with a 50 horsepower four-stroke outboard engine. All surveys began from Newcastle University's Blyth Marine Station, located in the Port of Blyth, before entering the MCZ.

Surveys initially began by following set transect lines, traversing between the northern and southern-most points of the MCZ. Thanks to limited success encountering individuals strictly following transect lines however, the survey switched to more opportunistic surveying based on reports from two citizen science groups: the Newbiggin-by-the-Sea Dolphin Watch<sup>1</sup> and the North East Cetacean Project<sup>2</sup>. The use of citizen scientists for photo-id surveys has seen increased prevalence in recent years, with multiple studies producing promising results if access to groups of dedicated citizen scientists is available, as in Northumberland [232–235]. Track lines showing movement of the vessel were recorded via GPS tracking, and can be seen in Figure 3.2. When dolphins were encountered, the time stamp was recorded alongside other effort data such as direction of travel, sea state, species, group size, and demographic composition.

Surveys were only conducted in Beaufort Sea States < 4 [236] without heavy rain. Outside of these conditions surveying can become unsafe and the photographs unusable for photo-id because of swell and lens splash. Due to the nature of the North Sea, conditions outside of these restrictions can be common. Surveying was performed using the constant scanning method [237], with cues including sight of dorsal fins breaching the waterline, splashing, and leaping. For each survey the vessel was manned by at least two dedicated observers and a skipper, in line with other photo-id surveys [10, 238, 239].

Individual dolphins in an encounter were photographed randomly using a Canon EOS 550D Digital SLR with a Canon 70–200mm zoom lens, aiming to capture photographic data for every individual present. Camera settings can be found in Appendix B. Multiple photographs were captured of each cetacean over the course of the encounter to ensure identifiable information could be fully captured.

When capturing an encounter care was taken not to approach individual cetaceans at an angle less than 30°, keeping as parallel as possible and to speeds no greater than 6 knots in order to prevent the cetaceans becoming stressed or injured as per Marine Management Organisation guidelines. All members of the survey team were trained in minimising wildlife disturbance through the WiSe Scheme by the Yorkshire Wildlife Trust<sup>3</sup>, with the survey itself having the approval of Newcastle University's Ethics Board.

---

<sup>1</sup>Newbiggin-by-the-Sea Dolphin Watch: [facebook.com/groups/NEWILDDOLPHINMONITORINGPROJECT](https://facebook.com/groups/NEWILDDOLPHINMONITORINGPROJECT)

<sup>2</sup>North East Cetacean Project: [facebook.com/groups/NorthEastCetaceanProject](https://facebook.com/groups/NorthEastCetaceanProject)

<sup>3</sup>WiSe Scheme: [wisescheme.org](http://wisescheme.org)

### 3.2.4 Field Season Summary

In total, 27 surveys were conducted with 14 containing encounters. Of these, 12 were made up of bottlenose dolphins; only two were made up of white-beaked dolphins. No encounters contained both species. Groups were defined using the 10m chain rule [240]. Group size averaged 12 for bottlenose dolphins, typical for the species [241]. Altogether 44 individuals were identified and catalogued, broken down into 30 bottlenose and 10 white-beaked dolphins. Of all animals encountered, 27% were calves. They have been excluded from this analysis as they could not be considered independent due to reliance on their mothers, and had not yet developed permanent markings.

Images collected were processed for use in the photo-identification catalogue to remove any images with no value, such as those which were out of focus or did not contain any cetacean. Animals present in the images were coded according to their distinctiveness as per the guidelines presented by Urien *et al.* [18]. Those coded D1 were considered very distinctive with little chance of misidentification, whilst those coded D2 were considered moderately distinctive with small prominent markings which could allow for a high chance of correct classification provided the image is clear. CF coded individuals were those that contained little to no identifying information and have a high chance of misclassification. Once coded, animals considered D1 and D2 were individually identified.

## 3.3 Creation of NDD20

The fieldwork season and data processing undertaken resulted in a photo-identification catalogue of bottlenose and white-beaked dolphins currently inhabiting the Coquet to St. Mary's MCZ. Photo-identification catalogues utilised in marine ecology however are not in the form required for training or validating a computer vision model. As such, further processing of the catalogue was required. This section discusses the creation of the Northumberland Dolphin Dataset 2020 (NDD20) [40], the computer vision dataset created from the photo-identification catalogue collected in the MCZ.

### 3.3.1 Above Water Data

During fieldwork 4940 images were collected which contained part of a cetacean above the water line. Of these, 2201 images were considered usable for the creation of NDD20. Issues rendering images unusable included a significant amount of water splash obscuring the cetacean, poor lighting conditions, or where individuals in a pod were too close together to accurately determine by eye the outline of all individuals.

From manual analysis it was determined that not all images contained enough identifying information for individual classification labels. Because of this, the decision was made to include multiple levels of granularity to the dataset. As all images contained part of a cetacean, each one could be labelled to allow for instance segmentation training. To enable this, each mask located was given the label `dolphin`. Next, masks could be provided a fine-grained species classification. Thanks to the difference in colour between bottlenose and white-beaked dolphins, every mask labelled for instance segmentation could also be provided a species label – either `BND` or `WBD` representing bottlenose and white-beaked dolphin respectively.

At the highest level of granularity, some masks contained enough information to allow for individual identification. If an ID could be attained with high confidence, likely from images with D1 or D2 coded individuals, the mask containing the individual was provided an `ID` label. Recent work has shown that publicly available datasets containing animals may aid poachers [242]. In response, to protect ongoing cetacean research efforts a pseudo-anonymisation has been performed. This however does not diminish the value of the dataset to computer vision researchers. It is not the case that images with sequential filenames were captured sequentially, and all individual IDs have been randomly allocated a numerical value rather than the code given to them by the Marine MEGAfauna Lab. All EXIF data found in the images has also been removed.

Data was labelled at a pixel level using the VGG Image Annotator [217] in a similar fashion to the Zanzibar dataset as discussed in Section 3.1. In order to speed up the process data labellers were employed through Newcastle University’s Jobs On Campus service<sup>4</sup> to manually annotate the masks and label them for instance segmentation. Once complete all masks were checked for error correcting and consistency purposes. Afterwards, the extra labels for both species and individual level identification were added. As this required expert knowledge, data labellers were not utilised for this. Example above water images from NDD20 and the labels assigned can be seen in Figure 3.3.

### 3.3.2 Below Water Data

In addition to the above water imagery captured during fieldwork, NDD20 also contains below water images. Underwater photo-id is not as widely used compared to its above-water counterpart at present, however uses have been noted for certain species and environments in recent years [11, 55]. Whilst this data has not been utilised in the work on automatic photo-id presented in this thesis it is important to discuss all aspects of the dataset created. Images

---

<sup>4</sup>Newcastle University Jobs On Campus: [ncl.ac.uk/careers/jobs/opportunities-on-campus/jobsoc/](http://ncl.ac.uk/careers/jobs/opportunities-on-campus/jobsoc/)

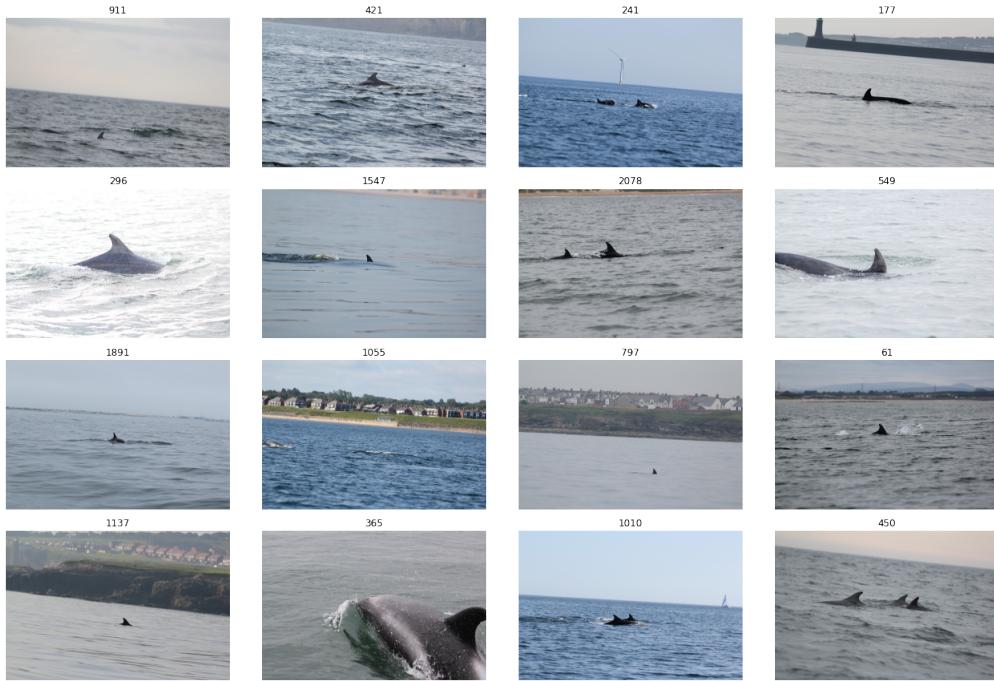


Figure 3.3 Example above water images from NDD20 with filenames displayed. Class labels for masks in each image are noted in Appendix C.

contained in the below water section of NDD20 are a subset of a much larger collection of images produced by the Marine MEGAfauna Lab during work in the Farnes Deep MCZ, a glacial trench situated approximately 11km from the Northumberland coast. Opportunistic surveys undertaken since 2011 have shown the area to contain a high abundance of white-beaked dolphin activity [11]. Data from these surveys takes the form of screen grabs from high definition video footage captured by a diver using GoPro Hero 3 and Go Pro Hero 4 cameras.

To mirror the above water section, there are 2201 below water images included in NDD20 labelled for multiple levels of granularity. As before, the first attribute level is `dolphin` to allow for instance segmentation. Unlike the above water images, all below water images contain at least one mask with an `ID` label. It is not the case that masks in the above and below image sets contain the same individual animal even if they have the same `ID` class label – the numbering systems are independent of one another. No species label is provided as all images are of white-beaked dolphins. Below water images are also labelled with an `out of focus` flag, denoting if the individual is deemed to be out of focus. Example below water images from NDD20 can be seen in Figure 3.4.

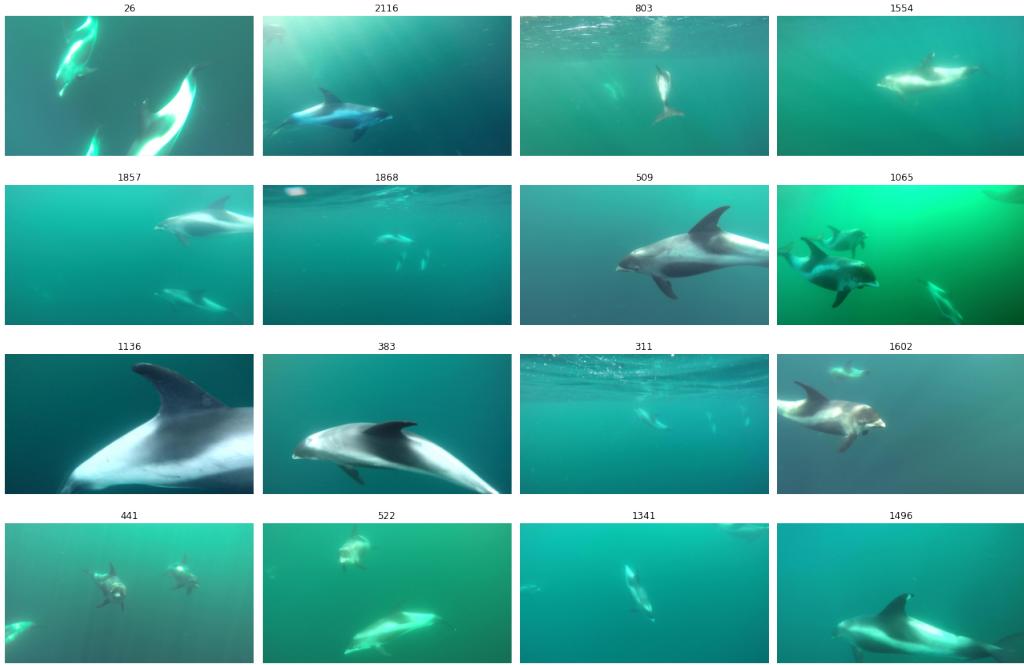


Figure 3.4 Example below water images from NDD20 with filenames displayed. Class labels for masks in each image are noted in Appendix D.

### 3.3.3 NDD20 Summary

As NDD20 is split into two related but distinct sets of images, a summary is provided below for each. Due to the nature of cetacean group dynamics, multiple images contain more than one individual animal. As a result, there are 2900 masks present in the above water set of 2201 images. These masks contain both a `dolphin` label for instance segmentation as well as either a `BND` or `WBD` label to facilitate species level fine-grained classification. It should be noted however that the distribution of species class labels is imbalanced, with 73% of masks being labelled `BND`. Some above water masks also contain an individual level `ID` label to allow for extreme fine-grained classification. Due to the nature of the task only 14% of masks contain an `ID` class label, with 44 distinct individuals present. Once again these classes are imbalanced presenting both a fine-grained and few-shot learning problem. The above water `ID` class label distribution can be seen in Figure 3.5. Many of the challenges associated with manual above water photo-id apply here too, particularly the likelihood that unique features are specific to one side of an animal's body which may not have been captured in the image.

Like its above water counterpart, the below water set also contains 2201 images each with at least one mask containing a `dolphin` label. Masks in the below water set are significantly larger than in the above water set, as far more of the cetacean is visible when captured below

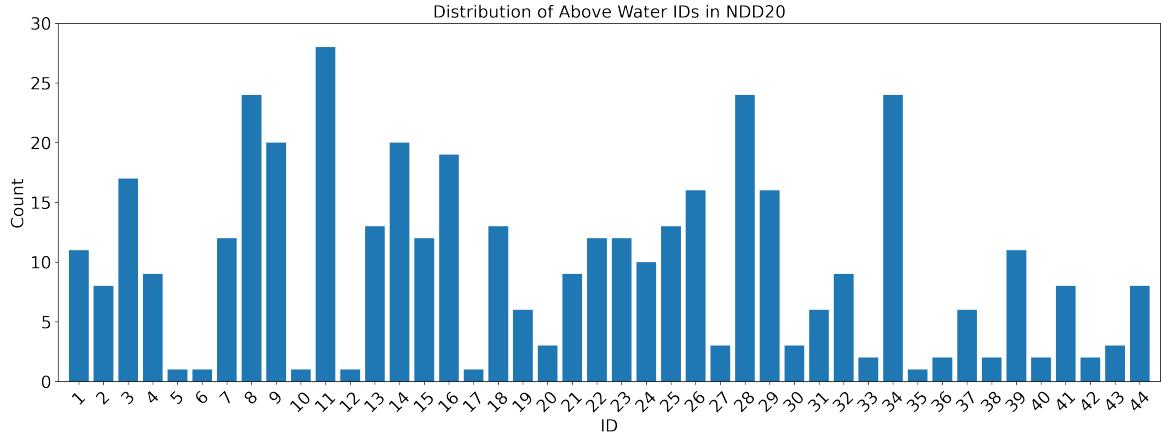


Figure 3.5 The ID class label distribution for the above water set of NDD20.

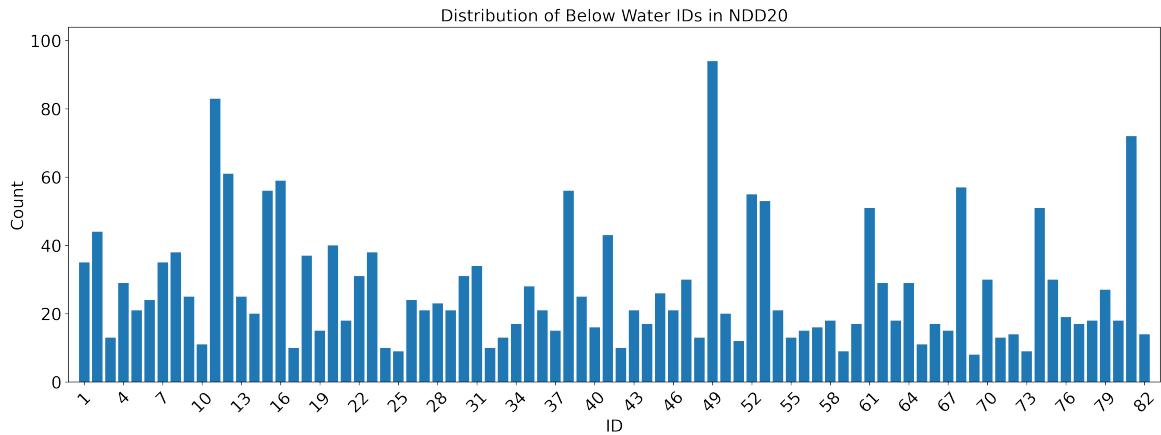


Figure 3.6 The ID class label distribution for the below water set of NDD20.

the waterline. Unlike the above water set however, all below water images also contain at least one mask with an ID class label, with 82 classes represented. The distribution of ID class labels in the below water set can be seen in Figure 3.6. This set represents a significantly more challenging fine-grained and few-show learning problem, both due to the higher number of classes as well as decreased image quality thanks to the nature of underwater photography. The main challenges are water clarity, affected by factors such as algal bloom and sunlight refraction which may both obscure areas of the individual useful for identification or add artificial markings which may hinder this.

NDD20 is small compared to traditional benchmarking computer vision datasets such as ImageNet [70] or those which are domain specific such as iWildcam [176] or Caltech-UCSD Birds 200 [172]. Even when compared to other related non-benchmark marine ecology

Table 3.1 A comparison of computer vision datasets capable of training models for individual animal identification, sorted by number of individuals.

Dataset	Species	Number of Images	Number of Individuals
Beluga ID 2022 <sup>†</sup>	Beluga whale ( <i>Delphinapterus leucas</i> )	5902	788
Leopard ID 2022 <sup>‡</sup>	African leopard ( <i>Panthera pardus</i> )	6795	430
Hyena ID 2022 <sup>§</sup>	Spotted hyena ( <i>Crocuta crocuta</i> )	3104	256
Cows2021 [245]	Holstein-Friesian cow ( <i>Bos taurus taurus</i> )	10,402	186
BearID [30]	Brown bear ( <i>Ursus arctos</i> )	4675	132
Multi Camera Pig Tracking [246]	Domestic pig ( <i>Sus scrofa domesticus</i> )	380	33
Jaguar ID [247]	Jaguar ( <i>Panthera onca</i> )	176	16
<b>NDD20 (Above Water)</b>	Bottlenose dolphin ( <i>Tursiops truncatus</i> ) & white-beaked dolphin ( <i>Lagenorhynchus albirostris</i> )	2201	44
<b>NDD20 (Below Water)</b>	White-beaked dolphin ( <i>Lagenorhynchus albirostris</i> )	2201	82

<sup>†</sup> Beluga ID 2022: [lila.science/datasets/beluga-id-2022](https://lila.science/datasets/beluga-id-2022)

<sup>‡</sup> Leopard ID 2022: [lila.science/datasets/leopard-id-2022](https://lila.science/datasets/leopard-id-2022)

<sup>§</sup> Hyena ID 2022: [lila.science/datasets/hyena-id-2022](https://lila.science/datasets/hyena-id-2022)

datasets such as The Fishnet Open Images dataset [243], SEAMAPD21 [244], or FathomNet [182], the number of images in NDD20 is much lower. Whilst it may be tempting to solely compare NDD20 to the above datasets however, it is important to note the difference in use-case. NDD20 is not designed for use as a large scale dataset for model pre-training like those considered benchmark. In contrast to the non-benchmark ecology datasets mentioned, NDD20 contains class labels which allow for more fine-grained classification. As such, it is better to compare the quality of NDD20 with other individual animal identification datasets.

Unlike other datasets for individual animal identification, NDD20 is unusual in that it covers two species. Other ecology datasets that cover a range of species do not include individual identification labels [171, 176, 216]. This combination of multiple species and individual class labels provides novelty to NDD20. As seen in Table 3.1, whilst NDD20 is not the smallest dataset available for individual animal identification it is still on the lower end both in terms of number of images and individuals. This is due to the nature of the populations surveyed. The Coquet to St. Mary’s and Farnes Deep MCZs are small geographic areas, resulting in smaller population catalogues.

## 3.4 Summary

This chapter explores the creation of two datasets required for further work undertaken in this thesis to be performed. The first, known as the Zanzibar dataset, was developed utilising existing imagery collected by Newcastle University’s Marine MEGAfauna Lab and contains coarse-grained labels useful for the creation of an instance segmentation model. The second, known as the Northumberland Dolphin Dataset 2020 (NDD20), was developed through the use of imagery collected during fieldwork undertaken as part of this thesis. Containing

both coarse and fine-grained labels, NDD20 provides the data required for both instance segmentation and few-shot classification tasks. The usefulness of NDD20, irrespective of size limitations, is highlighted through its inclusion in a publicly hosted Kaggle competition<sup>5</sup>, acceptance to the 7<sup>th</sup> Fine-Grained Visual Categorization Workshop (FGVC7) hosted at CVPR 2020 [40], and its use in the evaluation of zero-shot segmentation models such as Segment Anything [128].

The use of the Zanzibar dataset to train and evaluate a model capable of instance segmentation of cetaceans in photo-id data, as well as the post-processing of model output images, is explored in Chapter 4, alongside the use of the NDD20 dataset to evaluate model performance on photo-id data collected of a different species of interest and over a different spatio-temporal scale.

---

<sup>5</sup>Kaggle competition: [kaggle.com/c/happy-whale-and-dolphin](https://www.kaggle.com/c/happy-whale-and-dolphin)



# Chapter 4

## Cetacean Detection Using Deep Learning

When building any large-scale project, it is important to break the task down into various subcomponents. This chapter examines one such subcomponent utilised in the development of an automatic photo-id system, the cetacean detector. This component takes images captured during photo-id surveys and locates regions of interest (RoIs) – defined as areas in which a dorsal fin breaches the waterline. This chapter will discuss the requirements a detector must meet, as well as its training and hyperparameter optimisation.

### 4.1 Requirements of a Cetacean Detector

Before a system for automatic cetacean detection can be developed, it is important to first define the problem and understand the requirements of the system. The overall aim of the detector is to be able to take large-scale images as input, fed in one at a time, and process them in order to locate RoIs. This detector will only be required to identify one object class, dolphins. These detected regions can then be passed further down the system pipeline for photo-identification.

As such, this detector can be considered a coarse-grained task, and at first glance may seem somewhat trivial. However due to both the nature of the environment in which the RoIs must be detected, and the technical requirements the system must perform under, this is actually a complex problem.

#### 4.1.1 Environmental Requirements

Firstly the area in which this system is to be deployed, open water, is susceptible to adverse weather conditions such as high winds. This in turn leads to sub-optimal conditions for



Figure 4.1 Some cetaceans, such as bottlenose dolphins, travel in pods. The developed detection system must be capable of splitting this pod into individual animals to be passed to the identifier.

detection which the system must be capable of handling, most notably high amounts of sea swell. Further to this, cetaceans are communal and travel in pods. An example of this behaviour can be seen in Figure 4.1. Thus, the system must be capable of differentiating between overlapping individuals. Even if not all of the overlapping individuals are suitable for identification downstream, the system must still be able to separate them into individual detections to prevent misclassification.

Next, the detector must be capable of differentiating between dolphin fins and waves. Again this might sound trivial, but thousands of years of evolution have resulted in fins and waves looking extremely similar to the untrained (artificial) eye. Especially from a distance and in choppy waters, fins and waves often have extremely similar shape and structure. Furthermore, the animal's bodies are also similarly coloured to their surroundings. These adaptations allow the animals to be better protected and camouflaged in their environment, but can cause issues with detection systems. This becomes apparent when thinking about how CNNs *see*. As described in Chapter 2.4, CNNs see input images as a matrix of pixel values. When training an object detection system the CNN is told which parts of this matrix are related to a class – any without a class label are considered background. If fins and areas of background contain similar pixel values, and these pixel values are clustered in similar ways, this can result in issues when training a model to detect instances of a class without misclassifying the background.



Figure 4.2 Two images of the same individual taken from different angles of approach, directions of travel, and distances from the vessel. Note how this changes the make-up of the dorsal fin but keeps the identifying notch, highlighted, visible.

Another important requirement is for the detector to be able to handle objects of varying size, shape, direction, and angle of approach. When working in an open water environment with live animals, there is no guarantee how the animal will approach the camera, and thus the detector must be generalisable enough to handle this.

Furthermore, how the animals breach the water is also extremely variable. Breachings may occur in any direction relative to the boat and the animal could itself be travelling in a different cardinality. The ideal scenario would be for a breaching to occur either directly East or West of the boat (off the port or starboard side respectively) with the animal travelling perpendicular as this provides the best chance for researchers, who often position themselves to capture from these sides to minimise photographing parts of the vessel, to capture markings – however this rarely occurs. For example, a breaching may occur off the port-side of the bow (approx North West relative to the boat), but the animal may be travelling in a South-Easterly direction. These approaches greatly change the look of the fin, although they may still contain identifiable markings. The detector should be able to detect these fins and pass them along for identification. An example of this can be seen in Figure 4.2, which also shows how distance from the vessel can change the camera's view of the dorsal.

As mentioned previously, weather conditions can also greatly affect how a dorsal fin is captured by a camera. However in photo-id surveys there are only two conditions that need to be worried about; swell and lighting. This is due to most research groups limiting travel in rough seas for safety reasons. Within Newcastle University's Marine MEGafauna Lab for example, this limit is a sea state  $< 4$  on the Beaufort Sea State scale [236]. As such, a mild amount of swell and splash can be expected which the detector should be capable of handling. Lighting conditions are not considered in the Beaufort scale, but for operational

reasons the vast majority of photo-id surveys take place during daylight hours. This can lead to large amounts of glare in images, especially on clear days. The detector should be invariant to these conditions.

### 4.1.2 Technical Requirements

On top of being able to handle a variety of environmental factors, there are also some technical requirements that the detector must meet. As outlined in Chapter 2, this thesis makes use of deep computer vision approaches. When using these types of tools, there is often a trade off that must be made between speed and accuracy. In most cases, these are inversely proportional to each other; the faster a system is required to perform, the lower an accuracy you must be willing to tolerate – Huang *et al.* discuss this in greater detail [248].

Because this trade off must be made, it is important to decide where a deep learning model will be utilised before it is deployed. As photo-id surveys are performed on small vessels such as Rigid Inflatable Boats (RIBs), space is severely limited on board. As such, it is not appropriate to add additional hardware to the vessel to perform this analysis during the survey. Furthermore, the current methodology of cetacean researchers is to perform identification once back on land, even when utilising photo-id aids. As the system proposed in this thesis is intended to fit into existing procedures rather than replace them, it is appropriate for the system to be land based rather than on the vessel. Thinking about the current procedure further, this thesis' proposed system could be, for example, left running overnight performing identifications whilst the researchers are asleep or during the day whilst they are on another survey. As such, there is no need for the system to operate in real-time to fit in with the current workflow of cetacean researchers provided the system completes its task within a reasonable time frame. Further to this, as the output of the detection model will be passed to an identification model, it is imperative that as much noise, defined as any non-animal related nuisance such as splash, waves, and other vessels, is removed as possible during the detection. In order to do this, the accuracy of the detection must be as high as possible, furthering the case for an accurate system over a fast one.

This idea of reducing as much noise as possible can be used to further narrow down the requirements of the detection system. As discussed in Section 2.4, the output of detection systems can be provided in different formats. In bounding box detection systems the detected objects are described by a set of at least two pixel coordinates denoting the top-left and bottom-right extremes of the object. These detections are often more cost-effective, both from a labelling perspective requiring less person-hours to complete, and to perform computationally. Bounding box-based detections are limited in their ability to remove background noise however, with only the background outside of the box removed.

If pixel-wise mappings are utilised, then each pixel is given a classification. This allows the system to be more discrete with its detection, allowing for the removal of as much background as possible. Both semantic (one mask for all objects of the same class) and instance (one mask for each object of a class) segmentation methods allow the detector to utilise pixel-wise mappings to remove background noise. Pixel-wise labelling is far more labour intensive and costly to produce compared to bounding box labelling. Utilising the requirements as defined in Section 4.1.1, specifically that the detector must be capable of reducing an overlapping pod to its individual component animals, the use of pixel-wise mappings at an instance level would be preferable over semantic or bounding box level detections. This requirement reveals a further trade-off the system must make. The amount of noise removed by the detector is proportional to the cost and labour needed to create data to train the system. This is discussed in more detail in Section 4.2.1.

Furthermore, any system performing cetacean detection from photo-id survey data must be capable of working with large scale images. In most image based tasks where deep learning is utilised, images fed to the network are downscaled to allow for faster training and a reduction in overall network size. Downscaling images reduces the number of pixels in the image, which by definition reduces the amount of information present as pixel values need to be pooled (one pixel needs to now display what multiple would have previously). For most detection tasks this would not be an issue, and indeed if this thesis' goal was solely cetacean detection there would be no issues with downscaling. This detector is not stand-alone however but rather the first stage of a pipeline of networks with the end goal of photo-identification. The identification task relies on potentially minute details in the fin such as notches; any downscaling of the image at the detection stage runs the risk of removing potentially identifiable information in the fin. As such, the image must only be reduced in size once it is certain that no identifying information will be lost. As this cannot be guaranteed at the stage of detection, the detector must be capable of operating on images without resizing.

## 4.2 Deciding on Approach

Based on the requirements outlined in Section 4.1, it is possible to begin deciding on how the cetacean detector is to be developed. One of the most important factors in the overall approach taken in the detector's development, and ultimately the overall automatic photo-id system, would be the use of either bounding boxes or pixel-wise mappings. As mentioned previously, the use of pixel-wise mappings would allow for a greater removal of background



Figure 4.3 Left: an example input image. Right: the corresponding manual crop utilised in bounding box suitability testing. The ROI used to produce the manual crop is highlighted red in the input image.

noise, but is extremely costly and labour intensive to produce. In contrast, bounding box labels are easier and cheaper to produce but will lead to less background noise removal.

#### 4.2.1 An Investigation into Bounding Boxes

Due to their relative cheapness and ease to produce, the use of bounding boxes would be extremely beneficial. However, if the use of bounding boxes at this stage would hinder the accuracy of individual identification downstream, then this would outweigh the cost of pixel-wise mappings.

As such, an investigation was undertaken to decide whether bounding boxes would be a viable option or if their use would hinder downstream identification. To begin, a small subset of the Zanzibar dataset, discussed in more detail in Section 3.1, was manually cropped to simulate the output of a bounding box detector, an example of which can be seen in Figure 4.3. This manually cropped data included some background but ensured the ROI, the dorsal fin, was centred and prominent representing an optimal output from a bounding box detector.

#### Feature Extraction with SURF

To begin, processing of the cropped images focussed on the use of feature extractors such as SURF [209]. Like its predecessor SIFT [215], SURF is invariant to scale, a major advantage for use with cetacean survey data where the ROI's size may change depending on when the image of the dorsal fin breaching the water is captured. If SURF was capable of producing feature descriptors of the dorsal fins with only partial background removal, this would show potential for individual identification where some background is present, possibly through the use of the feature descriptors.



Figure 4.4 Left: an example input image. Right: the corresponding manual crop with the result of SURF feature extraction overlaid. The majority of features extracted are from the surrounding background water. The RoI used to produce the manual crop is highlighted red in the input image.

First SURF was performed over the entire cropped image. This proved unfruitful however, picking out relatively few features in areas of the image which contained the animal's dorsal fin and instead focussing on the feature heavy areas present in the sea – an example of this can be seen in Figure 4.4. This result indicated that further refinement was required, potentially reducing the area SURF was allowed to explore.

Reduction of the search space available to SURF was achieved through the use of colour thresholding. As such, SURF would only be performed in areas of the image where pixel values fell within some defined range. Here, a mask was created programmatically for each image based on bounded RGB colour values found in the dorsal fins, giving an upper threshold of (14, 16, 26) and a lower threshold of (54, 51, 66) – see Section 2.4 for a description of the RGB colour space. An example result of SURF after colour thresholding can be seen in Figure 4.5, with coloured circles surrounding extracted features. As can be seen, colour thresholding helps in removing a large amount of background water from the computation. Issues arise however where areas of water are also within the threshold's bounds. Because of this, colour thresholding before SURF only reduces the amount of features extracted from the water, it does not remove them, which may result in misidentification downstream.

Further to this, it can be seen that SURF is incapable of extracting relevant prominent markings from the species in the image, Indo-Pacific bottlenose dolphins (*Tursiops aduncus*). For example, in Figure 4.3, a notch is clearly present on the dorsal fin which is a good marker for individual identification. However, when performing SURF on this dorsal as seen in Figure 4.5, note how this notch has not been detected by SURF, which has instead detected an area above the notch which contains no identifiable information.

Feature extraction methods such as SURF are also incapable of extracting other identifiable markers such as fin shape. As such, the use of feature extractors was deemed improper

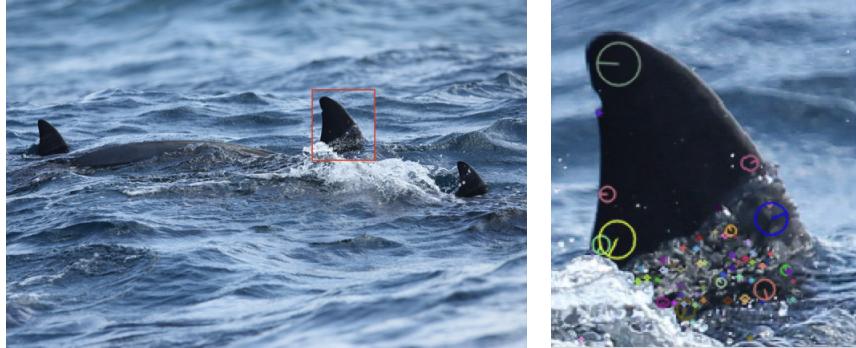


Figure 4.5 Left: an example input image. Right: the corresponding manual crop with the result of SURF feature extraction when thresholded based on RGB colour values overlaid. A large number of features have been extracted from splash surrounding the dorsal fin, alongside few identifying features. The RoI used to produce the manual crop is highlighted red in the input image.

for this use case. It is important to note here that the use of feature extractors may be appropriate for cetacean species other than this project’s data subjects of bottlenose and white-beaked dolphins. For example, the use of both SURF and SIFT has been shown to be appropriate to aid in identification of individual Risso’s dolphins [203, 210].

### Background Removal with GrabCut

Testing the suitability of SURF as described in Section 4.2.1 highlighted the need for complete background removal before identification. In order for bounding boxes to be a viable option in this scenario, a robust background removal process would need to be created. Further, the process would need to be capable of operating under unseen conditions in an unsupervised manner without pixel labelled data to train on. If the background removal process required training data to operate, this would increase the overall cost and labour required to use bounding boxes, and as such reduces the suitability of them compared to utilising pixel-wise mappings from the beginning.

GrabCut, an algorithm proposed by Microsoft Research [249], allows for the segmentation of foreground objects from the background with minimal or no human interaction. As GrabCut would be utilised in a fully automated setting, the algorithm would be required to perform background removal with no human interaction. Testing of the suitability for GrabCut was performed using the same cropped images as those used for SURF testing. Again, issues arose when performing GrabCut on the cropped image data. The algorithm struggled to understand which parts of the image were background and foreground, resulting in imperfect segmentations. This was especially an issue where the dorsal fin was present in

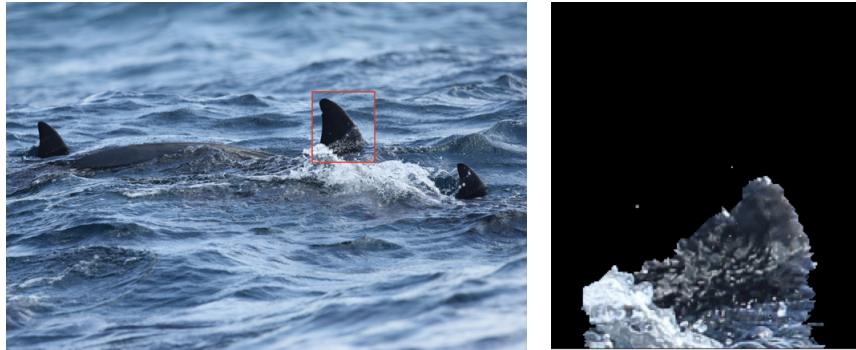


Figure 4.6 Left: an example input image. Right: the corresponding manual crop with the result of GrabCut background removal overlaid. A large portion of the dorsal fin has been discarded, having been deemed a background object due to the foreground splash. The ROI used to produce the manual crop is highlighted red in the input image.

rough water, where splash would be in front of the dorsal fin when captured by the camera. The use of GrabCut on Figure 4.3 can be seen in Figure 4.6.

As can be seen, the use of GrabCut as a background removal tool does not perform as expected on data the detector is required to operate on. Because of this, as well as the unsuitability of feature extraction as seen in Section 4.2.1, the use of bounding boxes in the cetacean detector stage was deemed improper. As such, the focus of testing moved to the use of pixel-wise mappings and instance segmentation.

## 4.2.2 Instance Segmentation Architectures

One of the major decisions to be made is which model architecture should be utilised in order to provide the required pixel-level detections. As this thesis is devoted to improving existing procedures and introducing deep learning to a novel application domain, it is far more advantageous to utilise existing model architectures rather than develop a custom one – it would not be appropriate to develop a new approach from scratch should an existing one achieve the desired outcome. The development of a custom architecture for this stage of the project would also be extremely time consuming, taking away from more novel parts of the project (notably the identification of the individual animals). Further, as this project is introducing deep learning methods to an application domain where it is not commonplace, the project needs to be able to convince researchers in the space that the system is reliable; this is more easily achieved using a pre-existing architecture where use cases already exist in the literature.

To this end there are two main model architectures that can be considered for this task; U-Net [208] and Mask R-CNN [36]. Both of these architectures work in different ways.

Vuola *et al.* provide a more detailed comparison between the two models [250], however the main focus for this thesis is their resultant output mask structure.

U-Net is based on an encoder-decoder architecture. This allows for fast and simple segmentation when working with images where only one output is required. For example taking U-Net's original use case of biomedical imaging, the model is able segment a group of cells efficiently into individual components through boundary estimation to locate the outer edges of the cells. This allows them to be segmented from each other. However this results in an output of the same dimensions as the input, that is, all segmentations are provided in a single binary mask.

In contrast, Mask R-CNN utilises a multi-stage architecture (described in more detail in Section 2.4.5) allowing the model to place each detection on its own binary output mask. This is extremely important for the outlined use case. As the detector will be used as part of a larger system for automatic photo-id, any detections made will need to be passed to the identifier as a stand-alone image. If U-Net was utilised for the detection stage, whilst initially being more efficient than Mask R-CNN, further processing of the binary output mask would still be required to split this into its individual components. In contrast, if Mask R-CNN was utilised then the processing required between the detection and identification stage would be far simpler. Again, this allows for more time to be spent working on the novel aspects of this thesis whilst keeping the pipeline as simple as possible. These outlined reasons were a major factor in deciding to focus on Mask R-CNN for this stage in the pipeline.

### 4.3 Initial Testing of Mask R-CNN

The following sections detail the creation of the Mask R-CNN based cetacean detector. Figure 4.7 highlights where in the system pipeline this model will be placed.

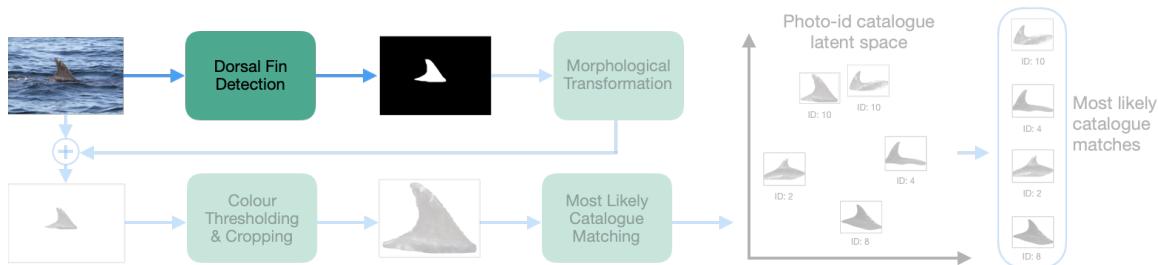


Figure 4.7 The high level pipeline overview, shown in Figure 1.1, with the Mask R-CNN component highlighted. It is this part of the pipeline that will be discussed in the following Sections.

Rather than developing a custom Mask R-CNN implementation for this project, Matterport’s repository [168] was adapted. To determine the suitability of a Mask R-CNN based model for the task of a cetacean detector, the Zanzibar dataset was divided randomly using an 80-20 split, where 80% of the images are designated for training the Mask R-CNN model, known as the training set, and the remaining 20% were held back for model evaluation, known as the test set. By evaluating on previously unseen data this affords researchers the ability to understand the generalisability of the trained model, mitigating overfitting.

### 4.3.1 Transfer Learning

Whilst the Zanzibar dataset provides experimental data similar to that which the Mask R-CNN model will be required to process, the amount of data is extremely small. Deep learning models often require thousands of images when training to produce generalisable and accurate models. As such, this dataset alone would not be enough to train the cetacean detector. One way to approach this issue would be to locate more photo-id data. However, little extra data were readily available from the Marine MEGafauna Lab, and data from other labs was deemed too costly to obtain. Cetacean catalogues are closely guarded by research labs due to the large amount of effort required to obtain them. Second, any further data collected would also need to be labelled and incorporated into the now existing dataset, which again would require significant time and effort. These issues rendered the prospect of expanding the Zanzibar dataset unachievable in the time required.

Another approach available is the concept of transfer learning. This is a technique whereby models trained to perform one task are repurposed to aid in a second, usually more specialised task. These initial models have typically been trained on large generalised datasets such as ImageNet [70] or Microsoft’s Common Objects in Context, more commonly known as MSCOCO [251]. These datasets often contain hundreds of thousands of images covering a large number of classes, which make them perfect for the task of transfer learning.

By first training a model on these large datasets, the model is able to learn the basics of image understanding, for example the concept of basic shapes and colour, allowing for the development of a generic visual understanding model. Through this, the model is effectively given a head start in its learning process as there is no need to utilise the small amount of data available in the Zanzibar dataset for low level learning; it can instead be saved for allowing the model to understand and generalise to the domain-specific task, such as cetacean detection. For a more in-depth analysis of transfer learning, see Pan *et al.* [71].

Training a neural network, or model, is extremely computationally and time expensive due to the large dataset sizes used. As such, many models suitable for transfer learning can be obtained in a pre-trained state. These pre-trained models are hosted by model zoos, which

provide frozen model weight files in a format which allows for transfer learning to take place through a process known as fine-tuning. Here, a model from the zoo is downloaded and  $N$ -number of deeper layers are unfrozen. Next, additional layers are added to the model which perform the domain-specific task. The unfrozen and additional layers are then trained on the domain-specific task, allowing for the fine-tuning of the higher-level feature extraction.

### 4.3.2 Utilising Transfer Learning to Train the Mask R-CNN

The use of transfer learning can be easily adapted for the training of the cetacean detector for use with the Zanzibar dataset. First, a backbone model architecture is chosen. For the cetacean detector, it was decided that a ResNet50 [103] backbone would be utilised. This was chosen over ResNet101 as during initial experimentation, no significant improvement in accuracy was achieved using the deeper 101 layer model although a significant increase in training time was observed.

Next, the pre-trained model weights are passed to the chosen architecture from the model zoo. These weights denote the strength of the connections between the model’s layers. In the case of a model being trained from scratch without transfer learning, the weights of each layer are randomly initialised and then manipulated through backpropagation to achieve a desired model output. In transfer learning however, the model’s starting weights are not initialised randomly. Instead, the weights of the trained network hosted on the model zoo are used as a starting point. This replicates the final state of the model trained on the larger dataset.

As previously mentioned, there are multiple different models available in the zoo, all trained on a large variety of benchmark datasets. This work makes use of MSCOCO [251] for model pre-training. This is due to the fact that MSCOCO is primarily an instance segmentation dataset, and thus one of the most appropriate to use for transfer learning to another instance segmentation task. The use of MSCOCO for pre-training on R-CNN models has in recent years been well documented in literature for a variety of tasks [252–254] including in land-based photo-id systems, with Kulits *et al.* utilising MSCOCO as a transfer learning dataset when training a modified Faster-RCNN system for African elephant re-identification [204]. As Mask R-CNN builds on Faster-RCNN, it was deemed reasonable to assume MSCOCO would also work well for pretraining a Mask R-CNN based system.

Whilst it may at first seem to make sense to pre-train on large scale natural datasets such as iNaturalist [216], it is important to note that datasets such as these often contain example images of classes in an ‘iconic’ view. If class overlap exists between the pre-training and target datasets, where objects in the latter are shown in a variety of views, this may bias the

learning process [255]. By utilising a pre-training dataset where no class overlap with the target exists, the potential for object view bias is negated.

When utilising an MSCOCO pre-trained architecture for a Mask R-CNN based task, it is important to note that certain layers must be excluded when loading in the pre-trained weights as these are only utilised in Mask R-CNN models, such as those which deal with the per-pixel masks. This is because these layers require the same number of neurons as dataset classes, similar to a fully connected layer. If the MSCOCO weights were utilised at these final layers for the task at hand there would be a mismatch between the number of classes in MSCOCO, 80, and in the Zanzibar dataset, 1 (plus background).

Once the backbone architecture has been loaded with pre-trained weights, the total number of layers to fine-tune must be decided. This can be considered similar to a hyperparameter, as it must be chosen at run time by the user. Whilst any number of the layers can be chosen for fine-tuning, only the model heads are selected. These are layers required for the Region Proposal Network, the pixel classification, and masking layers of the model. Whether model weights are randomly initialised or loaded from a pre-trained model is determined during hyperparameter tuning.

### 4.3.3 Data Augmentation

As well as transfer learning, the use of data augmentation was also explored to help mitigate the issue of dataset size. This technique allows for datasets to be artificially expanded by performing random perturbations to each data point which are then automatically class labelled the same as the original input. Figure 4.8 shows examples of data augmentations found in the literature.

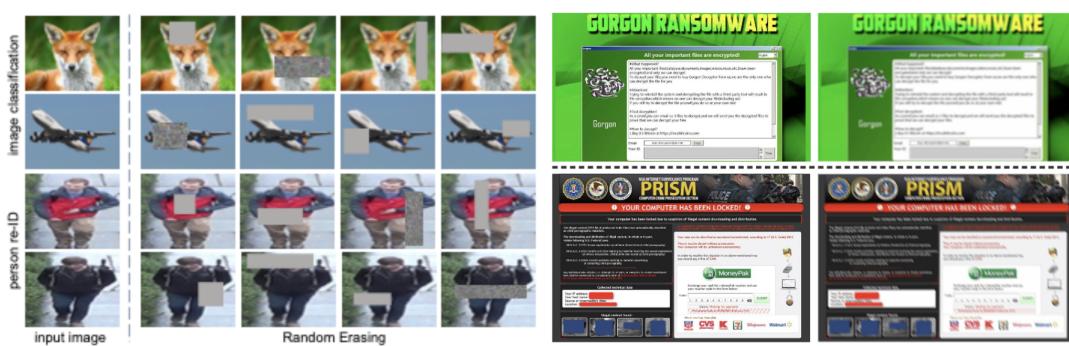


Figure 4.8 Examples of data augmentations found in the literature. Left: randomly erasing parts of images, from Zhong *et al.* [256]. Right: augmentations to simulate screenshot capture. Defocus blur (Top), motion blur (Bottom) from Atapour-Abarghouei *et al.* [257].

As the Zanzibar data contains relatively few images, it is a prime candidate for data augmentation. This can be performed in one of two ways; in either an offline or an online manner. In offline data augmentation, the entire train split is augmented before the images are passed to the model, occurring as a pre-processing step. The major issue with offline augmentations however is that, because the data is perturbed and then passed to the model, offline augmentation produces a fixed number of augmented images.

This can be solved with online augmentation. Occurring in real-time as the model trains, this allows for a potentially unlimited number of ‘new’ images, as each input is randomly perturbed before being used for training. Once training on the batch has been completed, the augmented images are discarded and new perturbations performed. As such online augmentation is, if possible, greatly preferred and allows for a much higher chance of model generalisation.

Whilst the Zanzibar dataset is small compared to many others used for deep learning, it is large enough to allow for online augmentation. In order to begin testing the effect of data augmentation on the Mask R-CNN training process, two different augmentation strategies were created which contained unique workflows.

The first strategy, *aug1*, selected between zero and three of the following perturbations: (1) *horizontal flip*: flip the image horizontally with a probability of 0.5, (2) *vertical flip*: flip the image vertically with a probability of 0.5, (3) *rotation*: rotate the image either 90, 180, or 270 degrees each with equal probability of occurring, (4) *scaling*: scale the image between 80% and 120% on both axis independent of each other, (5) *brightness*: multiply all pixels in the image with a random value between 0.8 and 1.5, (6) *Gaussian blur*: blur the image with a Gaussian kernel with radius randomly assigned between 0 and 5.

The second strategy, *aug2*, was more complex, performing the following perturbations in a sequentially random order on 67% of the images: (1) *horizontal flip*: flip the image horizontally with a probability of 0.5, (2) *cropping*: crop each side of the image randomly between 0% and 10% of the total side length, (3) *Gaussian blur*: blur the image with a Gaussian kernel with radius randomly assigned between 0 and 2.5, with a probability of blurring of 0.5, (4) *contrast*: strengthen or weaken the contrast of the image by a random factor between 0.75 and 1.5, (5) *additive Gaussian noise*: sample the noise per channel – adding noise to the colour of the pixels, (6) *brightness*: multiply all pixels in the image with a random value between 0.8 and 1.2, (7) *scaling*: scale the image between 80% and 120% on both axes independent of each other, (8) *rotation*: rotate the image randomly between -180 and 180 degrees.

The use of two augmenters allowed for evaluation on whether a simple or more complex augmentation strategy would be appropriate for this use case. By using multiple augmenters

we can treat them as a hyperparameter of model training, allowing the augmenter chosen to be added to the search space.

## 4.4 Mask R-CNN Model Selection

When training a Mask R-CNN model there are a large range of hyperparameters, or user defined values, which must be set before training can occur. These hyperparameters each have influence on the final model’s performance, and can be broken down into two subgroups; detection hyperparameters influence the output of the model, and training hyperparameters which influence the training of the model. Thankfully most deep learning frameworks provide default values for most, if not all hyperparameters. These default values are known to work well regardless of dataset or task, and so many have been used when training the Mask R-CNN. Some hyperparameters however can have a large effect on the final model and so an exploration of the optimal value for these has been undertaken with the goal of producing the optimal overall model for the task of cetacean detection, both on the Zanzibar dataset and on other similar datasets.

### 4.4.1 Detection Hyperparameters

With regards to the detection hyperparameters, only the minimum confidence of the model was changed from the default of 0.7 to 0.9. This was changed as during initial trials it was found that models trained on the Zanzibar data would often produce a high number of false positives (for example detecting a wave as a fin) or create duplicate detections (one fin detected twice). By increasing the minimum confidence of the model to 0.9, the threshold at which the model returns a detection is increased to 90% – or in other words, for every detection the model must be 90% sure that the detection is actually a cetacean before notifying the user. This has the effect of reducing the number of erroneous detections passed downstream, such as in Figure 4.9 where increasing the minimum confidence from 0.7 (Left) to 0.9 (Right) has removed a low confidence, erroneous, detection.

### 4.4.2 Training Hyperparameters

The vast majority of hyperparameters influence the training process. Selection of the optimal hyperparameters is however an extremely computationally and time expensive task, as the optimal values of the hyperparameters are not known before training begins. Indeed, even after training has finished and a model which produces satisfactory results has been found

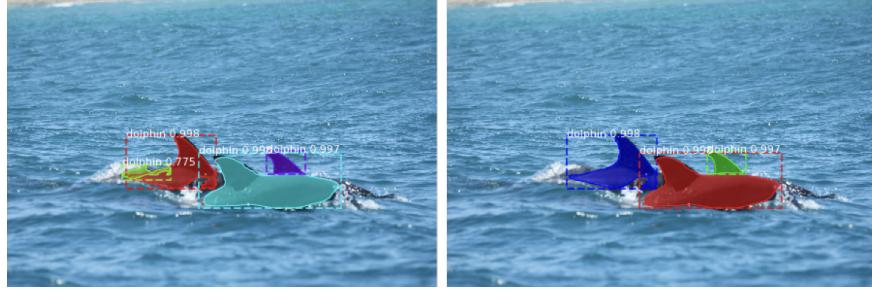


Figure 4.9 An example image showing the effect of changing the Mask R-CNN’s minimum detection confidence. Left: a threshold of 0.7. Right: a threshold of 0.9. The number of detections has reduced from four to three, with the erroneous left-most detection (highlighted green in the Left Figure) removed.

there is no guarantee that the hyperparameters of this model are the best, just that they are the best found so far.

As such, in order to determine the best hyperparameters for a given model and task, the search space of all possible hyperparameters must be searched. This is infeasible due to time and resource constraints however, therefore a technique known as grid searching was performed. During a grid search, each hyperparameter is defined as a range of possible values. A model is then trained using the data and each combination of defined hyperparameter values. Once each model has trained, they are then evaluated using the test data to determine the best hyperparameters.

### Learning Rate Scheduling and Optimisers

One of the most important hyperparameters to tune is the learning rate, which dictates how much the weights of the model should change in response to the estimated error calculated during backpropagation. If the learning rate is too large this will lead to an unstable training process whereby gradient descent can never reach the minimum value but rather bounce either side of it. If the learning rate is set too small the training process will take an extremely long time to converge.

In order to help the model reach its optimal minima in a reasonable time, the learning rate can be adjusted using a scheduler. These allow for the learning rate to be modified when some criteria is met, such as after a set number of epochs, allowing for larger weight changes initially for fast training before reducing the descent steps as time goes on, decreasing the chance of gradient descent jumping over the minima.

As well as learning rate schedulers, adaptive rate optimisers can also be used. These optimisers provide an alternative to SGD (an overview of which can be found in Section 2.3.1)

and are capable of adapting to the dataset it is given and the current training process, changing the learning rate without a defined schedule. This often allows for a more optimised and efficient training process when compared to using SGD, as discussed in Section 2.3.1. During hyperparameter tuning of the Mask R-CNN, two optimisers were chosen for evaluation.

The first, SGD with restarts (SGDR) [74] allows for decreases in the learning rate through a process known as cosine annealing, whereby the decrease follows a cosine waveform. This results in a high starting learning rate allowing for a fast approach to a local minima before reducing the rate as the number of epochs increases to prevent a jump over the minima, similarly to how a scheduler works. However it may not be the case that this local minima is the global minima, the lowest possible point in the space. Due to cosine annealing it would not be possible to leave the local minima, the learning rate needs to be increased again to allow for this. As such the learning rate is *restarted*, or increased back to its maximum, to allow for the training process to jump out of the local minima; if it is indeed the case that this local minima is also the global minima then the training process will return to the point it was at before the restart, however if the local minima was not the global minima, the restart will allow for the training process to leave the sub-optimal minima it previously found.

The second learning rate optimisation explored during hyperparameter tuning is Adam, or adaptive moment estimation. This optimiser is extremely popular in the world of deep learning [258], capable of achieving impressive results in relatively short training times. This is possible through the use of one learning rate for each model weight, in contrast to the singular learning rate for the whole model as seen in SGD or SGDR. Adam also utilises parts of other optimisers such as AdaGrad [259] and RMSProp [79] to allow the optimiser to work well with both sparse and noisy data. For a complete breakdown of the inner workings of Adam, see Kingma *et al.* [80].

## Weight Decay

The goal of neural network development is to utilise the training data in such a way that the resulting generated model performs well on unseen data. In order for this to be achieved the model must be generalisable, having learnt enough from the training data to perform well at the given task but not having learnt so well that it is unable to perform adequately on unseen data. If a model fails to generalise, it is said to have overfitted the data. For example, a model may be developed to detect cats in images, but is only presented with examples containing white cats. In this situation, the model may operate perfectly over the training and test data, however may fail when deployed to detect black cats. The model has learnt the training data too well and believes cats can only be white – the model has overfitted.

There are many different techniques to reduce overfitting in neural networks, one of the easiest is to simply collect more training data. As previously mentioned, due to how closely guarded cetacean photo-id catalogue data is and how expensive it is to collect, this was not possible. As such, the use of weight decay was explored during hyperparameter tuning. Weight decay is a regularisation technique which allows the model training to be penalised in proportion to the size of its weights. This incentivises the training process to keep the weights small, which has been shown to improve generalisation to unseen data [260]. As the Zanzibar dataset is comparatively small compared to the usual size of datasets for this task, allowing the model to generalise well using small amounts of data is extremely important.

### Region Proposal Network Anchor Scales

As discussed in Section 2.4.2, Region Proposal Networks (RPNs) can be utilised in object detection due to their ability to determine potential regions of interest (RoIs) in the image, known as anchors. These anchors are then classified as either background or of a learnable class, such as `dolphin`. To allow the RPN to be object-size invariant, anchor scales are utilised. These scales, provided as a list of values which correspond to the square anchor size in pixels, determine what sizes the RoIs proposed by the RPN should be. For example, if an anchor scale of [32] is passed to the RPN, this would mean that all RoIs proposed by the RPN would be of size 32x32 pixels. The anchor scale provided to the RPN can be considered a hyperparameter as the best scales for the RPN to allow for the detection of objects regardless of their size must be determined.

#### 4.4.3 Hyperparameter Tuning via Grid Search

Although only a few hyperparameters have been chosen to tune, the size of the possible search space to evaluate is still extremely large. As mentioned previously, it is not feasible both from a time and resource perspective to evaluate the entire space and find the truly optimal value for each hyperparameter. Instead the search space is discretised using a grid search, for each hyperparameter a subset of possible values is selected. Each combination of hyperparameter values is then evaluated to determine which subset produces a satisfactory model.

The list of models generated by the grid search, their hyperparameter combinations, and their mAP@IOU[0.5, 0.75, 0.85] scores can be seen in Table 4.1. This hyperparameter tuning run required a significant amount of time and resources, running over three cloud instance virtual machines (VMs), each with two Tesla K80 GPUs, taking approximately one week to produce a total of 50 models. Model runs were split between the VMs based on

augmentation strategy with one VM running only *aug1*, the other *aug2*, and the final with no augmentation strategy. It should be noted here that this computational and time expense would most likely be reduced should the images used to train the Mask R-CNN not be so large, although the reasons for this decision are discussed in Section 4.1.2.

Model Name	Weight Decay	RPN Anchor Scales	Optimiser	Augmentation Strategy	Pre-trained on MSCOCO?	mAP@IOU[ $x$ ]		
						0.50	0.75	0.85
20190829T1458	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	Adam	aug1	✓	0.774	0.555	0.265
20190829T2020	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	Adam	aug2	✓	0.739	0.513	0.180
20190830T0145	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	Adam	None	✓	0.843	0.586	0.245
20190830T0714	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	Adam	aug1	✗	0.793	0.492	0.174
20190830T1443	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	Adam	aug2	✗	0.732	0.531	0.199
20190830T2019	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	Adam	None	✗	0.858	0.609	0.278
<b>20190902T0946</b>	<b><math>1 \times 10^{-2}</math></b>	<b>(16, 32, 64, 128, 256)</b>	<b>SGDR</b>	<b>aug1</b>	<b>✓</b>	<b>0.914</b>	<b>0.793</b>	<b>0.497</b>
20190904T2004	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	SGDR	None	✓	0.896	0.665	0.235
20190905T1813	$1 \times 10^{-3}$	(32, 64, 128, 256, 512)	SGDR	aug1	✓	0.937	0.775	0.452
20190905T1826	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	SGDR	aug2	✓	0.915	0.762	0.418
20190905T2202	$1 \times 10^{-3}$	(32, 64, 128, 256, 512)	Adam	None	✓	0.817	0.594	0.281
20190905T2336	$1 \times 10^{-3}$	(32, 64, 128, 256, 512)	Adam	aug1	✓	0.822	0.640	0.265
20190906T0332	$1 \times 10^{-3}$	(32, 64, 128, 256, 512)	SGDR	None	✓	0.902	0.738	0.417
20190906T0851	$1 \times 10^{-2}$	(32, 64, 128, 256, 512)	Adam	None	✓	0.856	0.640	0.277
20190907T0932	$1 \times 10^{-3}$	(16, 32, 64, 128, 256)	Adam	aug1	✓	0.834	0.567	0.239
20190907T0933	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	Adam	aug2	✗	0.844	0.592	0.243
20190907T0934	$1 \times 10^{-2}$	(32, 64, 128, 256, 512)	SGDR	None	✗	0.921	0.796	0.457
20190907T1451	$1 \times 10^{-3}$	(16, 32, 64, 128, 256)	Adam	None	✓	0.804	0.560	0.197
20190907T1500	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	Adam	aug1	✗	0.837	0.649	0.289
20190907T1545	$1 \times 10^{-2}$	(32, 64, 128, 256, 512)	Adam	aug2	✗	0.850	0.612	0.229
20190907T2026	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	SGDR	None	✓	0.919	0.782	0.412
20190907T2126	$1 \times 10^{-3}$	(16, 32, 64, 128, 256)	Adam	aug1	✗	0.827	0.544	0.257
20190907T2215	$1 \times 10^{-3}$	(16, 32, 64, 128, 256)	SGDR	aug2	✓	0.928	0.789	0.452
20190908T0202	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	Adam	None	✓	0.848	0.619	0.281
20190908T0352	$1 \times 10^{-2}$	(32, 64, 128, 256)	Adam	aug2	✓	0.836	0.566	0.259
20190908T0417	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	Adam	aug1	✓	0.827	0.619	0.287
20190908T0957	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	Adam	None	✗	0.884	0.681	0.324
20190908T1102	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	Adam	aug2	✗	0.790	0.581	0.256
20190908T1204	$1 \times 10^{-3}$	(16, 32, 64, 128, 256)	SGDR	aug1	✓	0.901	0.733	0.453
20190908T1939	$1 \times 10^{-3}$	(16, 32, 64, 128, 256)	Adam	aug2	✗	0.811	0.579	0.203
20190908T2043	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	SGDR	aug1	✓	0.929	0.804	0.482
20190908T2139	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	Adam	None	✗	0.910	0.686	0.349
20190909T0723	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	Adam	aug1	✗	0.798	0.644	0.298
20190911T1922	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	Adam	aug2	✗	0.765	0.508	0.174
20190912T0045	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	Adam	aug2	✗	0.780	0.539	0.214
20190912T0608	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	SGDR	aug1	✓	0.910	0.771	0.463
20191101T1633	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	SGDR	aug2	✓	0.909	0.780	0.419
20191101T2104	$1 \times 10^{-3}$	(8, 16, 32, 64, 128)	SGDR	aug2	✓	0.901	0.750	0.340
20191102T0140	$1 \times 10^{-2}$	(32, 64, 128, 256, 512)	SGDR	aug1	✓	0.916	0.813	0.442
20191102T0615	$1 \times 10^{-2}$	(8, 16, 32, 64, 128)	SGDR	aug2	✓	0.902	0.745	0.410
20191102T1051	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	SGDR	None	✗	0.914	0.812	0.461
20191102T1528	$1 \times 10^{-3}$	(32, 64, 128, 256, 512)	SGDR	aug2	✓	0.919	0.778	0.425
20191102T2006	$1 \times 10^{-4}$	(8, 16, 32, 64, 128)	SGDR	aug2	✓	0.919	0.747	0.391
20191103T0044	$1 \times 10^{-4}$	(8, 16, 32, 64, 128)	SGDR	aug1	✓	0.901	0.736	0.348
20191103T0520	$1 \times 10^{-3}$	(8, 16, 32, 64, 128)	SGDR	None	✓	0.913	0.778	0.443
20191103T0959	$1 \times 10^{-2}$	(32, 64, 128, 256, 512)	SGDR	aug2	✓	0.929	0.766	0.437
20191103T1441	$1 \times 10^{-4}$	(8, 16, 32, 64, 128)	SGDR	None	✓	0.882	0.739	0.401
20191103T1921	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	SGDR	aug2	✓	0.926	0.768	0.360
20191104T0011	$1 \times 10^{-3}$	(8, 16, 32, 64, 128)	SGDR	aug1	✓	0.912	0.788	0.391
20191104T0450	$1 \times 10^{-2}$	(8, 16, 32, 64, 128)	SGDR	aug1	✓	0.915	0.782	0.394

Table 4.1 Hyperparameter values and mAP@IOU[0.5, 0.75, 0.85] scores used for each Mask R-CNN grid search model trained on the Zanzibar data. All models were trained for 50 epochs with an initial learning rate of  $1 \times 10^{-3}$ . The model chosen for use is highlighted in bold. The mAP@IOU[0.5:0.95] scores for all models can be seen in Appendix A.

#### 4.4.4 Model Selection Based on Grid Search

Once a grid search has been performed, the results can then be evaluated to determine if a suitable model had been found using the test set. All models trained were evaluated using MSCOCO’s Mean Average Precision metric<sup>1</sup>, a commonly used metric for segmentation tasks. This metric, written as mAP@IOU[0.5:0.95], calculates precision-recall graphs for each dataset class at incremental IOU levels, from 0.5 to 0.95 in 0.05 steps. Once each class’ precision-recall graph for a given IOU threshold has been calculated, the mean of these values is derived giving an overall mean average precision score for all classes at a given IOU threshold; these thresholds are explained in more detail in Section 2.4.3. Appendix A provides the full mAP@IOU[0.5:0.95] scores for all models trained in the grid search.

By evaluating over multiple thresholds the models can then be compared and their performance more easily understood and ranked, as well as allow for the determination of an acceptable loss in IOU overlap. For example if all models were evaluated using mAP@IOU[0.5] only, it may be the case that all models achieve a similar high score, making it difficult to determine which model will be best for the task. However if too high a threshold is used, for example mAPIOU[0.95], it is unlikely that any model will achieve a high score as this constant near pixel perfect detection.

At mAP@IOU[0.5] there is a large gap in model performance with model 20190830T1443 having the lowest mAP@IOU[0.5] of 0.73 and model 20190905T1813 having the highest at 0.94. This shows that the combination of hyperparameter values provided to the model before training have a significant effect on the model’s overall performance, although even the lowest score here is still high.

At mAP@IOU[0.75], whereby detections would overlap with 75% of pixels in the ground truth mask, the minimum model performance has dropped significantly with model 20190830T071 achieving a score of 0.49. The highest score at this threshold is model 20191102T0140 with a score of 0.81; this model achieved an mAP@IOU[0.5] score of 0.92, only two percentage points behind the best model at that threshold. This again shows the need for hyperparameter tuning when selecting models, as they are shown here to have a significant effect on how well the models perform at higher thresholds.

This effect is even greater when comparing map@IOU[0.85] scores, with the worst performing model, 20190830T0714, achieving a score of just 0.17 whilst the best model, 20190902T0946, achieves a score of 0.50, a difference of 33%. Model performance drops significantly at the highest threshold with four models achieving an mAP@IOU[0.95] score of 0.016, with most models achieving a score of 0.0. This is to be expected however as it

---

<sup>1</sup>COCO mAP Definition: [cocodataset.org/#detection-eval](http://cocodataset.org/#detection-eval)

would be highly unlikely that any model, regardless of hyperparameters, would be able to perform near perfect pixel level detections on the test set data.

To determine the most appropriate model trained, filtering based on the highest mAP@IOU[0.5, 0.75, 0.85] scores was performed. The thresholds 0.5 and 0.75 were chosen to remain consistent with other segmentation literature [147, 149?], with the 0.85 threshold chosen as some models trained still achieve impressive results here, allowing for more in-depth model filtering. Further, the pixel-wise detections of cetaceans is required to filter as much background noise as possible and so finding high performing models at top thresholds is important.

When filtering, the top five performing models at each threshold were extracted. If a model achieved top five ranking at multiple thresholds it was only included once, resulting in a list of the ten best performing models. The hyperparameters and mAP@IOU[0.5, 0.75, 0.85] scores for these best performing models can be seen in Table 4.2.

Model Name	Weight Decay	RPN Anchor Scales	Optimiser	Augmentation Strategy	Pre-trained on MSCOCO?	mAP@IOU[x]		
						0.50	0.75	0.85
<b>20190902T0946</b>	$1 \times 10^{-2}$	(16, 32, 64, 128, 256)	SGDR	aug1	✓	0.914	0.793	0.497
20190905T1813	$1 \times 10^{-3}$	(32, 64, 128, 256, 512)	SGDR	aug1	✓	0.937	0.775	0.452
20190907T0934	$1 \times 10^{-2}$	(32, 64, 128, 256, 512)	SGDR	None	✓	0.921	0.796	0.452
20190907T2215	$1 \times 10^{-3}$	(16, 32, 64, 128, 256)	SGDR	aug2	✓	0.928	0.796	0.457
20190908T2043	$1 \times 10^{-4}$	(32, 64, 128, 256, 512)	SGDR	aug1	✓	0.929	0.804	0.482
20190912T0608	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	SGDR	aug1	✓	0.910	0.771	0.463
20191102T0140	$1 \times 10^{-2}$	(32, 64, 128, 256, 512)	SGDR	aug1	✓	0.916	0.813	0.442
20191102T1051	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	SGDR	None	✓	0.914	0.812	0.461
20191103T0959	$1 \times 10^{-2}$	(32, 64, 128, 256, 512)	SGDR	aug2	✓	0.929	0.766	0.437
20191103T1921	$1 \times 10^{-4}$	(16, 32, 64, 128, 256)	SGDR	aug2	✓	0.926	0.768	0.360

Table 4.2 Hyperparameters of the best performing Mask R-CNN models on the Zanzibar dataset. Subset of Table 4.1. All models were trained for 50 epochs with an initial learning rate of  $1 \times 10^{-3}$ . The model chosen for use is highlighted in bold. The mAP@IOU[0.5:0.95] scores for all models can be seen in Appendix A.

When deciding on which model hyperparameters are best for the task of cetacean segmentation, it is important to find a model with a high mAP@IOU[0.85] score. As the model will be used to perform segmentation before fine-grained classification, it is important the model is capable of removing as much background from the input image as possible. Any background included in the segmentation may adversely effect the photo-id process. Using this as criteria, model 20190902T0946 was selected as the best performing model. The model achieves an mAP@IOU[0.85] score of 0.5, an excellent result given the difficulty of the segmentation task. The model also performs well at the other evaluation thresholds, achieving mAP@IOU[0.5, 0.75] scores of 0.91 and 0.79 respectively. These scores verify that the model is capable of segmenting cetaceans from background with as little noise being included in the segmentation mask as possible.

An interesting point to note here is that 20190902T0946 did not achieve the highest mAP@IOU[0.5, 0.75] scores. As previously mentioned, these thresholds are often the ones

included in segmentation literature to evaluate model performance. If just these thresholds were used for model selection, 20190902T0946 would not have been chosen. This highlights the need to select models based on metrics which make sense for the task at hand. As the model is required to remove as much background noise as possible, using a high threshold for evaluation makes sense. Thresholds higher than 0.85 were not utilised due to the low performance of all models at this threshold, although 20190902T0946 also achieves one of the best mAP@IOU[0.9] scores of 0.150. Only one model, 20191102T0615 achieves a better score at this threshold, 0.158, however this model achieves lower performance at the chosen evaluation thresholds of 0.5, 0.75, and 0.85.

#### 4.4.5 An Evaluation of Optimal Model Hyperparameters

As discussed in Section 4.4.4 a filtering of the trained Mask R-CNN models was performed to determine the best model hyperparameters for the task of cetacean instance segmentation, with model 20190902T0946 ultimately being selected for future use. Model hyperparameters and mAP@IOU[0.5, 0.75, 0.85] scores can be seen in Table 4.2.

The hyperparameters of the best performing models provide an interesting insight into the training process. All ten of the models were trained using SGDR. This is interesting, as the current trend in deep learning network training is to utilise Adam [258]. Furthermore each model trained utilised transfer learning, with each model's parameters being initialised from a trained MSCOCO model provided by the model zoo. This highlights the need to utilise pre-trained models, especially in cases where relatively small amounts of data are available to train a model from scratch.

Half of the best models utilise the *aug1* data augmentation strategy, defined in Section 4.3.3. The smallest RPN Anchor Scale, (8, 16, 32, 64, 128), has not been utilised by any of the best models, and the value of the weight decay hyperparameter is split between the three possible values. These splits highlight the need for a robust and in-depth hyperparameter search, as with the majority of hyperparameters searched no clear trend can be identified.

#### 4.4.6 Limitations of the Model

As with any network trained on real world data, the best performing model, 20190902T0946, is not perfect. As highlighted by the mAP scores in Table 4.1, the model still fails to correctly detect in some instances. This section examines under what conditions 20190902T0946 fails for the purposes of model transparency.

Figure 4.10 provides a good example of environmental conditions which can cause the model to fail to correctly detect an individual. Due to wave movement covering a part of the animal's body, the model has split this individual into two separate detections.

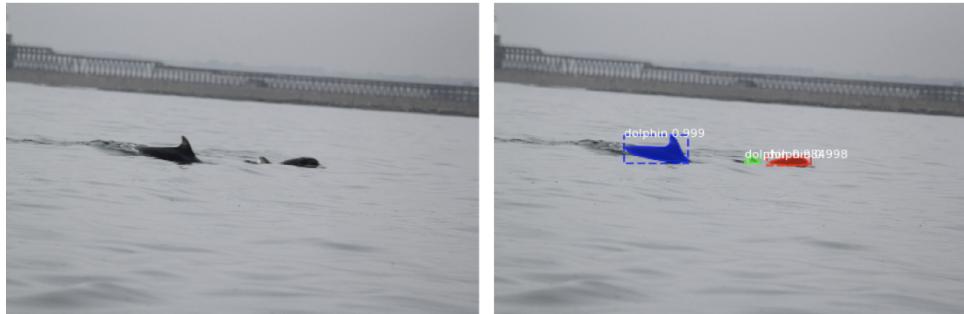


Figure 4.10 Left: the input image passed to the cetacean detector. Right: the detection masks produced by the model overlaid onto the image. Note how the cetacean on the right has been split over two masks due to occlusion from a small wave.

The model also struggles in cases where the image contains an animal under the waterline but, due to the clarity of the water, that animal is partly visible in the image. In this case the model often detects the individual under the waterline. Due to these individuals not being useful for identification purposes however, they were not labelled in the dataset and thus are deemed to be misclassifications when evaluating the model. Figure 4.11 shows an example of this issue occurring.



Figure 4.11 Left: the image passed to the cetacean detector. Right: the detection masks produced by the model overlaid onto the image. Note how the green detection is of an individual under the waterline and only partly visible, and thus useless for identification purposes.

The Zanzibar dataset contains a large number of images which contain other vessels as well as dolphins. This is due to the large marine eco-tourism industry within Zanzibar [10, 261]. Whilst this issue may not be present in other survey areas where this model may be deployed, it still denotes an example of the model failing. In this case often parts of the boat,

or a combination of the boat and the humans on the boat, may cause the model to incorrectly identify a grouping of pixels as a dolphin. An example of this can be seen in Figure 4.12.



Figure 4.12 Left: the image passed to the cetacean detector. Right: the detection masks produced by the model overlaid onto the image. Note how the red detection is a misclassification. The model believes a section of the boat’s hull and the leg of the human to be a dolphin.

All of these mis-detections have an impact of the overall evaluation score of the model. These mis-detections will then be passed further down the system pipeline to be classified as individuals. To reduce the chances of this happening, a robust post-processing technique was developed, as discussed in Chapter 5.

## 4.5 Model Evaluation Using NDD20

To examine the dorsal fin detector’s robustness to changes in geography, time, and species, evaluation of the model was then undertaken using the NDD20 dataset.

### 4.5.1 Evaluating the Effect of Geography, Time, and Species Change

As discussed previously, a Mask R-CNN model capable of above water cetacean detection was trained on indo-pacific bottlenose dolphin data collected in Zanzibar, Tanzania. One important requirement of the detector created is that it must be robust enough to output detection masks with high mean average precision (mAP) when operating on data from a different geographic or temporal area without re-training. The creation of NDD20 provides a valuable opportunity to test this requirement. Not only was the data collected in a different location and time, but both species of data subject (bottlenose and white-beaked dolphin) are not present in the Zanzibar dataset.

To test this requirement the best performing model found on the Zanzibar data in Section 4.4.4, 20190902T0946, was utilised to generate instance segmentation mask predictions for

<b>Dataset</b>	<b>mAP@IOU[<math>x</math>]</b>									
	<b>0.50</b>	<b>0.55</b>	<b>0.60</b>	<b>0.65</b>	<b>0.70</b>	<b>0.75</b>	<b>0.80</b>	<b>0.85</b>	<b>0.90</b>	<b>0.95</b>
<b>Zanzibar</b> <b>(Test Set)</b>	0.91	0.91	0.89	0.86	0.85	0.79	0.69	0.50	0.15	0.00
<b>NDD20</b> <b>(Above Water)</b>	0.96	0.95	0.93	0.91	0.88	0.83	0.71	0.51	0.16	0.00

Table 4.3 mAP@IOU[0.5:0.95] scores for model 20190902T0946, the best performing Mask R-CNN dorsal fin detector. The model has been trained using the Zanzibar training set, and evaluated using both the Zanzibar test set and the full NDD20 above water set. Hyperparameters for this model can be seen in Table 4.1.

the above water set of NDD20. These model outputs were then evaluated against the labelled ground truth data to produce an mAP score over multiple IOU thresholds.

Model 20190902T0946 still achieves a high mAP at multiple IOU thresholds without the need for re-training or fine-tuning on NDD20, as seen in Table 4.3. Utilising the same evaluation thresholds as during model selection in Section 4.4.4, the model achieves mAP@IOU[0.5, 0.75] = [0.96, 0.83] on NDD20. This is in comparison to mAP@IOU[0.5, 0.75] = [0.91, 0.79] on the Zanzibar data.

Interestingly, the model achieves a higher mAP at these IOU thresholds on NDD20. This is hypothesised to be due to the lack of other large objects in NDD20 in comparison to the Zanzibar data. For example, some images in the Zanzibar dataset contain other vessels as well as humans as a result of the data being captured in an area with high levels of eco-tourism [262]. This is not the case for data contained within NDD20. Whilst eco-tourism is present in and around the Coquet to St. Mary's MCZ, the levels are significantly lower than in Zanzibar. The evaluated model has been seen to struggle when presented with images containing tourist activity, shown in Figure 4.12. As NDD20 lacks this, the model's false positive rate may be reduced. Regardless, this evaluation presents evidence that model 20190902T0946 is robust enough to deal with data from a different geographic area, time, and cetacean species without the need for re-training or fine-tuning. This suggests the model could be deployed to aid in the speed up of future photo-id fieldwork seasons undertaken by marine ecologists.

#### 4.5.2 Below Water Detection Baseline

Though the below water set of NDD20 is not utilised within this thesis further, work was also undertaken to produce a baseline instance segmentation score for this data. Here, a second Mask R-CNN model was trained using the optimal hyperparameters located in Section 4.4 on the below water data divided randomly into an 80:20 train-test split, with the out of

focus flag ignored. Results for this model can be seen in Table 4.4, and indicate that a Mask R-CNN model is capable of accurate instance segmentation on this data – in spite of the large variation in mask shape and a lack of clarity between the background and the dolphin due to water condition.

<b>Dataset</b>	<b>mAP@IOU[x]</b>									
	<b>0.50</b>	<b>0.55</b>	<b>0.60</b>	<b>0.65</b>	<b>0.70</b>	<b>0.75</b>	<b>0.80</b>	<b>0.85</b>	<b>0.90</b>	<b>0.95</b>
<b>NDD20 (Below Water)</b>	0.986	0.982	0.979	0.974	0.964	0.938	0.871	0.654	0.202	0.00

Table 4.4 mAP@IOU[0.5:0.95] scores for the best performing Mask R-CNN model trained using the below water set of NDD20.

## 4.6 Summary

This chapter discusses the thesis’ need for a model capable of cetacean detection, both from a technical and environmental perspective. The key reasons behind the use of instance segmentation masks rather than the relatively less computational and time expensive bounding boxes is explained, with evidence showing how the difficulty of the task influenced this move.

Once the system requirements and underlying model architecture have been identified, the chapter then examines the use of model hyperparameter optimisation to train a model capable of cetacean detection via instance segmentation masks contained in the Zanzibar dataset (outlined in Section 3.1). Model pre-training is also explored, and highlights the benefits of this approach even when using a pretraining dataset whose domain and distribution are vastly different to the final model’s goal. The chapter then examines the use of post-processing techniques to improve the output of the Mask R-CNN’s detections for use in downstream tasks.

It is of great importance that the trained Mask R-CNN model is capable of detecting cetaceans in photo-id data which has been gathered in different geospatial and temporal areas. It is also important that the detector is capable of similarly high levels of accuracy without re-training on data from that geographic area. As such, the detector’s use on the NDD20 dataset, collected in a different geospatial and temporal area (outlined in Section 3.2), is explored. The final result of this chapter is a Mask R-CNN model capable of high mAP even at large IoU thresholds.

Before dorsal fin detections produced by the Mask R-CNN model developed in this chapter can be passed to a most likely catalogue matching system, some post-processing of the output must be performed. This will allow for the retention of any individually

identifying markings potentially missed by the model's binary output mask, as well as allow for a reduction in the size of the input passed for catalogue matching, reducing downstream computational expense. This post-processing methodology is outlined in Chapter 5.



# Chapter 5

## Mask Post-Processing Techniques

As shown in Chapter 4, at this stage the Mask R-CNN model is capable of detecting cetaceans at an individual pixel level. Before these detections can be passed to the identification module, some post-processing of the output must be performed to allow for both a reduction in the computational expense of operating on the detector's output as well as ensuring that no potentially important information which will assist in an identification is lost. As such, the following chapter details the mask post-processing techniques implemented as required by the system pipeline, as shown in Figure 5.1.

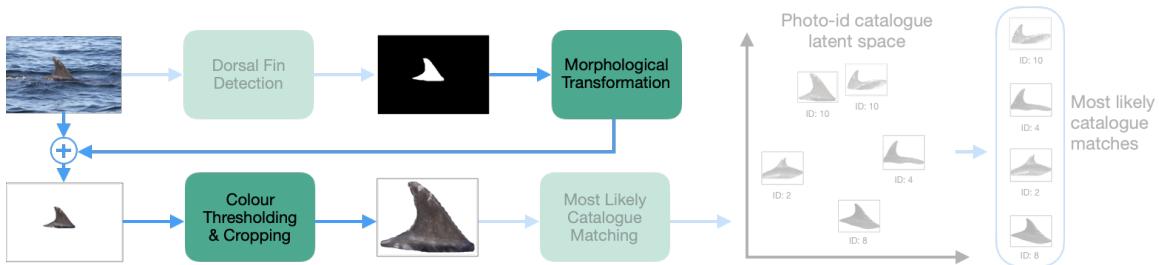


Figure 5.1 The high level pipeline overview, shown in Figure 1.1, with the post-processing techniques highlighted. It is this part of the pipeline that will be discussed in the following section.

### 5.1 Handling Multiple Detections

When an image is run through the Mask R-CNN detector, the number of outputs can vary depending on how many detected objects have a confidence score higher than 90% (as discussed in Section 4.4.1). If no detections reach this threshold the image is discarded from further processing. This can occur if, for example, an image is taken by accident capturing

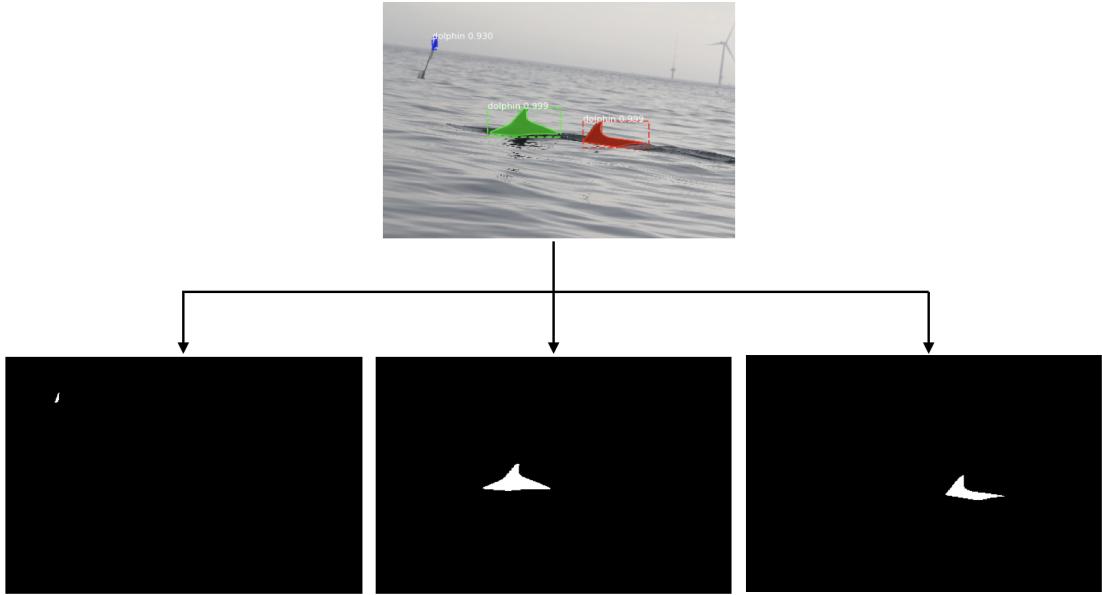


Figure 5.2 A visualisation of the three dolphin detections for an input image produced by the Mask R-CNN detector. The detections are highlighted by the blue, green, and red overlays on the input image, top. The resultant detection masks once split are shown bottom, where a dolphin object is displayed in white.

the vessel's floor. If a single detection reaches the threshold, the resultant mask is passed downstream.

Thanks to the tendency for cetaceans to travel in pods however, it is unlikely an image will contain only a single dolphin detection. In these situations the Mask R-CNN will output multiple detection masks, one per object. To handle this, the first stage of the post-processing methodology is to separate multiple detections for a single image. This ensures each dolphin detection is handled independently, mitigating the potential for an identification to be influenced by the others. An example of this behaviour can be seen in Figure 5.2. Here, an image inputted to the Mask R-CNN detector has produced three detections which are above the threshold. As such, they have been split into three output masks for individual processing. Note that the first detected mask, visualised with a blue overlay on the input image, has been incorrectly detected as dolphin with a high confidence resulting in a binary mask. Further post-processing must be capable of handling background which has been incorrectly labelled, removing it before individual identification occurs.

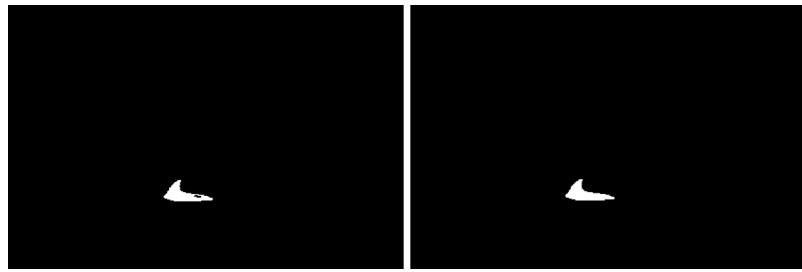


Figure 5.3 Left: a detection mask before closing has been applied. The detected dolphin object is displayed in white. Note the cluster of black background pixels inside. Right: the same detection mask after closing. Note the pixels which make up the hole have been converted to dolphin.

## 5.2 Morphological Transformations

In some situations a detected mask may contain an area of background inside the detection. This can be thought of as a hole in the detection, as seen in Figure 5.3 (Left). Using *a priori* knowledge of cetaceans, which would rarely if ever be captured with a hole, it can be deduced that a hole in the detection is highly unlikely and may cause a loss of useful identifying information. As such, any holes which are present in the masks must be filled.

This is achieved using morphological transformations, a set of operations which allow for the automated manipulation of the internal structure of binary images such as masks. The two fundamental morphological transformations are erosion, which erodes away the boundaries of the masked object, and dilation, which increases the size of the object by pushing the boundary out into the background space. These two operations can be utilised in various combinations to perform more complex transformations.

In order to remove a cluster of background pixels inside of a detection, each mask is closed – dilated then eroded. This has the effect of removing any holes present inside the mask, as can be seen in Figure 5.3 (Right). If no holes exist, the operation is still performed, however the mask remains unchanged. By performing closing, the system ensures that no potentially identifiable information is lost as a result of an incomplete detection.

## 5.3 Background Subtraction

Now that the masks have been cleaned using morphological transformations, it is possible to utilise them to perform background subtraction. This is an extremely important step in producing an accurate individual classification based on the detected dolphin object by ensuring a minimal amount of background noise is passed to the identification system.

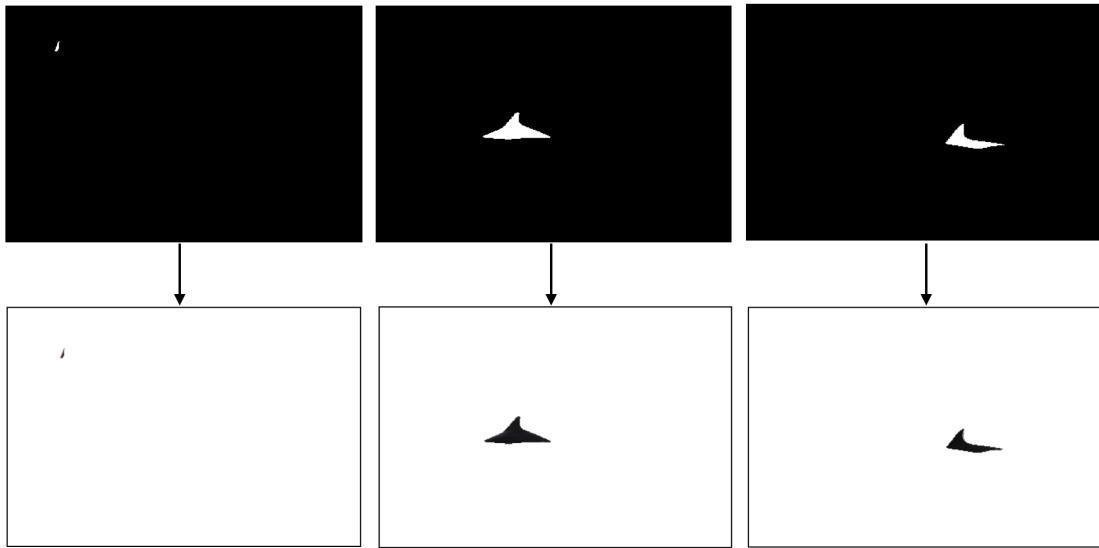


Figure 5.4 A visualisation of the three `dolphin` detections from Figure 5.2 before and after background subtraction. A border has been added to the background subtracted images for clarity.

As both the input image and resultant mask can be represented as matrices, these can be manipulated utilising a *bitwise and* operation such that if pixel<sub>*i,j*</sub> in the input image is denoted as background in the mask, the values of pixel<sub>*i,j*</sub> can be set to [255, 255, 255] (white). This has the effect of whiting out any pixels not detected as part of the `dolphin` in the input, whilst keeping the pixels detected as `dolphin` intact.

An example of background subtraction utilising cleaned masks can be seen in Figure 5.4. Using the same input image as in Figure 5.2, it can be seen that the *bitwise and* operation whites all pixels in the image except those which have been classified as `dolphin` for each of the three output masks. Note at this point that the erroneous classification still remains.

Whilst the background subtraction aims to reduce noise passed downstream to the individual identification module, it will not be possible to remove all noise. It may be the case, such as in Figure 5.5, that some background pixels are mislabelled as `dolphin` but are connected to the edges of a correct detection. As a result, morphological transformation and background subtraction are unable to remove the mislabelled pixels. This may affect the accuracy of identification downstream unless the system is robust enough to deal with this.

## 5.4 Colour Thresholding Mask Components

In some cases a single detection mask may consist of multiple components. This may occur if, for example, an area of splash has been erroneously included as part of a detection. As

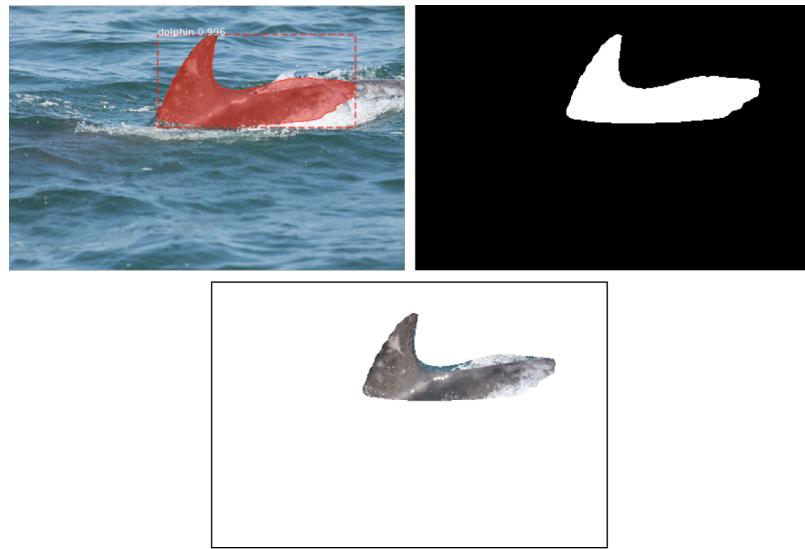


Figure 5.5 Top Left: a visualisation of the dolphin detection for an input image produced by the Mask R-CNN detector, alongside the confidence score. The detector has incorrectly labelled some pixels as dolphin. Top Right: the resultant detection mask after morphological transformation. Bottom: the resultant output image after *bitwise and* operations performed between the input image and the cleaned detection mask. Note the mislabelled pixels are present after cleaning and background subtraction. A border has been added for clarity.

cetaceans cannot be made up of multiple disjoint components, it is known that some of these must be noise and can be removed.

The outer layer of a cetacean’s skin is often a consistent grey colouring. This information can be utilised to filter out noisy components of the mask during post-processing. By comparing the colour composition of each detected object against a calculated *dolphin-like* threshold, it is possible to discard mask components which have been erroneously detected.

In order to be able to compare each mask component’s composition against a *dolphin-like* colour threshold, the values of the threshold must first be determined. To achieve this, each image in the Zanzibar dataset was run through the Mask R-CNN detector. Histograms of the three RGB colour channel pixel intensities for each object classification (dolphin or background) were recorded, giving a total of six histograms per image.

Once complete the histogram groups were combined to give six global pixel intensity distributions, which can be seen in Figure 5.6. From the charts it can be seen that, regardless of colour channel, there is a near inversion in the distribution of pixel intensities between those detected as dolphin and those not, strongly suggesting it is possible to determine whether a component is erroneous based on its colour composition.

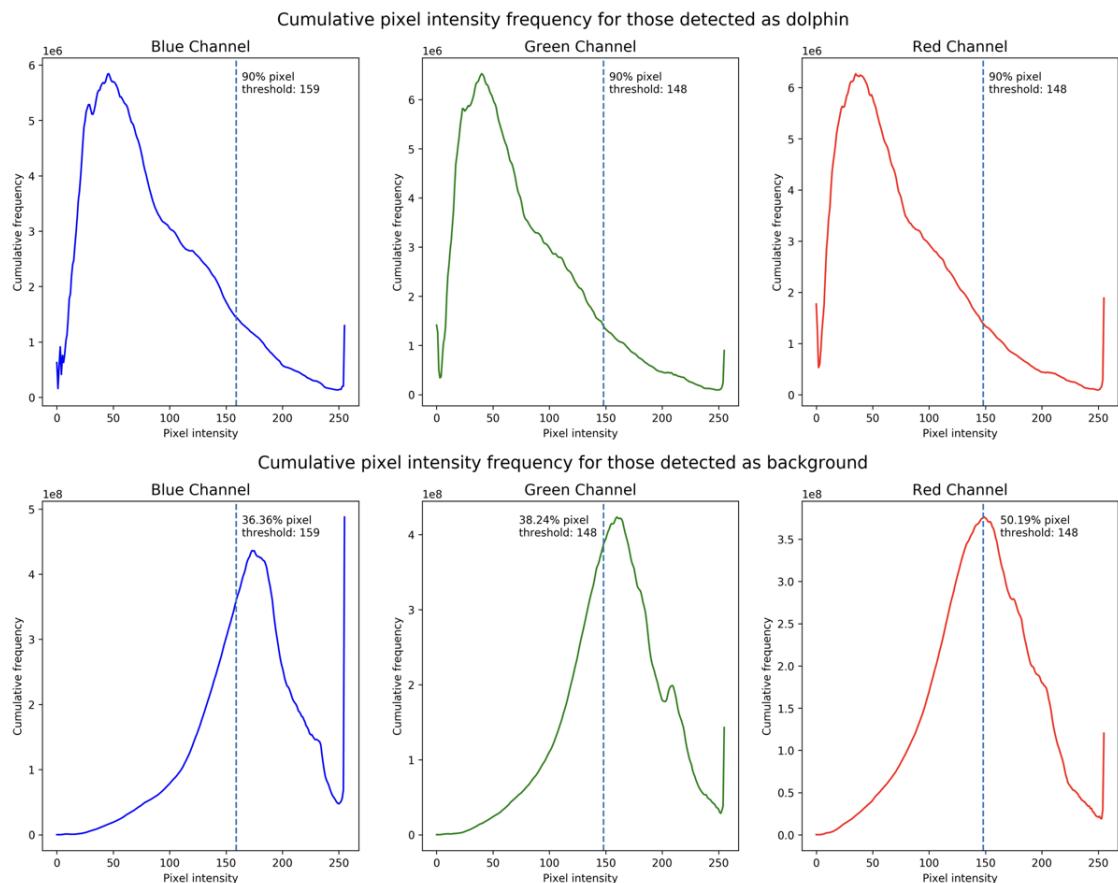


Figure 5.6 The global range of pixel intensities for each RGB colour channel, split by pixel classification.

Using the global distribution histograms, a *dolphin-like* threshold was determined. For all masks detected as *dolphin*, 90% of the RGB pixel intensities are below [148, 148, 159]. In contrast only 50.19%, 38.24%, and 36.36% of the background pixels for each RGB channel respectively are below this threshold. As noise components in the mask are often areas of water or splash, these components will be likely much lighter in composition than cetaceans, and thus can be removed from the mask with confidence. An example of colour thresholding removing noise from a mask can be seen in Figure 5.7.

During testing however it was found that when checking detections at an individual image level rather than globally, considering a mask component to be *dolphin-like* if 90% of the pixels were below the colour threshold was too restrictive. Utilising such a high percentage bar sometimes rejected valid detections which may have been over-exposed due to lighting conditions. As such, whilst the colour threshold was kept the same, it was found that reducing the percentage check to 50% provided enough leeway such that over-exposed but valid detections were kept whilst still rejecting a large portion of erroneous ones. Figure 5.8 shows an example detection which would be discarded if the 90% global threshold was used per image, but is retained if this is dropped to a 50% check.

It may be the case that multiple components in an image meet the conditions set by the threshold to be kept. In this case, each component of the mask is split into its own image, in a similar process as outlined in Section 5.1. If a mask only contains one component, then colour thresholding is not applied. This ensures no detections by the Mask R-CNN are completely ignored due to post-processing, preventing the discarding of a *dolphin* object mask which contains no disjoint components but is above the threshold, such as in the event of over-exposure. This condition also has the effect however of allowing fully erroneous detections to pass downstream, such as the flag detected in Figures 5.2 and 5.5. Any erroneous detections which pass downstream at this stage must now be handled by the identification module.

## 5.5 Cropping

At this point outputs from the Mask R-CNN detector have been post-processed to remove as much noise from the detected masks as possible, and these have been utilised to perform background subtraction. This results in an output image containing mostly white pixels surrounding a detected *dolphin* object. This image is the same size as the one inputted to the detector, which can often be many thousands of pixels, although the vast majority of these are now un-needed.

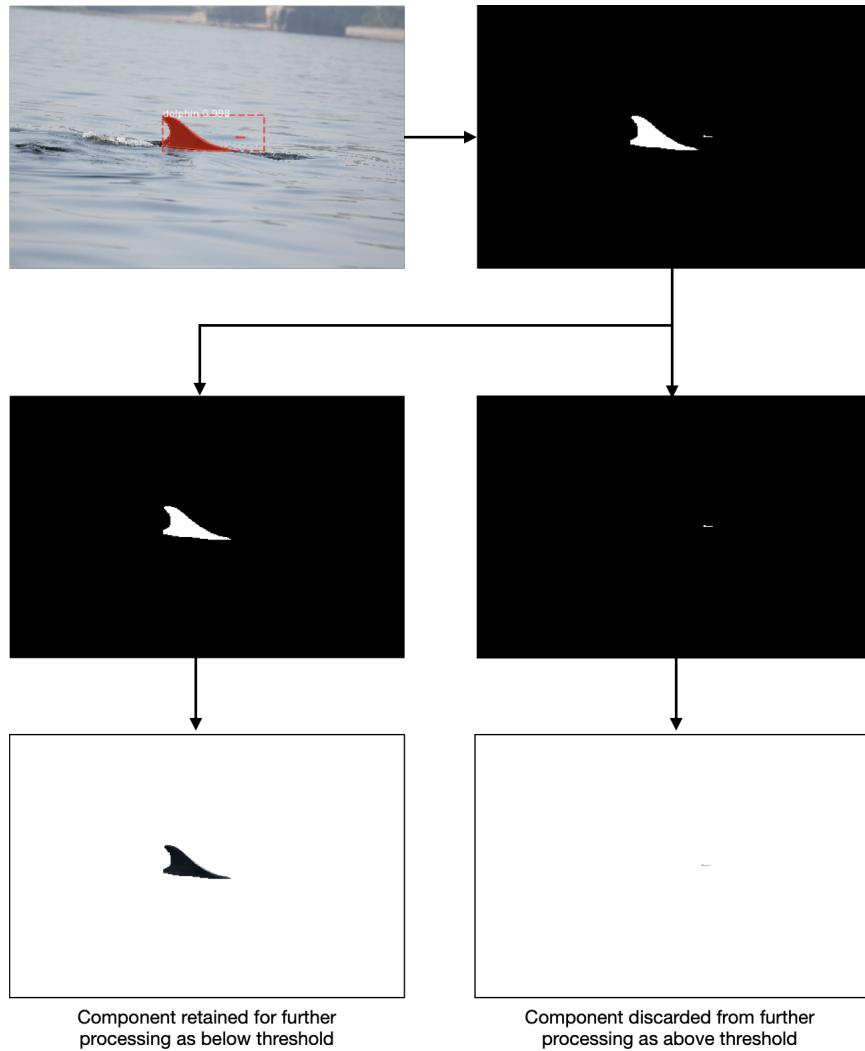


Figure 5.7 Workflow detailing colour thresholding to remove an area of disjoint splash which has been detected as part of a dolphin object. The detection mask is split into each component. The resultant background subtracted images are then colour thresholded. As a result, the erroneous splash is discarded.

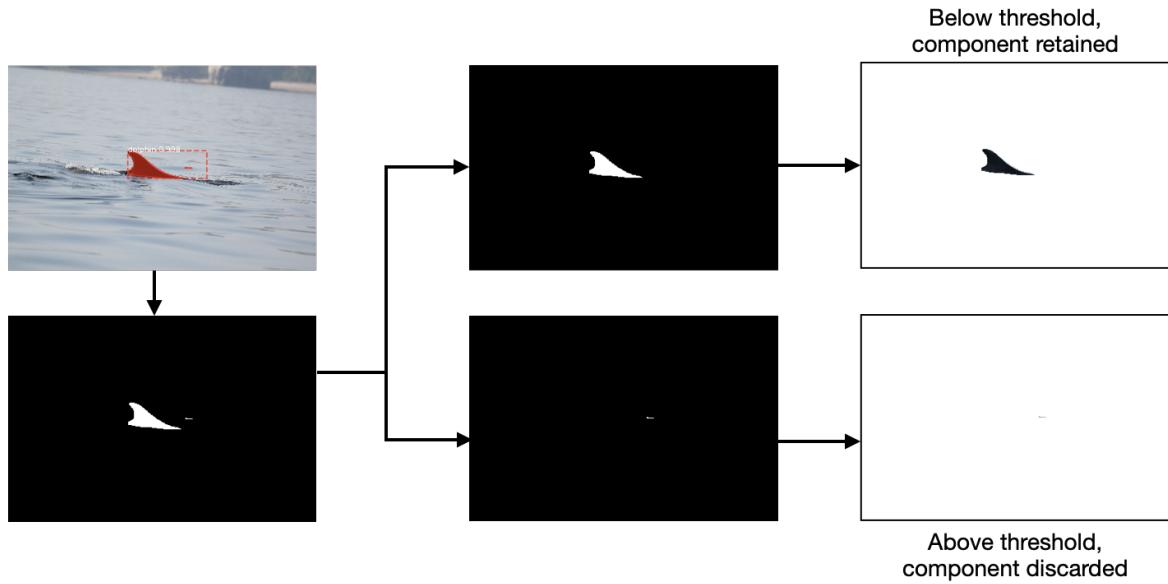


Figure 5.8 Workflow showing how utilising the 90% global threshold may lead to over-exposed detections being erroneously discarded. A dolphin object is detected whose mask consists of multiple components. One of the components, which contains a valid area of detection, is shown in stage 3 of the workflow. Checking to see if 50% of the pixels in the component are below the threshold retains the detection, whilst checking at 90% discards it.

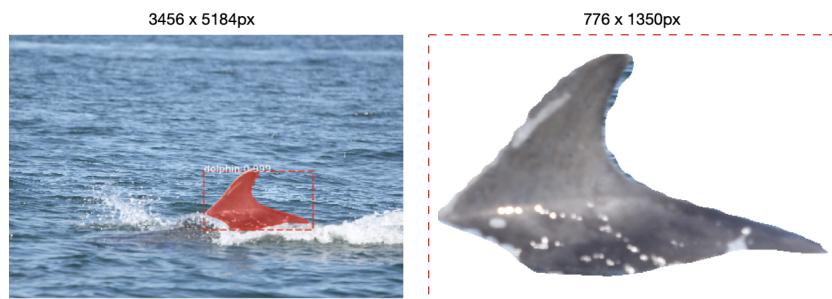


Figure 5.9 Left: an input image and overlaid detection mask. Right: the corresponding cropped output image after post-processing. Original image sizes are displayed above each image. Images have been resized for clarity.

As a result, the images outputted from the background detector are now cropped down to contain just the object of interest. This has the effect of vastly reducing the image file size, which in turn reduces the computational expense of operating on them downstream. In Figure 5.9 for example, the input image to the detector is of size 3456x5184 pixels. After post-processing, the resulting output image is now of size 776x1350px, approximately a 94.2% reduction in image size whilst still keeping identifying information, such as the white pigmentation on the dorsal fin, present. This cropping also has the effect of centring the detected dolphin objects in the final output image.

## 5.6 Post-Processing NDD20

NDD20 consists of large scale panoramic images, similar to the Zanzibar data before post-processing. As a result, there is a high amount of noise present in the images which should be ignored and removed before the dataset is utilised for individual identification. Whilst Chapter 6 will focus on both the theory behind, and implementation of, a model capable of individual identification, this section will detail the processing of NDD20 into a form usable for the task of individual identification.

Utilising the post-processing techniques as outlined previously in this chapter, the detections from the Mask R-CNN model were used to generate a dataset of images to train another model capable of individual identification. This dataset, known as Segmented NDD20, is a collection of images which contains only the segmented masks from NDD20.

Once images of the segmentations had been created these were then processed further to create a folder structure which allowed for easy training of an identification model. To facilitate this, each segmentation was checked against the ground truth for the image it was produced from. Any images which contained a dorsal fin with identifying information were placed in a directory containing other examples of that individual. Any fins that did not contain enough identifying information, for example due to poor segmentation such as those seen in Figure 5.10, were removed. Whilst this removal may reduce the difficulty of the problem, as the identity of the individuals could not be confidently verified manually this also ensured mislabelling issues were not present. Noise which had passed through the mask post-processing was included in a `noise` directory with the goal of allowing any future model to learn how to identify erroneous masks that have made it through post-processing.

Example images from Segmented NDD20 are shown in Figure 5.11. As can be seen there is low inter-class but high intra-class differences. For example there is relatively little difference between the images shown for individuals 32 and 39, whilst there is a large difference between the three example images for individual 11. It is this variance and fine-

grained nature which makes the task of automatic individual photo-identification particularly challenging.

In total, Segmented NDD20 contains 1243 images representing 43 classes including **noise**. Individuals 6 and 27 were removed from the dataset by the post-processing algorithm. Upon examination, images of these individuals were captured from extreme angles or with large amounts of splash, making it difficult for the detector to produce accurate segmentations. The dataset is highly skewed with approximately 66% of images in the dataset labelled as **noise**. Classes representing individual animals contain a non-uniform number of example images (median = 8.5) varying between 33 examples for individual 11 to just one for individuals 5 and 35. As such this dataset represents not just a fine-grained but also a few-shot problem, a combination which is extremely challenging for current computer vision methods.

### 5.6.1 Additional External Data

After post-processing the data collected during fieldwork into the Segmented NDD20 dataset, it was clear from the class distribution present that it may be beneficial to increase the number of examples per individual, providing a photo-id model more data to learn from. As it was not possible due to sea state conditions to continue data collection in the Coquet to St. Mary's MCZ after 10/10/2019, work was undertaken to procure data from other external sources.

As a result of the large range traversed by cetaceans [241] there was a chance that the individuals catalogued in the MCZ had also been recorded in other areas. The underwater habitat found in the MCZ extends upwards as far as the Moray Firth in Scotland. This, along with the knowledge the animals prefer colder waters, led to the assumption that the animals found in Northumberland would likely also be found in catalogues maintained further north. Examination of photo-id catalogues from the University of Aberdeen, provided after discussions with the manager of the University's long-term bottlenose dolphin studies Dr. Barbara Cheney, confirmed a catalogue overlap. It was determined that 23 individuals



Figure 5.10 Example images removed when post-processing NDD20 as their identity could not be verified manually.

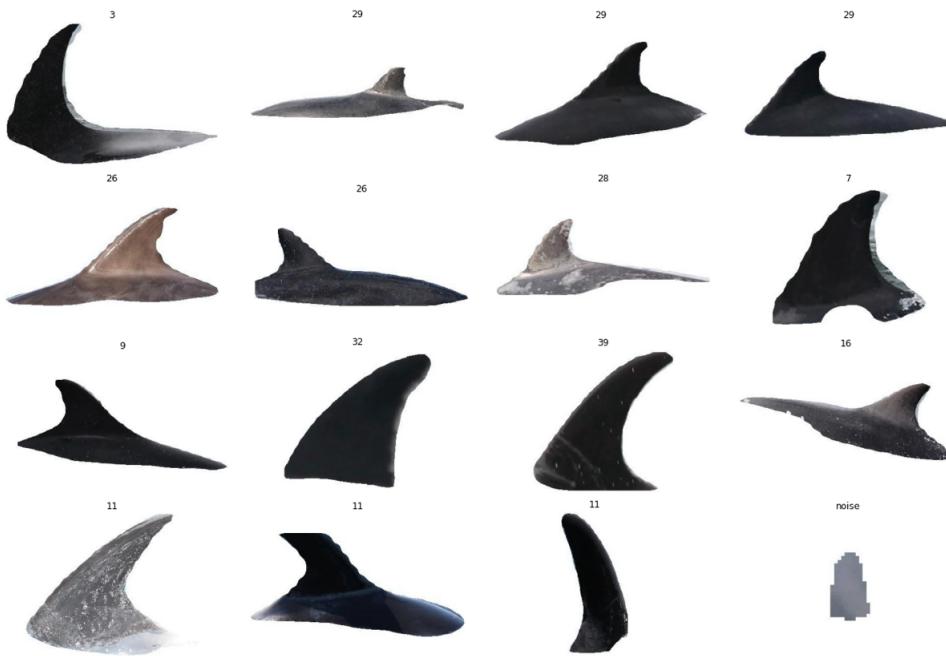


Figure 5.11 Example images from Segmented NDD20. The individual class label is displayed above each image.

(all bottlenose dolphins) were present in both catalogues. As a result of this analysis, Dr. Cheney provided 1827 additional images of the overlapping individuals to complement Segmented NDD20. Images provided were collected by both the University of Aberdeen and the University of St. Andrews' Sea Mammal Research Unit (SMRU) with the approval of Dr. Mònica Arso Civil. These images were captured during surveys undertaken between 2003-2019 and were of the highest quality rating on the scale used by the institutions. All images contained only a single individual.

The additional external data procured was passed through the Mask R-CNN detector and post-processing algorithm, producing images similar to those already present in Segmented NDD20. The post-processed Aberdeen and SMRU data was combined with Segmented NDD20 to produce the NDD AU SMRU dataset, consisting of 2626 images (1383 more than Segmented NDD20) representing 44 classes including *noise*. Individual 6, whilst not present in Segmented NDD20 is accounted for in NDD AU SMRU, however individual 27 still remains absent. Like Segmented NDD20, NDD AU SMRU is heavily skewed towards *noise* with 61% of images labelled as this class. Non-noise classes however now have a median of 22 examples per class, providing a larger number of example images to train an automatic photo-id model from.

Dataset	Number of Images	Number of Individuals	Noise Class Included?	Total Number of Classes
<b>NDD20 (Above Water)</b>	2201	44	✗	44
<b>NDD20 (Below Water)</b>	2201	82	✗	82
<b>Segmented NDD20</b>	1243	42	✓	43
<b>NDD AU SMRU</b>	2626	43	✓	44

Table 5.1 A comparison between the NDD20, Segmented NDD20, and NDD AU SMRU datasets.

## 5.7 Summary

This chapter discusses the creation of a post-processing methodology for dorsal fin detections generated by the Mask R-CNN model as outlined in Chapter 4. The model’s outputted detections are coupled with post-processing techniques capable of greatly reducing the amount, and improving the quality of, data on which subcomponents further into the system pipeline are required to operate. Whilst no quantification of the post-processing methodology is undertaken, the inclusion of these stages in the pipeline allows for greater computational efficiency downstream as well as more accurate individual identifications.

The developed post-processing methodology is then utilised for the creation of Segmented NDD20, a dataset to be utilised for the training of a neural network capable of individual identification. Finally, the NDD AU SMRU dataset is created by combining Segmented NDD20 and additional data provided by collaborators at the Universities of Aberdeen and St. Andrews after a cross-catalogue matching process was undertaken, highlighting a 23 individual overlap between the catalogues maintained by Newcastle University and the two collaborating universities. A comparison between the NDD20, Segmented NDD20, and NDD AU SMRU datasets is provided in Table 5.1.

Masks post-processed by the methodology outlined within this chapter are then passed to the next stage of the system pipeline, consisting of a model trained to perform accurate most likely catalogue matching. This model is examined in detail in Chapter 6.



# **Chapter 6**

## **Individual Cetacean ID via Automatic Most Likely Catalogue Matching**

This chapter examines the final component in the automatic photo-id pipeline, focussing on individual identification. The component takes as input photo-id catalogue images which have been passed through the dorsal fin detector and post-processing methodology outlined in Chapter 4 to produce a list of most likely catalogue matches. It is important to note here that this component is not intended to replace photo-id researchers by performing the job for them. Instead, the component aims to vastly reduce the search space the researcher needs to examine in order to verify a catalogue match; a list of most likely matches is suggested, but ultimately the final decision lies with the researcher.

To begin, the requirements an automatic most likely catalogue matching system must meet are outlined. The chapter then discusses possible approaches to the problem and justification for the selected approach. Model development is discussed in detail, using the NDD AU SMRU dataset created in Section 5.6.1 for training and evaluation. The effect of class structure and how this impacts most likely catalogue matching accuracy is discussed, alongside the current limitations of the approach.

### **6.1 Most Likely Catalogue Matching System Requirements**

Before development can begin, it is important to outline the system requirements. Unlike the detector which could be considered a coarse-grained task, identification of individual cetaceans is an extreme fine-grained problem as they are distinguished from each other

using small prominent markings present on the dorsal fin. As the animals are free roaming, there can be high variation in how the fin is captured in the image, discussed in greater detail in Section 4.1.1. This can lead to photo-id catalogues with low inter-class but high intra-class differences between the individuals present, seen in Figure 5.11. As a result of this, any system capable of accurate catalogue matching must be able to recognise these minute differences between individuals even when there is high variation in the examples for each individual class.

The system must also be capable of operating using all information provided to it. Other photo-id aids which perform most likely catalogue matching such as finFindR [201] operate using only the trailing edge of the fin, with matching performed using notches and shape. This misses other prominent markings such as long term scarring or pigmentation, as well as the shape of other fin edges. As such, it may be the case that finFindR struggles when operating over a catalogue with few to no notches. To avoid this issue, the system developed must be capable of matching using all available prominent markings.

Further, the system must also be capable of performing accurate catalogue matching under the presence of noise, both classified and misclassified. Datasets developed for the training of this system such as NDD AU SMRU contain a noise class which encapsulates all detected mask components which are erroneously retained after post-processing has been applied. This class has extremely high intra-class variance, however it is imperative the system is able to match erroneous components to it. Misclassified noise is defined as that which has been passed downstream as a result of being attached to a valid individual detection mask. In Figure 6.1 for example, the swell captured in the post-processed crop would be considered misclassified noise. Any system performing automatic most likely catalogue matching must be resistant to small amounts of misclassified noise in order to produce accurate identity suggestions.

Any developed system must also be capable of handling examples of individuals which are not present in the photo-id catalogue. Due to the free roaming nature of cetaceans (or indeed any wild animal) and the limitations on photo-id survey size dictated by both weather and workforce, there is no guarantee that every animal who makes use of the survey area will be captured. New animals may also become resident in the area through birth or migration. When these animals are eventually captured during a survey and their image processed, the system must be capable of recognising this as an individual not currently present in the catalogue and highlight this to the researcher. This is made more difficult given the extreme fine-grained nature of the catalogues. As a result, this requirement necessitates that the system must be capable of recognising uncertainty or understand a notion of similarity between an input and the class examples present in the catalogue.



Figure 6.1 Left: An example input image. Right: The corresponding post-processed crop which contains some misclassified noise.

Traditional computer vision classification models do not meet this requirement. If an example image of a new individual was seen by a traditional CNN trained on a photo-id catalogue dataset, this model would still attempt to provide a classification based on the classes present at train time. As deep learning based vision models operate on point estimations of parameters, unlike Gaussian processes where the probability distribution is defined over a function, this removes the ability to produce helpful indicators of uncertainty such as prediction confidence bounds [263].

## 6.2 Siamese Neural Networks

Rather than producing a direct classification, Siamese Neural Networks (SNNs) aim to incorporate the notion of similarity into the model. This is achieved by connecting two or more identical CNNs in parallel, each sharing the same backbone architecture, initial and updated weights, and hyperparameters. Each CNN in the SNN is designed to produce an embedding, or a  $d$ -dimensional representation, of the input. The size of this embedding is set via hyperparameter and dictates how many  $d$  dimensions the output of the SNN will be. For example, if an SNN is created with an embedding size of 10, each CNN may take a high dimensional input of size  $width * height * channels$  and output a 10-dimensional embedding, a float vector of size 10 which represents the input image. A visualisation of a two branch SNN can be seen in Figure 6.2.

At train time, each CNN branch receives a different image and generates an embedding. These embeddings are compared to one another in order to optimise a loss function, whereby

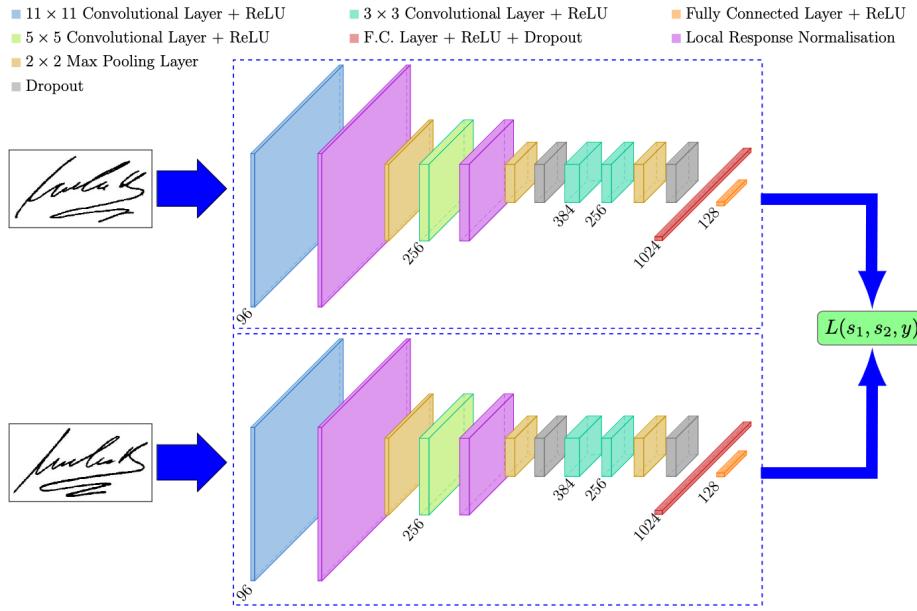


Figure 6.2 An example two-branch SNN architecture for signature verification. Each branch takes as input an image of a signature and passes this through a CNN, where each branch's CNN is identical. A 128-dimensional embedding is produced as output by each branch. Embeddings are then compared using some loss function,  $L$ , to generate a similarity value. Image from [37].

input images of the same class have similar embeddings but those of different classes are dissimilar. In this way the SNN can be tuned to provide a measure of image similarity. Once trained only one branch of the model is retained, allowing a single image to be embedded by the model into the same embedding space and compared to previously embedded images.

It is this ability that has resulted in the widespread use of SNNs for verification or identification problems in computer vision [37, 264]. Specifically in ecology, SNNs have found use in fine-grained species identification problems [265, 266] as well as in more extreme fine-grained individual animal identification [30] and behaviour classification [267].

### 6.2.1 Clustering Embeddings in a Latent Space

By storing the embeddings generated for each trained class it is possible to produce a list of likely class predictions for a new image by measuring the distance between the generated embedding and those previously produced when plotted into some  $d$ -dimensional latent space. This thesis makes use of Euclidean distance measurement, though the use of cosine distance measurement is also present in the literature [268–270]. If the SNN has trained in such a way as to produce low intra-class, high inter-class difference between generated embeddings then this will create class clusters when plotted in the latent space.

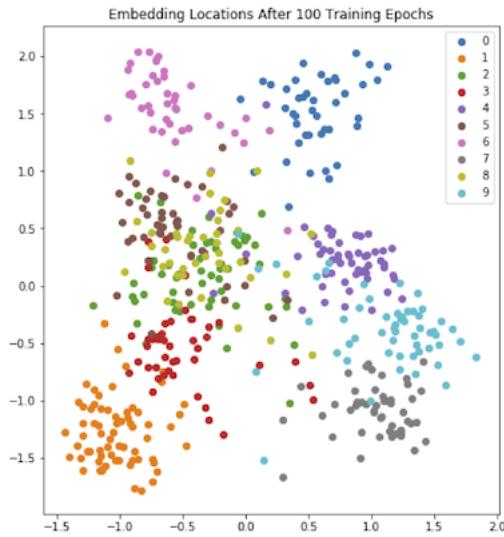


Figure 6.3 A 2-dimensional visualisation of a multi-dimensional latent space produced by an SNN trained on the MNIST dataset [68] for 100 epochs.

An example of this behaviour can be seen in Figure 6.3 which shows a 2-dimensional visualisation, produced using Principal Component Analysis (PCA), of the embedding locations for a subset of the MNIST dataset [68]. Here, an SNN has been trained for 100 epochs using a Triplet Ranking Loss (outlined in Section 6.2.3) to generate embeddings of images for the 10 unique classes. As can be seen, the model is able to generate embeddings in such a way as to cluster those of the same class in the latent space. Note that some clusters are visualised on top of each other due to the dimensionality reduction performed in order to show the latent space on the page.

It is important to note here that the value of the embeddings is not necessarily important, just the distances between them. Notice how all points in Figure 6.3 lie within approximately -1.5, 2.0 on the x-axis and -2.0, 2.0 on the y-axis. There is nothing inherently good or bad about an SNN that embeds within this range, all that matters is the points are clustering in their respective classes.

### 6.2.2 Meeting the Outlined Requirements

The computational expense of performing inference with an SNN is relatively small. Whilst training requires the use of a branched CNN architecture in order to optimise the loss function, this is reduced to just one branch at inference time. Generating a list of most likely catalogue matches would only require an image to be passed through the network once in order to generate an embedding, and similarity via Euclidean distance measurement in the latent

space is cheap to perform. As such, producing a list of most likely catalogue matches is overall computationally efficient using SNNs.

The clustering of class embeddings in the latent space also allows for easy identification of potential previously uncatalogued individuals. Passing the dorsal fin of an individual not present at train-time through the model would result in, theoretically, a distinct embedding which would plot into a unique point in the latent space far from any existing class clusters. By implementing a threshold on the Euclidean distance measurement, potentially uncatalogued individuals could be easily flagged to the researcher for further investigation. Clustering also removes the need for re-training to allow for matching to previously uncatalogued individuals when they are added to the catalogue. Adding a new class to the latent space can be achieved simply by defining embeddings to a new class cluster and including these in future distance measurements.

In addition, SNNs are capable of operating over all information provided to them. This can be achieved by not limiting the embedding generation to one specific part of the dorsal fin. It is for these reasons the decision was taken to first begin development of a model capable of most likely catalogue matching using SNNs.

### 6.2.3 Pairwise vs Triplet Ranking Loss

Training of any neural network is performed through the optimisation of a loss function. For SNNs, a group of loss functions known as Ranking Losses is utilised. Here, the goal is not to predict a class label but rather a distance between model inputs. As such, they are perfect for training SNNs.

During training an SNN will generate embeddings for some received inputs and generate a similarity value (e.g. via Euclidean distance when plotted into a latent space). This similarity value is then used to optimise the Ranking Loss, which in turn tells the model how to modify embeddings to achieve better overall performance, for example how to bring two embeddings closer when they are of the same class. The type of Ranking Loss utilised for training and the number of branches present in the SNN are intertwined. Two of the most commonly used Ranking Losses are Pairwise Ranking Loss and Triplet Ranking Loss.

#### Pairwise Ranking Loss

SNNs which make use of two branches can be optimised using a Pairwise Ranking Loss [271], a visualisation of which can be seen in Figure 6.4. Here the model is trained using data points made up of two inputs. The first input is called the Anchor, which defines the class the model is training to optimise for. The second input can be either a Positive containing

another example of the Anchor class, or a Negative containing an example of some class other than the Anchor.

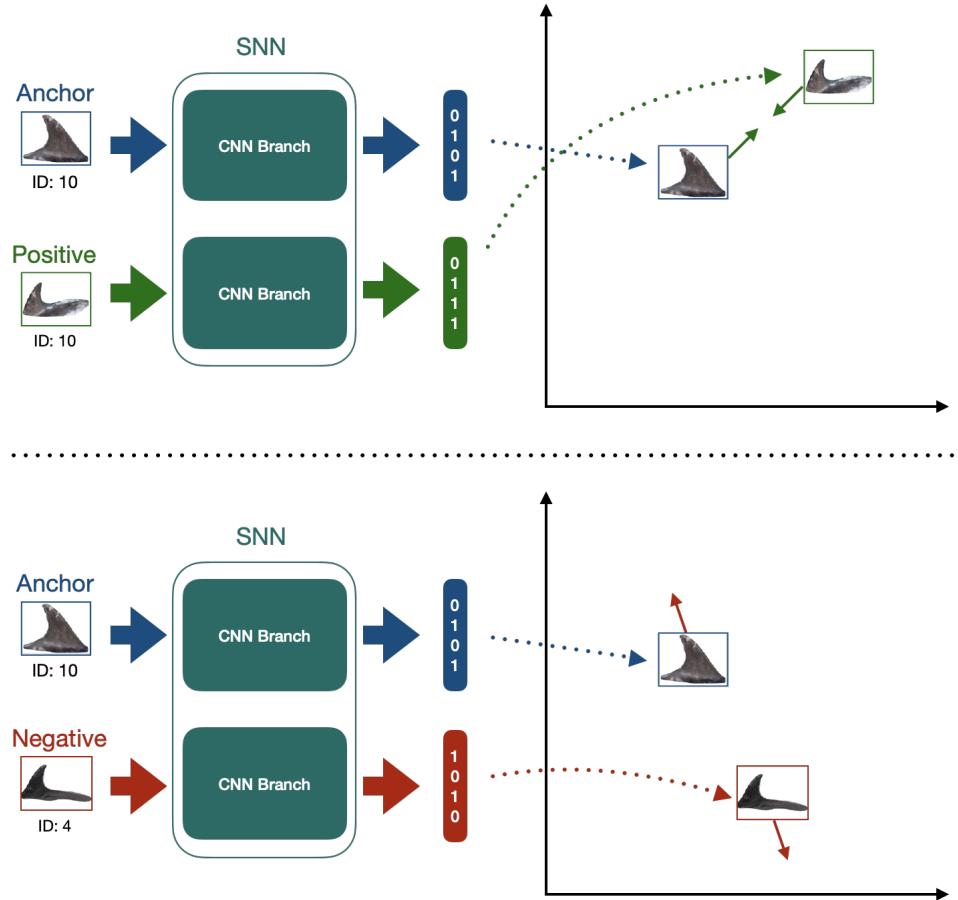


Figure 6.4 SNN optimisation using Pairwise Ranking Loss. Each input is passed to a branch of the SNN and an embedding is produced. These embeddings are used to optimise future embedding generation, aiming to either pull the Anchor and Positive together or push the Anchor and Negative apart.

Using these two input types Pairwise Ranking Loss can be used to optimise in such a way that the model learns to produce embeddings with a small distance between Anchors and Positives, and a large distance between Anchors and Negatives. Mathematically Pairwise Ranking Loss can be defined using Equation 6.1:

$$L = \begin{cases} D(A, P) & \text{if Positive Pair} \\ \max(0, m - D(A, N)) & \text{if Negative Pair} \end{cases} \quad (6.1)$$

Where  $L$  is the loss,  $D(A, P)$  is the distance between the Anchor and the Positive, and  $D(A, N)$  is the distance between the Anchor and the Negative. When optimising for Positive Pairs the

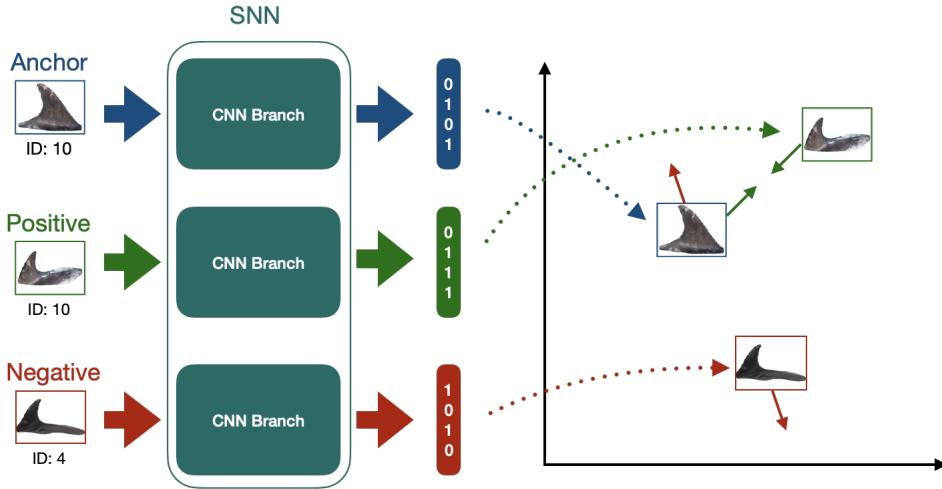


Figure 6.5 SNN optimisation using Triplet Ranking Loss. Each input is passed to a branch of the SNN and an embedding is produced. These embeddings are used to optimise future embedding generation, aiming to both pull the Anchor and Positive together whilst pushing the Anchor and Negative apart.

loss function will only ever return 0 when the distance between the Anchor and the Positive is 0, ensuring these embeddings are nearly always pulled closer. When optimising for Negative Pairs, the loss function will return 0 when the distance between the Anchor and the Negative is greater than some margin  $m$ . As such, a weight update is not performed when the distance between the Anchor and the Negative is sufficiently large.

### Triplet Ranking Loss

One of the main problems presented by Pairwise Ranking Loss is the issue of model collapse, occurring after a large amount of Positive Pair optimisations. In this scenario, the distance between Anchors and Positives is pushed so close together in the latent space as to produce the same embedding. This can in turn affect the model's ability to understand variation in input and similarity scoring.

Triplet Ranking Loss aims to avoid this issue by training on triplets of data points rather than pairs, with each triplet containing an Anchor, a Positive, and a Negative. SNNs which make use of Triplet Ranking Loss are often named Triplet Networks in the literature [272], however the only difference between the structure of an SNN using Pairwise Ranking Loss or Triplet Ranking Loss is the number of branches – two or three respectively.

Just like with Pairwise Ranking Loss, Triplet Ranking Loss takes as input the generated embeddings for each branch and optimises to pull the Anchor and Positive close whilst

pushing the Negative away, as visualised in Figure 6.5. Optimisation is performed using Equation 6.2:

$$L = \max(0, D(A, P) - D(A, N) + m) \quad (6.2)$$

By utilising a triplet, the loss function evaluates to 0 when  $D(A, N) > D(A, P) + m$ . This occurs only when the triplet contains examples the model is already well trained on and no further optimisations can be gained. By enforcing  $m$ , where typically  $m = 0.2$  thanks to work by Schroff *et al.* [38], the function ensures embedding variation between distinct inputs thus allowing for a similarity score to be computed between the Anchor and the Positive in all cases. Thanks to the advantages of Triplet Ranking Loss over its Pairwise counterpart, the decision was made to make use of this loss function and train an SNN with three branches. Further, Triplet Ranking Loss has been shown to perform well on individual identification tasks in both humans [273] and animals [265], providing evidence to support its use for training a most likely catalogue matcher.

#### 6.2.4 Semi-Hard Triplet Mining

When training a model, care should be taken to ensure learning occurs at every step. When using Triplet Ranking Loss however, learning does not occur during training steps where the loss evaluates to 0, such as when  $D(A, N) > D(A, P) + m$ . Negatives provided should be sufficiently difficult such that the triplet allows the loss to evaluate to a non-zero value, ensuring the model learns and the training step is not wasted. However care should also be taken so as not to provide the model with triplets that are too difficult, as this will increase optimisation and thus overall training time.

This leads to somewhat of a Goldilocks problem. Triplets must be not so soft as to prevent learning, but not so hard as to dramatically increase training time. Semi-Hard Triplet Mining aims to fix this problem, providing triplets which are *just right*. A triplet,  $T$ , is defined using Equation 6.3:

$$T = \begin{cases} D(A, P) + m < D(A, N) & \text{if Easy} \\ D(A, N) < D(A, P) & \text{if Hard} \\ D(A, P) < D(A, N) < D(A, P) + m & \text{if Semi-Hard} \end{cases} \quad (6.3)$$

The goal of Semi-Hard Triplet Mining is to locate as many Semi-Hard triplets from the training set as possible. These are triplets whereby the loss still evaluates to a positive value however the Anchor is closer to the Positive than the Negative when plotted in the latent

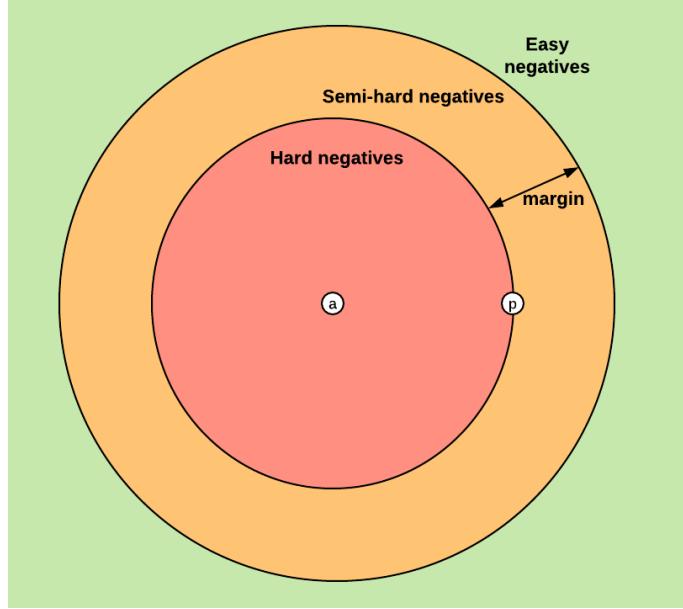


Figure 6.6 A visualisation of the areas in the latent space where Easy, Hard, and Semi-Hard triplets can occur, where  $a$  is the location of the Anchor and  $p$  is the location of the Positive. Image from [274].

space, as seen in Figure 6.6. This allows for fast training whilst still providing enough triplet difficulty for the model to learn during training.

Finding, or mining, Semi-Hard triplets can be performed either Offline or Online. In Offline mining, the entire training set is converted into triplets before the training epoch occurs and those that fit the Semi-Hard definition are utilised. With Online mining, Semi-Hard triplets are generated on the fly as required. Generally, Online mining results in faster training when compared to Offline mining as this allows for the ability to update our definition of a Semi-Hard Triplet as training progresses.

### 6.2.5 Class Prototyping

After SNN training it is possible to obtain likely classifications for an input based on Euclidean distance measurements between the input's embedding and the previously generated embeddings when plotted into the latent space. If there is a large number of embeddings in the space however this can increase classification time, as the input's embedding must be checked against every other in the space.

There are ways to reduce the time taken for this calculation to complete by reducing the number of distance measurements that occur. A naive approach would be to, for example, randomly select one embedding for each class and measure the distance between it and the

input embedding such that the distance to each class is only measured once, vastly reducing the computation required for classification. This may only work however when the class embeddings are perfectly clustered in the latent space, which will likely not be the case when using real world data.

There may also be cases where embeddings are not clustered with their class, such as in Figure 6.7 where an embedding of class **cross** has been generated such that it is surrounded by examples of class **square** – far from the other **cross** examples. There is also a triangle in the top-right of the Figure which represents the embedding location of an unclassified inference image.

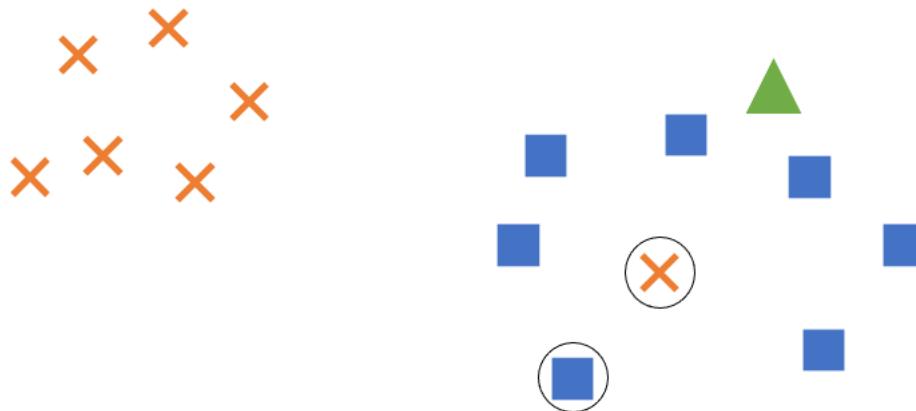


Figure 6.7 An example latent space with two classes (**cross** and **square**) alongside a triangle which represents the embedding location of an unclassified inference image. The two class examples selected for distance measurement to classify the triangle using the naive approach are circled.

Using the naive approach to classify this triangle as either a **cross** or **square**, assuming the two randomly selected class embeddings are the ones circled in the Figure, then the triangle would be classified as an example of class **cross**. However, looking at the space globally it is clear the triangle should more likely be classified as a **square**; the chosen **cross** is simply an outlier. By selecting embeddings to measure from, the risk of outliers skewing the distance measurement, and thus the final classification, increases. This risk can be mitigated through the use of class prototypes – generalised embeddings generated from the median embedding for all examples of each class. By making use of prototypes, the effect of outliers during classification is reduced.

Figure 6.8 shows the same example two class latent space as before, however it now also displays the generated class prototypes  $P_x$  and  $P_s$  respectively. If the distance measurement

is performed using the prototypes, the triangle is now classified as a **square**, which is more likely given the construction of the global space.

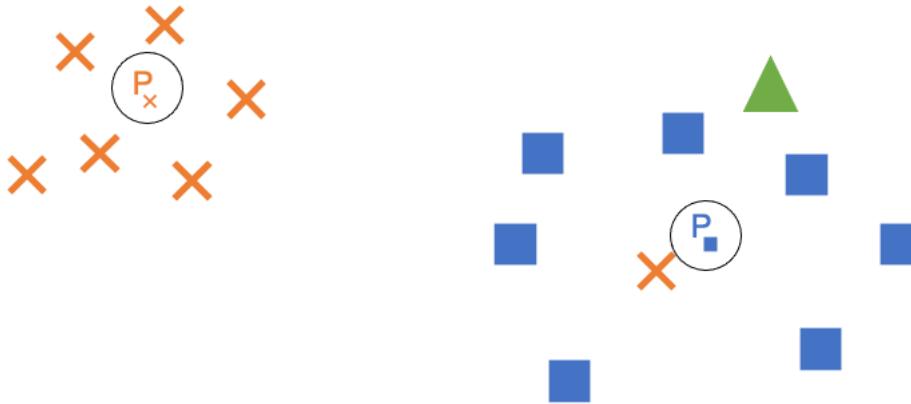


Figure 6.8 An example latent space with two classes (**cross** and **square**) alongside a triangle which represents the embedding location of an unclassified inference image. The two class prototypes used to classify the triangle,  $P_x$  and  $P_s$  respectively, are circled.

This method is not without its limitations either however. If all examples of the **cross** class formed a circle of radius 1 around the origin and all **square** examples formed a circle of radius 2, in both cases the class prototypes would be formed at the origin resulting in equal distance measurements. Although the chances of this are small, and these formations could be avoided using different hyperparameters or model architecture, it is important to be aware that this could happen when utilising prototypes.

### 6.2.6 Top-N Accuracy

In coarse-grained computer vision tasks such as image classification, the effectiveness of a model is evaluated using, among other metrics, an accuracy score. Given an image in the test dataset, the model's class prediction for the image, defined as the class which has the highest probability assigned to it, is compared against the ground truth class label. If the predicted and ground truth classes are the same, the model is correct and is operating as intended. Performing this process iteratively over all images in the test dataset provides an accuracy score, often written as a percentage denoting how many test images the model classified correctly.

Due to the low inter-class differences between the classes in a fine-grained dataset however, the same model may struggle to consistently assign the highest probability to an image's ground truth. Using the coarse-grained definition of accuracy, this model may now perform poorly. However for certain tasks, such as most likely catalogue matching, the model may be considered effective simply if it is able to reduce the range of classification possibilities.

As such, the top- $N$  accuracy metric is often utilised for fine-grained classification tasks [245, 275, 276]. Rather than only outputting the class with the highest probability, the model instead takes the  $N$  highest probabilities, outputting its prediction as a list of possible values. If the ground truth class is contained within the list, the model is considered to be correct.

For example, utilising top-10 accuracy the model would output the 10 highest probabilities for an image and would be considered correct if the ground truth label was within this list. Utilising top-1 accuracy would force the model to output a single prediction, which would be the same as utilising the standard accuracy definition.

As the task of most likely catalogue matching is a fine-grained problem, developed SNNs are evaluated using the top- $N$  accuracy metric. Furthermore, as the goal of this work is to produce a system which aids researchers through the task of catalogue matching rather than fully replace them, it is beneficial for the SNNs to produce a list of predictions as this will greatly reduce the number of individuals the researcher needs to examine in order to be confident of a match.

## 6.3 Siamese Neural Network Development

The following sections detail the creation of the SNN-based catalogue matcher. Figure 6.9 highlights where in the system pipeline this model will be placed.

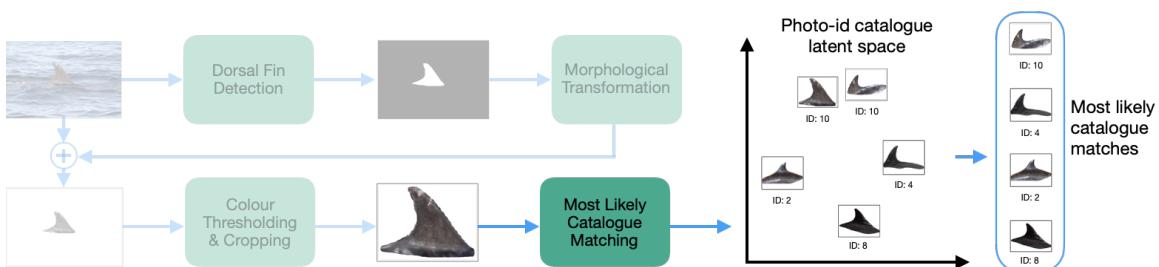


Figure 6.9 The high level pipeline overview, shown in Figure 1.1, with the SNN component highlighted. It is this part of the pipeline that will be discussed in the following sections.

Two backbone architectures were tested during SNN development. The first of these was the architecture defined in Vetrova *et al.* [265], hereafter denoted as VarvaraNet. This was utilised to examine whether a network which is proven capable at species identification is also able to perform well for the task of individual identification. The second was a custom architecture consisting of a Convolutional layer, a Dropout layer, a PReLU layer, and a MaxPool layer (stride = 2). This network was utilised to examine whether a more basic backbone would be capable of performing well given the fine-grained, few-shot nature of the task. This architecture is hereafter denoted as EmbeddingNet. To aid development, this work made use of Adam Bielski's PyTorch implementation of SNNs<sup>1</sup>.

### 6.3.1 Hyperparameter Tuning Via Bayesian Optimisation

Like all models, SNNs have multiple hyperparameters which must be tuned. As such, work began to select which hyperparameters should be tuned and how. Since developing the Mask R-CNN fin detector, discussed in Chapter 4, the area of hyperparameter optimisation has advanced considerably. Multiple frameworks now exist which take a Bayesian approach to finding the optimal hyperparameter values. Unlike optimisation through a Grid Search whereby all combinations of user-defined hyperparameter values are evaluated (see Section 4.4.3 for an example of this), with Bayesian Optimisation the user only needs to define the upper and lower bounds for each hyperparameter. The search space is then explored using a probabilistic methodology, locating the optimal set of hyperparameters within the ranges provided. This speeds up the optimisation process as values unlikely to yield promising results are ignored. As such, a larger number of hyperparameters can be optimised when compared to a Grid Search, and this can be achieved in a smaller amount of time.

The Optuna framework [277] was utilised to facilitate hyperparameter tuning through Bayesian optimisation. Whilst Optuna allows users to make use of custom optimisation algorithms, this work specifically made use of the built-in Tree-structured Parzen Estimator (TPE) algorithm. Optuna performs optimisation iteratively. This means that, for each iteration and for each hyperparameter, TPE fits one Gaussian Mixture Model to the set of hyperparameter values,  $x$ , associated with the current optimal values,  $l(x)$ , and another to the remaining hyperparameter values,  $g(x)$ . Optimal values for each iteration are selected by maximising the ratio  $l(x)/g(x)$ , with the final trial producing the current optimal hyperparameter values. For a more in-depth discussion of TPE, see Bergstra *et al.* [278].

Through Optuna, TPE was utilised to set the learning rate to a `log uniform` value between  $1 \times 10^{-6}$  and  $1 \times 10^{-3}$ , for use with either the SGD or Adam optimiser. Weight

---

<sup>1</sup>*Siamese and triplet learning with online pair/triplet mining* repository by Adam Bielski: [github.com/adambielski/siamese-triplet](https://github.com/adambielski/siamese-triplet)

decay was set to a `log uniform` value between  $1 \times 10^{-6}$  and  $1 \times 10^{-1}$ . Step size was set to an `int` value between 5 and 10, with the  $\gamma$  value for this set to a `log uniform` between  $1 \times 10^{-3}$  and  $1 \times 10^{-1}$ . The margin,  $m$ , defined in the Triplet Ranking Loss (see Equation 6.2) was set to a `log uniform` value between 0.1 and 1.0. The final embedding layer was tuned to produce an `int` value between 16 and 128.

Optimisation of the number of network blocks was also examined. For VarvaraNet a block consisted of a Convolutional layer, a MaxPool layer (`stride = 2`), a ReLU layer, and a Dropout layer. For EmbeddingNet a block consisted of a Convolutional layer, a Dropout layer, a PReLU layer, and a MaxPool layer (`stride = 2`). During searching, the number of blocks was treated as a hyperparameter optimising for an `int` between 1 and 5 blocks. The size of the initial Convolutional layer was also tuned, searching for an optimal `int` value between 16 and 100. Subsequent layers were double the size of the previous. Dropout was set to search for a `log uniform` value between 0.1 and 0.7. The kernel size of the initial Convolutional layer was set to a `categorical` value of either 5, 6, 7, or 8 with subsequent layers set according to  $\max(1, k - 2)$  where  $k$  is the kernel size of the previous Convolutional layer.

### 6.3.2 Data Augmentation Strategy

The use of data augmentation was also examined. The decision was made to reduce the variety of augmentations performed compared to Mask R-CNN development, as discussed in Section 4.3.3. At this stage in the pipeline the data seen by the SNN has been post-processed, thus it would not be realistic to utilise an aggressive data augmentation strategy over the NDD AU SMRU dataset. Further, an aggressive strategy may obscure the identifying markers present on the fins too much for meaningful training to occur.

The first strategy, Colour Jitter, randomly perturbs the input images' brightness by a factor of between 0.8 and 1.2, contrast by a factor of between 0.8 and 1.2, saturation by a factor of 0.9 and 1.1, and hue by a factor of -0.1 and 0.1. The second, Perspective Shift, randomly distorts the input image's perspective by a factor of 0.5. The third, Greyscale, converted the three-channel RGB input image into a single-channel greyscale image. Tests examining combinations of these strategies were also examined, such as augmenting with both Colour Jitter and Perspective Shift. Note that Greyscale cannot be combined with Colour Jitter due to the reduction in colour channels required.

## 6.4 Siamese Neural Network Model Selection

Models with both VarvaraNet and EmbeddingNet architectures were trained for the task of most likely catalogue matching using the NDD AU SMRU dataset and the data augmentation strategies defined in Section 6.3.2. Hyperparameter optimisation was performed for each architecture-augmentation combination, as each architecture and augmentation strategy may influence the optimal hyperparameters for the model.

To perform hyperparameter tuning through Bayesian optimisation, both a train and validation set is required, with the former utilised to train the selected architecture using the current iteration’s selected hyperparameters and the latter utilised to evaluate how well this model performs against unseen data – acting like the test set for each iteration. As such the dataset was first divided randomly using an 80-20 train-test split. The train set was then divided randomly further for optimisation, with 30% of the train set held for validation, resulting in an overall 56-24-20 train-validation-test split.

Whilst the train and validation splits may feel unnatural (a 56-24 split is not common in the literature), splitting in this way ensures that a high variety of Semi-Hard triplets can be generated at all points in the training process. To further aid this, before splitting the dataset was filtered to remove any classes which contained fewer than six example images. Performing this step has the added benefit of allowing some individuals to be held back to examine the SNN’s ability to flag those it has not been trained to recognise, as these can be treated as uncatalogued individuals.

Once the final optimal hyperparameters had been located using Bayesian optimisation, the train and validation sets were recombined and used to train the selected architecture from scratch using the located hyperparameters, alongside the selected data augmentation strategy. Once trained, the model was then evaluated using the test set.

Table 6.1 shows the results of training an SNN to perform most likely catalogue matching on the NDD AU SMRU dataset. Each model trained is evaluated using top-1, top-5, and top-10 accuracies. As can be seen, the best performing model is a VarvaraNet trained without the use of any data augmentation. This model achieves the highest test set accuracy at all evaluated metric thresholds, achieving 40.85% top-1, 68.90% top-5, and 83.13% top-10 accuracies. These results provide evidence that SNNs are capable of fine-grained, few-shot individual level identification. If the model was deployed into production and utilised by cetacean researchers, then these levels of accuracy would vastly reduce the search space required to perform most likely catalogue matching.

On the whole, it is the case that models using an EmbeddingNet backbone perform worse than those using a VarvaraNet backbone, even when utilising the same data augmentation

Model Backbone	Data Augmentation Strategy	Accuracy (%)		
		Top-1	Top-5	Top-10
EmbeddingNet	Greyscale	30.69	58.13	75.00
VarvaraNet	Greyscale	42.07	62.20	74.80
EmbeddingNet	Greyscale & Perspective Shift	35.57	51.02	68.30
VarvaraNet	Greyscale & Perspective Shift	38.82	62.60	77.64
EmbeddingNet	None	39.63	66.46	79.06
<b>VarvaraNet</b>	<b>None</b>	<b>40.85</b>	<b>68.90</b>	<b>83.13</b>
EmbeddingNet	Colour Jitter	28.05	50.00	66.67
VarvaraNet	Colour Jitter	38.82	61.18	76.83
EmbeddingNet	Perspective Shift	22.36	47.15	65.24
VarvaraNet	Perspective Shift	23.17	51.22	73.58
EmbeddingNet	Perspective Shift & Colour Jitter	30.69	53.25	68.49
VarvaraNet	Perspective Shift & Colour Jitter	40.04	61.18	78.46

Table 6.1 Results of SNN training for the task of most likely catalogue matching on the NDD AU SMRU dataset. The model chosen for use is highlighted in bold.

strategies. Furthermore it can be seen that, in general, the data augmentation strategy chosen for training has little effect on final model performance against the test set as evidenced by the lack of variation in accuracy metrics between models when trained using different strategies. This is especially interesting with regards to the greyscale augmentation as whilst there has been a drop in accuracy when compared to the best performing model this is only slight. However, this does suggest that a reduction in colour channel leads to some information loss. The fact that the best performing model is one which makes use of no data augmentations may suggest that even strategies which only perturb the input slightly still have a negative effect on the model’s ability to extract identifying information.

#### 6.4.1 An Evaluation of Optimal Model Hyperparameters

The optimal SNN hyperparameters chosen through Bayesian optimisation for each architecture-augmentation training run can be seen in Table 6.2. Of the models trained, 83% of them use Adam [80] as an optimiser, including the best performing model (VarvaraNet without data augmentation). This aligns with the belief within deep learning research that Adam will often provide optimal model training [258].

Model Backbone	Data Augmentation Strategy	Network Blocks	Initial Convolutional Layer Size	Initial Convolutional Layer Kernel Size	Dropout	Learning Rate	Optimiser	Weight Decay	Step Size	$\gamma$	Embedding Size	Triplet Ranking Loss Margin
EmbeddingNet	Greyscale	5	37	8	0.677	$2.578 \times 10^{-6}$	Adam	$5.008 \times 10^{-3}$	6	0.048	120	0.551
VarvaraNet	Greyscale	2	60	5	0.358	$1.031 \times 10^{-4}$	Adam	$3.680 \times 10^{-5}$	5	0.036	17	0.863
EmbeddingNet	Greyscale & Perspective Shift	5	26	7	0.261	$1.214 \times 10^{-4}$	Adam	$8.364 \times 10^{-3}$	10	0.081	65	0.684
VarvaraNet	Greyscale & Perspective Shift	5	62	5	0.269	$1.817 \times 10^{-6}$	Adam	$1.954 \times 10^{-4}$	5	0.030	91	0.860
EmbeddingNet	None	3	29	7	0.242	$2.835 \times 10^{-5}$	Adam	$9.106 \times 10^{-3}$	8	0.001	69	0.758
VarvaraNet	<b>None</b>	<b>2</b>	<b>59</b>	<b>6</b>	<b>0.169</b>	<b><math>7.253 \times 10^{-6}</math></b>	<b>Adam</b>	<b><math>4.338 \times 10^{-2}</math></b>	<b>10</b>	<b>0.012</b>	<b>106</b>	<b>0.796</b>
EmbeddingNet	Colour Jitter	5	44	6	0.197	$1.876 \times 10^{-5}$	Adam	$3.567 \times 10^{-6}$	6	0.011	40	0.436
VarvaraNet	Colour Jitter	3	38	5	0.684	$9.251 \times 10^{-4}$	SGD	$7.512 \times 10^{-4}$	5	0.004	90	0.281
EmbeddingNet	Perspective Shift	5	42	6	0.120	$1.653 \times 10^{-5}$	Adam	$3.256 \times 10^{-3}$	5	0.014	60	0.273
VarvaraNet	Perspective Shift	1	45	6	0.447	$2.890 \times 10^{-4}$	Adam	$4.458 \times 10^{-4}$	6	0.004	24	0.635
EmbeddingNet	Colour Jitter & Perspective Shift	3	28	6	0.559	$1.348 \times 10^{-6}$	Adam	$1.608 \times 10^{-5}$	5	0.073	51	0.458
VarvaraNet	Colour Jitter & Perspective Shift	2	35	5	0.286	$4.093 \times 10^{-4}$	SGD	$5.352 \times 10^{-4}$	9	0.068	28	0.826

Table 6.2 Optimal SNN hyperparameters for each architecture-augmentation combination located using Bayesian optimisation over 100 iterations. Results given to 3 decimal places where applicable. The model chosen for use is highlighted in bold.

Furthermore, in general the best results are achieved when utilising a low probability of dropout. This suggests that the models are not overfitting even with relatively small amounts of training data, which may be due to the low inter-class, high intra-class differences present in the dataset, as seen in Figure 5.11.

Interestingly it is also the case that all hyperparameter optimisation runs, regardless of architecture or data augmentation strategy, settle on a Triplet Ranking Loss margin above 0.2, the value commonly used as a default for this hyperparameter thanks to work by Schroff *et al.* [38]. There is a high deviation in margin value between the models, which may suggest that the use of 0.2 in all cases by default will not lead to optimal results.

One important takeaway from the use of Bayesian optimisation techniques is that for some hyperparameters that require a continuous input, such as learning rate or weight decay, the optimal value may be one which would likely not be selected by a human when performing a Grid Search. This highlights the effectiveness of Bayesian optimisation over a Grid Search as, whilst the difference between a learning rate of  $2 \times 10^{-6}$  selected during a Grid Search and  $2.578 \times 10^{-6}$  selected by Bayesian optimisation, may only be a few percentage points increase in test set accuracy, this still results in an overall more generalisable model.

#### 6.4.2 NDD AU SMRU Uncatalogued Individual Thresholding

Further evaluation was performed to examine the best performing model's ability to flag uncatalogued individuals to the user. Before training, some classes from the NDD AU SMRU dataset were removed. As these are classes which have not yet been seen by the model, they can be utilised as if they were uncatalogued individuals.

### Thresholding via Class Prototypes

When an image is passed through the SNN its most likely catalogue matches are generated using the Euclidean distance between it and the available class prototypes. By clustering individuals together in the latent space based on embedding similarity and comparing new images using Euclidean distance to the generated class prototypes, the system can flag potentially previously uncatalogued individuals to the researcher - assuming the model has learnt to generate well defined class clusters.

New individuals who have entered the catalogue's geographical survey area (e.g. through migration or birth) will thus have numeric representations which plot them into their own distinct location in the latent space, resulting in large Euclidean distances between that location and the existing photo-id catalogue prototypes. In order to flag potentially uncatalogued individuals, the use of a threshold was required. If the Euclidean distances between the input image and all class prototypes are above this threshold, then the input image is flagged to the user as a potentially uncatalogued individual.

To decide on a threshold value for the NDD AU SMRU dataset, each image not utilised for SNN training was processed by the model and the distances between it and the class prototypes were empirically examined, in line with other works in this area [279]. Through this, a threshold value of 4.0 was determined – if the distance to the closest class prototype was above this value, then it is likely that the individual may be uncatalogued. An example of this can be seen in Figure 6.10 (Top Right), which shows an example image of individual 3, a class which was not included in the training of the model. As can be seen, the distance between the image's embedding and the closest class prototype is above the threshold, and thus the image has been flagged for manual review.

The use of class prototypes proved especially useful for determining noise. Through experimentation it was found that an image could be flagged as belonging to the noise class simply if the distance between the input image's embedding and the noise class prototype was the smallest, as seen in Figure 6.11 (Top Right). This provides evidence to suggest that the model is capable of clustering examples of noise together well in the latent space, especially impressive as this class contains the highest intra-class variation as a result of containing all images of erroneous non-fin detections.

### Thresholding via K-Nearest Neighbours

During experimentation it was found that when only utilising class prototypes, the model struggled to highlight uncatalogued individuals if some class clusters were spread over a large area in the latent space. To combat this, the use of K-Nearest Neighbours (KNN) to flag

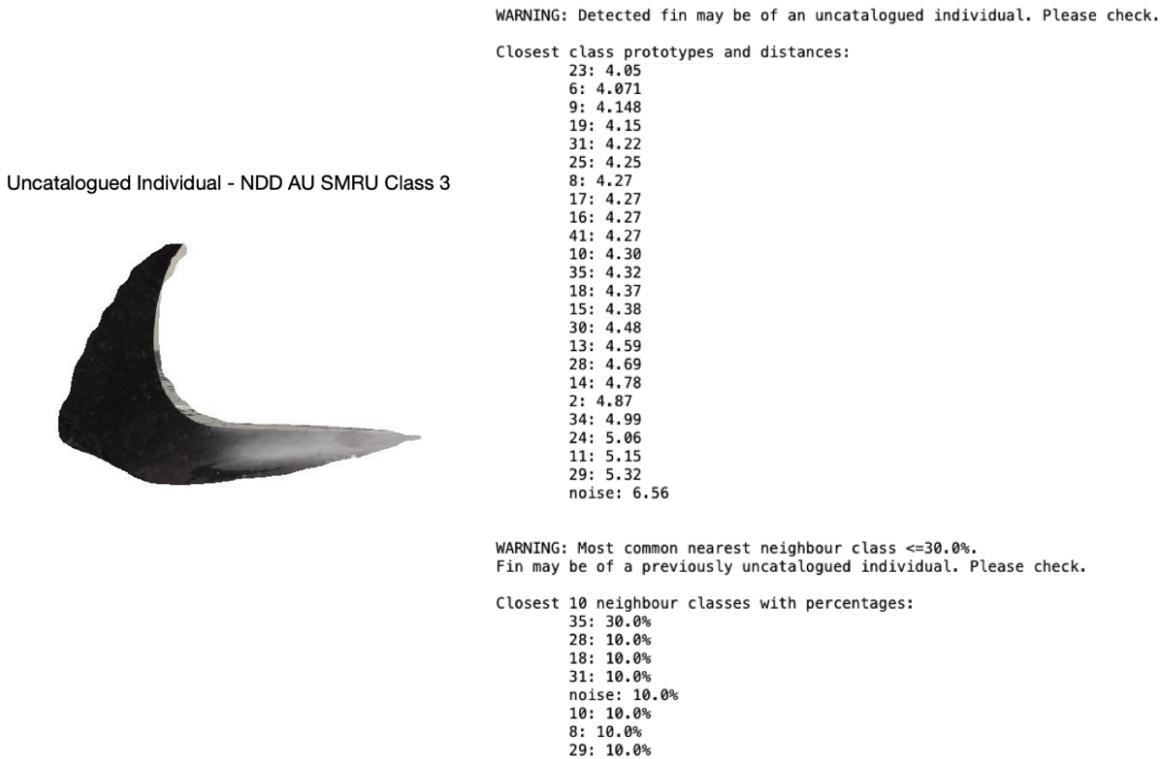


Figure 6.10 Example uncatalogued individual thresholding. Left: the input image seen by the model, of class 3. The model has not been trained using this class. Top Right: the resultant Euclidean distances between the input image's embedding and the existing class prototypes. As the distance to the closest prototype is above the threshold, the fin has been flagged for manual review. Bottom Right: uncertainty scores generated using K-Nearest Neighbours clustering. The model's most confident class is below the threshold, and thus the fin has been flagged for manual review.

potentially uncatalogued individuals was examined. KNN works by comparing each input image's embedding to the  $K$  closest embeddings in the latent space, where  $K$  is some user defined value.

If, for example,  $K = 8$  then the closest eight embeddings to that of the input image are examined. If seven embeddings are of class 1 and one embedding is of class 2, then we have an 88% certainty that the input image is an example of class 2. However, if the image is embedded and three of the closest eight embeddings are of class 1, two are class 2, one of class 3, one of class 4, and one of class 10, then the uncertainty score for any classification of the input image would be high. This could then be flagged to the user. As such, the use of an uncertainty score allows for greater interpretability when compared to a similarity score based on distances between points in the latent space, especially for those without an understanding of the underlying concepts behind SNNs like cetacean researchers.

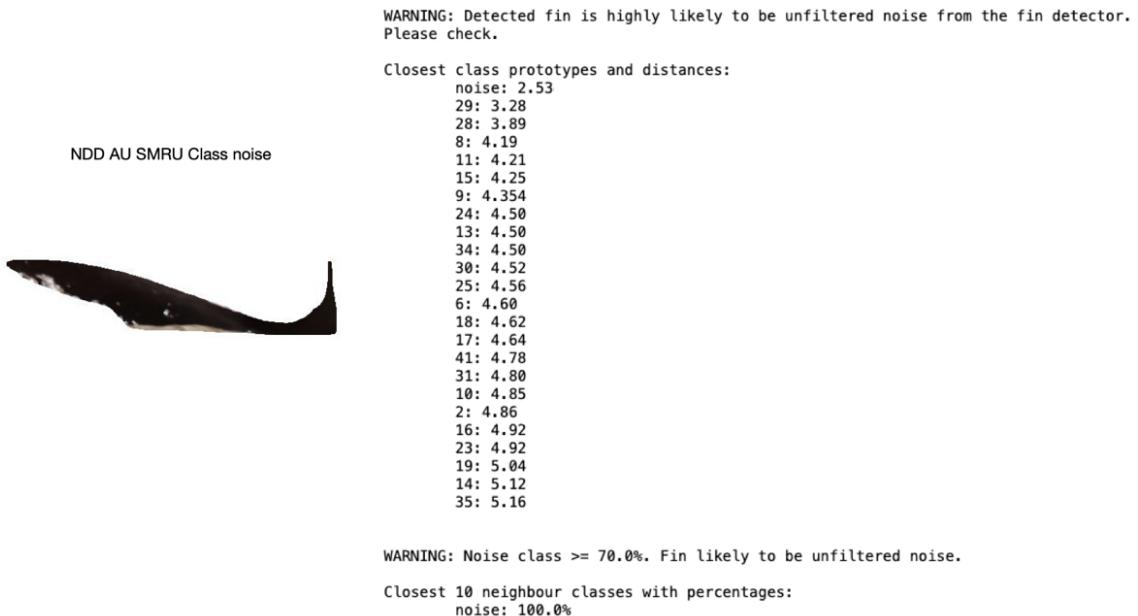


Figure 6.11 Example noise thresholding. Left: the input image seen by the model, of class noise. Top Right: the resultant Euclidean distances between the input image’s embedding and the existing class prototypes. As the closest prototype is `noise`, the input has been flagged. Bottom Right: uncertainty scores generated using K-Nearest Neighbours clustering. The model’s `noise` confidence is above the threshold, and thus the input has been flagged.

During experimentation, it was found that setting  $K = 10$  alongside an uncertainty threshold of 30% allowed for the best chance of flagging uncatalogued individuals. If no class reached above 30% certainty, this was flagged to the user as seen in Figure 6.10 (Bottom Right). A noise threshold of 70% was also enforced, whereby if the `noise` class confidence was above 70% this would be flagged as seen in Figure 6.11 (Bottom Right).

After experimentation with both methodologies, it was decided that utilising both class prototypes and KNN for flagging potentially uncatalogued individuals to the user would be best given the ability of the methodologies to complement one another. Whilst KNN allows for the production of an easily understandable metric for those without knowledge of the underlying processes of SNNs, the use of class prototypes provides a more nuanced, albeit latent space specific, metric. For all utilised uncatalogued images, 67.86% were flagged using the above methodologies and threshold values. Both methods also allow for extra classes to be added to the latent space without the need for model fine-tuning or retraining. This means any uncatalogued individuals identified by the model can be added to the latent space to aid future identifications, alongside addition to the biological catalogue.

### 6.4.3 Limitations of the Model

As discussed in Section 6.4, the best performing SNN trained for most likely catalogue matching using the NDD AU SMRU dataset achieves 40.9% top-1, 68.9% top-5, and 83.1% top-10 accuracies. Whilst these results are impressive given the fine-grained, few-shot nature of the task, it is important to also highlight the limitations of the approach.

One limitation is the need to re-train the SNN for each new photo-id catalogue. As a result, initial manual curation must be performed before the proposed methodology can be applied. Evidence outlined in Section 6.4.2 suggests that, once trained, new classes can be added to the SNN with ease when a new individual is added to an existing catalogue – however it is not yet clear whether this is the case in perpetuity or whether eventually the model is no longer able to generate meaningful clusters. The problem of drift, where model performance begins to degrade over time as the distribution of the data it receives naturally changes, is common with models once in production. It may be the case that as more individuals are added to the photo-id catalogue, and thus classes are added to the SNN’s latent space, the need for a full model retrain increases. Drift detection for computer vision models is an open area of research [280–283], and it may be the case that proposed works in this area, or in the area of continual learning, are able to detect when the SNN should be retrained on the updated catalogue. It may also be the case that uncatalogued individual thresholding is dataset dependent. If so, model retraining may also necessitate the location of a new threshold value.

Issues may also arise if existing individuals in the catalogue change significantly, both from natural or anthropogenic interactions. If an individual’s prominent markings change drastically due to an event such as a boat strike, then this may impact the SNN’s ability to perform catalogue matching. It is likely that example images of the individual after their markings have changed would result in the model believing the individual to be uncatalogued. This problem is one of the main driving factors behind the decision to keep a human involved in the cataloguing process.

Finally, it is not currently clear how well the approach taken in this chapter to most likely catalogue matching would perform with other cetacean species such as whales or porpoises, or with other body parts such as flukes rather than dorsal fins. Further studies with other species and body parts should be explored, assuming access to these catalogues can be obtained.

## 6.5 Per-Side Identification

When performing catalogue matching manually, cetacean researchers will often identify each side of an individual independently rather than as a whole if both the left and right side of the dorsal fin have been captured. During initial testing the decision was made to train the model using only a single class for each individual using example images of both the left and right side if available. This decision was made for two reasons. First, the number of catalogue images for some individuals was small. If a single class was utilised for both the left and right side, this increased the number of example images available for each class. This was deemed important for initial model development. Second, by combining both the left and right side images into a single class this further tests the model's ability to handle intra-class variation.

Once a baseline best model had been located, experimentation was undertaken to examine whether automated photo-id matching should, like its manual counterpart, split individual classes on a per-side basis. Where possible, each class in the NDD AU SMRU dataset was divided in two, with each new class containing only the left or right side examples of the individual. This increased the number of classes in the dataset from 24 to 47 (one original class only contained example images of a single side). The number of example images per side was slightly skewed, with a median of 11.5 examples for left classes and 13 for right classes.

Using this split dataset model training was performed, utilising the same architectures, data augmentations, and hyperparameter optimisation strategy as outlined in Section 6.3. Greyscaling was not performed here due to the results outlined in Section 6.4 highlighting that information was lost when using this data augmentation. Resultant models were again evaluated using top-1, top-5, and top-10 accuracies, the results of which can be seen in Table 6.3.

A comparison between the best performing combined-side and per-side models can be seen in Table 6.4. Only a 5.30% drop in top-10 accuracy is observed between the combined-side and per-side models, reduced to a 2.16% drop when utilising top-5 accuracy. When evaluating with top-1 accuracy however, the per-side model outperforms the combined-side model by 3.08%. Whilst model performance has dropped using some metrics when compared to training using the combined-side dataset, it is important to note that training using the per-side dataset nearly doubles the number of possible classes known by the model. As such, a drop in top- $N$  accuracy should be expected simply due to the increase in the number of possible classes. The fact that this drop is so small at top-10 and top-5 levels, and performance actually improves when measuring using top-1 accuracy, suggests that utilising

<b>Model Backbone</b>	<b>Data Augmentation Strategy</b>	<b>Accuracy (%)</b>		
		<b>Top-1</b>	<b>Top-5</b>	<b>Top-10</b>
<b>EmbeddingNet</b>	<b>None</b>	<b>43.93</b>	<b>66.74</b>	<b>77.83</b>
VarvaraNet	None	25.94	54.18	66.94
EmbeddingNet	Colour Jitter	42.68	65.06	76.36
VarvaraNet	Colour Jitter	31.17	51.46	64.22
EmbeddingNet	Perspective Shift	37.03	55.44	67.36
VarvaraNet	Perspective Shift	32.01	52.30	64.02
EmbeddingNet	Perspective Shift & Colour Jitter	17.36	52.51	67.36
VarvaraNet	Perspective Shift & Colour Jitter	28.45	56.69	71.96

Table 6.3 Results of SNN training for the task of most likely catalogue matching on the NDD AU SMRU dataset after per-side splitting (47 classes). The best performing model is highlighted in bold.

<b>NDD AU SMRU Dataset Split</b>	<b>Model Backbone</b>	<b>Data Augmentation Strategy</b>	<b>Accuracy (%)</b>		
			<b>Top-1</b>	<b>Top-5</b>	<b>Top-10</b>
Combined Side (24 classes)	VarvaraNet	None	40.85	68.90	83.13
Per Side (47 classes)	EmbeddingNet	None	43.93	66.74	77.83

Table 6.4 Comparison between the best performing SNN models for the task of most likely catalogue matching trained on the combined-side and per-side NDD AU SMRU datasets.

a per-side approach to automated catalogue matching is better than utilising a single class for all example images of an individual.

Interestingly the best performing per-side model utilises an EmbeddingNet backbone architecture, whilst the best performing combined-side model makes use of a VarvaraNet. Both models however did not make use of a data augmentation strategy. As the EmbeddingNet architecture is less complex than the VarvaraNet, this may suggest that the reduction in intra-class variation when utilising a per-side dataset reduces the complexity of the backbone architecture required for the model to perform well on the test dataset. This is also reflected in the optimal output embedding size located during hyperparameter optimisation. Whilst the combined-side model has an output size of 106 the per-side model has an output size of 36, suggesting that less dimensionality is required to capture the information needed to perform catalogue matching. A full list of model hyperparameters for the best performing per-side model can be found in Appendix E.

Model Backbone	Data Augmentation Strategy	Accuracy (%)		
		Top-1	Top-5	Top-10
<b>EmbeddingNet</b>	<b>None</b>	<b>44.56</b>	<b>68.62</b>	<b>80.54</b>
VarvaraNet	None	27.62	58.37	73.01
EmbeddingNet	Colour Jitter	43.72	67.15	78.87
VarvaraNet	Colour Jitter	32.22	53.56	68.62
EmbeddingNet	Perspective Shift	37.45	59.00	73.44
VarvaraNet	Perspective Shift	32.64	54.40	67.16
EmbeddingNet	Perspective Shift & Colour Jitter	18.41	56.90	72.80
VarvaraNet	Perspective Shift & Colour Jitter	29.50	60.25	76.36

Table 6.5 Results of SNN training for the task of most likely catalogue matching on the NDD AU SMRU dataset after per-side splitting, utilising the modified Top- $N$  accuracy measurement to only take into account the individual classification. The best performing model is highlighted in bold.

### 6.5.1 Individual Only Top-N Accuracy

When performing catalogue matching on a per-side basis, the final classification is still only recorded at an individual level. However, evaluation discussed previously in Section 6.5 does not take this into account.

When the model outputs a list of possible classifications, it may be the case that whilst the correct individual is in the list, the incorrect side is given. For example, with an input image of class 10\_R the model may output a list of likely matches which contains 10\_L but not 10\_R. Using the traditional implementation of top- $N$  accuracy, the model would be deemed incorrect even though the correct identification has been given.

To rectify this, a modification was made to the top- $N$  accuracy metric such that it ignored the side classification. For example, both 10\_L and 10\_R would evaluate to 10. Now, if the model provides a correct individual classification but an incorrect side classification, it is deemed to be correct during evaluation. Note that this change was only implemented at test time, so as to not hinder embedding generation through the training process.

Results of evaluating the models trained in Section 6.5 using this modified top- $N$  accuracy score can be seen in Table 6.5. On average, this truer-to-life modification improves a model's top-1 accuracy by 0.94%, top-5 accuracy by 3.01%, and top-10 accuracy by 4.34%.

Table 6.6 shows a comparison between the best performing combined-side and per-side models evaluated against the modified top- $N$  accuracy. Using this metric, an increase in top-1, top-5, and top-10 accuracies are observed for the per-side model when compared to those reported in Table 6.4, which reports non-modified accuracies. Whilst these accuracies

are lower than when the model is trained on the combined-side data, given the difference in number of classes between the datasets evidence suggests that using a per-side dataset for training improves the overall performance of a most likely catalogue matching model.

NDD AU SMRU Dataset Split	Model Backbone	Data Augmentation Strategy	Accuracy (%)		
			Individual Classification Only	Top-1	Top-5
Combined Side (24 classes)	VarvaraNet	None	40.85	68.90	83.13
Per Side (47 classes)	EmbeddingNet	None	44.56	68.62	80.54

Table 6.6 Comparison between the best performing SNN models for the task of most likely catalogue matching trained on the combined-side and per-side NDD AU SMRU datasets, evaluated against the modified top- $N$  accuracy metric.

## 6.6 Summary

This chapter discusses the creation of a Siamese Neural Network (SNN) for the task of individual cetacean most likely catalogue matching. Trained using data triplets consisting of automatically post-processed fieldwork imagery, the model is able to generate image embeddings in such a way that, when plotted into a multi-dimensional latent space, embeddings of the same individual cetacean are clustered together. By utilising Euclidean distance measurements between generated class prototypes as well as the K-Nearest Neighbours algorithm (KNN), the model is able to both produce as output a list of likely catalogue matches and flag when it receives as input a potentially previously uncatalogued individual. The use of this model would vastly decrease the time required for cetacean researchers to match against an existing photo-id catalogue, acting as a ranking mechanism to filter out unlikely matches – especially when placed at the end of a fully automatic pipeline.

The model is evaluated using the fine-grained few-shot NDD AU SMRU dataset, the creation of which is outlined in Section 5.6.1. Initial results show the model is capable of achieving 40.85% top-1, 68.90% top-5, and 83.13% top-10 accuracies when trained using classes which consist of example images of both sides of an individual’s dorsal fin. Further work detailed in Section 6.5 provides evidence to suggest that dividing individuals into two classes, one for each side of their fin, improves overall model performance. When this split is performed the number of classes known by the model is doubled however top-1 accuracy improves by 3.08%, with top-5 and top-10 accuracies only falling slightly by 2.16% and 5.30% respectively. When modifying the top- $N$  accuracy metric to account only for

the individual rather than the side, discussed in greater detail in Section 6.5.1, the model’s performance increases by 0.94%, 3.01%, and 4.34% for top-1, top-5, and top-10 metrics respectively.

Evaluating against the requirements outlined in Section 6.1, it is clear the model performs all functionality as intended. The model is able to operate to a high degree of accuracy utilising all available information provided by the dorsal fin, rather than utilising a single identifying marking such as the trailing edge. Embeddings are generated in such a way as to cluster extreme fine-grained dataset class examples close together, with each cluster located in its own distinct location in the latent space. Generated embeddings are not hindered by misclassified noise which may still be present after post-processing. Finally, the model is also capable of flagging potentially previously uncatalogued individuals to cetacean researchers through the use of Euclidean distance similarity and uncertainty scores generated using KNN. Whilst this research complements and expands upon work in the areas of animal re-identification [30, 265, 284], limitations to the approach do exist as outlined in Section 6.4.3.

In Chapter 7, the robustness of SNNs for the task of automatic most likely catalogue matching is evaluated. The effect of combining the NDD and AU SMRU data on model performance is explored, as well as the effect of background noise on embedding generation – justifying the use of instance segmentation over bounding box detection before matching. Finally, the generalisation of the approach is examined through the use of a second, smaller, real life photo-id catalogue.



# Chapter 7

## Evaluating Siamese Neural Network Catalogue Matching Robustness

In this chapter the robustness of Siamese Neural Networks (SNNs) for the task of automatic most likely catalogue matching is explored. Beginning by examining the effects of NDD AU SMRU dataset variation on model performance, the generalisability of the approach is then examined through the use of a second photo-id catalogue.

### 7.1 Effect of AU SMRU Data on Model Performance

As outlined in Section 5.6.1, additional photo-id data for 23 individuals were provided by the Universities of Aberdeen and St Andrews upon completion of fieldwork in the Coquet to St. Mary's Marine Conservation Zone. Whilst this data was primarily utilised to better understand the home range of resident cetaceans, it was also used to provide additional training data for automatic most likely catalogue matching, creating the NDD AU SMRU dataset and providing a larger number of class examples for initial SNN feasibility studies.

Work undertaken throughout Chapter 6 confirmed the ability of SNNs to perform most likely catalogue matching on the combined NDD AU SMRU dataset. To understand the effect of additional AU SMRU training data on model performance, two models were generated. First, a model was trained only on the data collected during the fieldwork season in Northumberland, UK, contained within the Segmented NDD20 dataset as discussed in Section 5.6.

As some classes within the Segmented NDD20 dataset are too small for semi-hard triplet mining to occur without extremely heavy data augmentation, only 17 classes in Segmented NDD20 were utilised here (hereafter denoted as Segmented NDD20 17). This is in contrast

to the 24 classes utilised when training using the full NDD AU SMRU dataset, as discussed in Section 6.4. Next a second model was trained using the NDD AU SMRU dataset, limited to just the classes present in Segmented NDD20 17 (hereafter denoted as NDD AU SMRU 17). If both models perform equally well on the test set, it can be deduced that the inclusion of the AU SMRU data during training has not aided model generalisability.

<b>Dataset</b>	<b>Model Backbone</b>	<b>Data Augmentation Strategy</b>	<b>Accuracy (%)</b>		
			<b>Top-1</b>	<b>Top-5</b>	<b>Top-10</b>
Segmented NDD20 17	VarvaraNet	None	42.26	60.44	78.13
NDD AU SMRU 17	VarvaraNet	None	42.75	69.78	88.45

Table 7.1 Comparison between two SNNs trained for most likely catalogue matching, one using the Segmented NDD20 17 dataset, and the other using the NDD AU SMRU 17 dataset. Evaluation is performed using the NDD AU SMRU 17 test set.

Table 7.1 shows the top-1, top-5, and top-10 accuracies for both models when evaluated on the NDD AU SMRU 17 test set. As can be seen, utilising the additional AU SMRU data as opposed to training solely on the Segmented NDD20 17 dataset provided a boost of 0.49%, 9.34%, and 10.30% to top-1, top-5, and top-10 accuracies respectively. These results suggest that, whilst the additional data has not provided any great increase in the model’s ability to distinguish the top-1 most likely individual, the extra training embeddings have allowed for more defined class clusters, greatly improving top-5 and top-10 performance.

This supports the hypothesis that additional training examples improve most likely catalogue matching ability, even though the data was obtained from a secondary source. As such, this provides evidence to suggest that models trained on multiple photo-id catalogues may perform better than those trained on a single study, assuming there is some individual overlap.

However, performance of the model trained solely on the Segmented NDD20 17 dataset shows that even a model trained on the relatively small amount of data collected during the Northumberland fieldwork is able to greatly reduce the search space and can be utilised on data collected in a different spatio-temporal environment than the one on which it was trained, highlighting the robustness of the approach.

## 7.2 Effect of Background on Embedding Generation

Due to the free roaming nature of cetaceans, those in the NDD AU SMRU dataset were often photographed during only a single encounter leading to data with small intra-class but high inter-class background variation. This is similar to domains such as person re-identification,



Figure 7.1 Example data used to examine the effect of retained background on most likely catalogue matching. (Left) Bounding box detection containing both a dorsal fin and background. (Centre) Corresponding dorsal fin mask. (Right) Corresponding background mask.

where images representing different classes often contain the same background information due to capture with a stationary camera. Recent work in this domain has shown that deep learning models may bias their similarity rankings based on background information [285].

To examine the effect that background removal has on downstream identification in the photo-id domain, an SNN was trained using the NDD AU SMRU dataset processed into bounding box class examples. All variables except the presence of background were kept consistent with those used when training the best performing masked SNN, a VarvaraNet without the use of any data augmentations (see Table 6.1). Data for this experiment was generated using the same Mask R-CNN detector as for previous experiments, modified to output bounding boxes rather than masks. Corresponding dorsal fin masks for each bounding boxed image were taken from the NDD AU SMRU dataset, with background masks generated by inverting the fin mask.

The bounding boxed fin, fin mask, and background mask were then embedded into the model’s latent space, and the Euclidean distances between them calculated. This analysis showed that embedding generation is likely to be influenced more by features in the background than the fin. For example, the Euclidean distance between the bounding box data in Figure 7.1 (Left) and its corresponding dorsal fin mask (Centre) is 0.36, compared to a distance of 0.30 between the bounding box and the background mask (Right) and a mean distance of 0.97 between the bounding box and generated class prototypes.

This suggests the SNN is performing likely matching based on features found in the background rather than on the dorsal fins, reflected in the increased model performance as seen in Table 7.2 – which shows the SNN trained using the bounding boxed data sees an increase of 22.94% top-1, 15.58% top-5, and 6.53% top-10 accuracies when compared to the best performing SNN trained on masked data.

By removing all background, the masked SNN is prevented from utilising environmental conditions to aid matching. This finding raises important questions regarding the performance

Model Backbone	Data Augmentation Strategy	Data Format	Accuracy (%)		
			Top-1	Top-5	Top-10
VarvaraNet	None	Bounding Boxed	63.79	84.48	89.66
VarvaraNet	None	Masked	40.85	68.90	83.13

Table 7.2 Comparison of the Top-1, Top-5, and Top-10 accuracies between training an SNN on bounding boxed or masked photo-id data. Metrics for the model trained using the masked data are for the best performing SNN as found in Table 6.1.

of photo-id aids which do not remove all background before performing matching. Section 2.5.1 provides a summary of currently available photo-id aids. Of these, four works (Bouma *et al.* [199], Lee *et al.* [200], finFindR [201], and DolFin [203]) perform dorsal fin detection and downstream individual identification, however only Bouma *et al.* and DolFin remove all background beforehand. If the photo-id catalogue utilised for evaluation of these systems has been collected over a small temporal scale, such that a temporally-robust train-test split cannot be achieved, then results obtained in this experiment suggest that performance may be artificially inflated by the retention of feature heavy background.

## 7.3 Examining Siamese Neural Network Catalogue Matching Generalisability

As outlined in Chapter 6, an automatic approach to most likely catalogue matching was developed through the use of SNNs. This approach, when tested using the NDD AU SMRU dataset developed in Section 5.6.1, yields high top-1, top-5, and top-10 accuracies. However, it is not yet clear if these results are to be expected regardless of the photo-id catalogue utilised, or if there is an underlying property inherent to the NDD AU SMRU dataset that makes it particularly susceptible to an SNN-based approach. In this chapter, automatic most likely matching is performed on a second, previously unseen, photo-id catalogue, allowing for an evaluation of the approach’s generalisability.

### 7.3.1 The SDRP Dataset

To evaluate SNN generalisability, a subset of photo-id catalogue data was obtained from the Chicago Zoological Society’s Sarasota Dolphin Research Program (SDRP). This subset consisted of 250 images of 23 individual common bottlenose dolphins captured in the waters around Naples, FL, USA [47]. Unlike the datasets collected from fieldwork in Northumberland, UK, the SDRP dataset was provided in a pre-processed form as the dataset



Figure 7.2 Example images from the SDRP dataset with filenames displayed.

had been previously utilised to compare photo-id methodologies [35]. Images were provided in a cropped format, removing a large amount of background noise and centring the dorsal fin, examples of which can be seen in Figure 7.2.

The SDRP data was provided pre-split, with 200 images (each of a unique individual) acting as the existing photo-id catalogue, with the remaining 50 serving as images captured during a given day's fieldwork. Each image in the encounter set contained a single individual, however some individuals were captured multiple times. As such, there was a 23 individual overlap between the catalogue and encounter sets.

To generate a train-test split capable of training an SNN, the catalogue set was reduced down to contain only the 23 individuals contained within the encounter set. Once filtered, both sets of images were run through the Mask R-CNN dorsal fin detector and post processed using morphological transformations, colour thresholding, and cropping – outlined in Chapter 5. No images in this dataset had been seen by the detector previously, either during training or evaluation. Once generated, a `noise` class was manually created which contained all erroneously detected mask components. However, as the detector failed to accurately detect examples of individual 19 this class was removed, resulting in a final dataset consisting of 23 classes (22 individuals plus `noise`). In cases where the detector had mistakenly detected the



Figure 7.3 Left: an image of individual 13 from the original SDRP catalogue set. Right: example masks generated for the Left image. Both masks are kept for use in training as they were deemed to be sufficiently different.

same fin twice, provided the two masks were not identical then both were kept – analogous to offline data augmentation. An example of this can be seen in Figure 7.3.

After detection and processing the resultant SDRP dataset contained a total of 123 images, significantly smaller than the NDD AU SMRU dataset which was used to evaluate the SNN-based approach previously. Retaining the split provided by the SDRP, whereby the train set was generated from the catalogue and the test set from the encounter, leads to a 35-65 train-test split, an inversion of what would be expected when training machine learning models. The class distribution for the SDRP dataset can be seen in Figure 7.4. As with the NDD AU SMRU dataset, the `noise` class is once again dominant. A colour threshold of 50% was again utilised during post-processing with no correct detections erroneously discarded, suggesting this is an acceptable general value.

These properties lead to the SDRP dataset being extremely challenging for an SNN to train on. However it is also an accurate representation of what a real life photo-id catalogue dataset would look like in the initial stages of a survey, providing an excellent test of both the robustness and generalisability of the SNN-based approach to automatic most likely catalogue matching when only small amounts of training data are available.

### 7.3.2 Evaluation Using the SDRP Dataset

Due to the small amount of data, it was not possible to create a meaningfully large and diverse validation set for the SDRP dataset. This prevented hyperparameter optimisation, and as such the decision was made to utilise the optimal hyperparameters located for the best

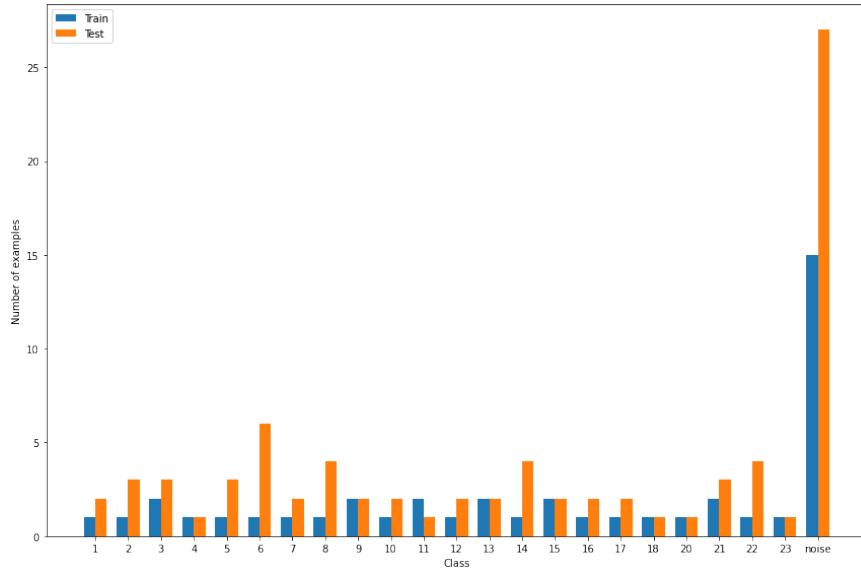


Figure 7.4 The class distribution for the SDRP dataset, split by set.

performing SNN on the NDD AU SMRU dataset, alongside the same backbone architectures and data augmentation strategies defined in Section 6.3.

The results of model training on the SDRP dataset can be seen in Table 7.3, with each model evaluated using top-1, top-5, and top-10 accuracies. Higher scores were achieved across the board on the SDRP dataset compared to the NDD AU SMRU dataset even without hyperparameter optimisation, however it is important to remember the smaller number of possible classes for the model to choose from which may inflate relative model performance.

Model Backbone	Data Augmentation Strategy	Accuracy (%)		
		Top-1	Top-5	Top-10
EmbeddingNet	None	72.50	91.25	95.00
VarvaraNet	None	72.50	85.00	92.50
<b>EmbeddingNet</b>	<b>Colour Jitter</b>	<b>81.25</b>	<b>95.00</b>	<b>97.50</b>
	Colour Jitter	70.00	91.25	96.25
EmbeddingNet	Perspective Shift	62.50	92.50	97.50
VarvaraNet	Perspective Shift	73.75	91.25	96.25
EmbeddingNet	Perspective Shift & Colour Jitter	71.25	92.50	96.25
	Perspective Shift & Colour Jitter	63.75	88.75	97.50

Table 7.3 Results of SNN training for the task of most likely catalogue matching on the SDRP dataset. The best performing model is highlighted in bold.

Unlike the NDD AU SMRU data where best results were achieved without any augmentation, here the results are more mixed. Whilst the best top-10 accuracy, 97.50%, is obtained using Colour Jitter and Perspective Shift augmentations (both together and separately), the best top-5 and top-1 accuracies were obtained using Colour Jitter only. These findings suggest that data augmentation strategy may be catalogue dependent and have a large impact on final model performance, and as such a search of possible data augmentation strategies should be performed each time an SNN model is trained using a new or updated photo-id catalogue.

Variation in backbone architecture had little effect on overall model performance. Interestingly models trained using a VarvaraNet backbone were more consistent, with a 10.00% difference between the best and worst performing model compared to an 18.75% difference between those trained using an EmbeddingNet backbone. Overall however, the best performing model was determined to be an SNN using an EmbeddingNet backbone architecture and Colour Jitter data augmentation, which achieved 81.25% top-1, 95.00% top-5, and 97.50% top-10 accuracies. This is in contrast to the best performing NDD AU SMRU model, made up of a VarvaraNet backbone architecture without any data augmentation. Using the optimal NDD AU SMRU model setup achieved 72.50% top-1, 85.00% top-5, and 92.50% top-10 accuracies on the SDRP dataset. The use of an EmbeddingNet backbone as optimal for this dataset suggests that a simpler model structure may be best when working with smaller catalogues.

### 7.3.3 SDRP Uncatalogued Individual Thresholding

Evaluation of the best performing SDRP model’s ability to flag uncatalogued individuals was undertaken. Unlike the model trained on the NDD AU SMRU dataset, a write-up of which is provided in Section 6.4, when trained on the SDRP dataset embedded images are placed closer together in the latent space. This means that the threshold values utilised for uncatalogued individual detection using prototype distance measurement and K-Nearest Neighbours (KNN) required modification to accurately output the necessary warnings.

Previous experimentation determined it was possible to flag potentially uncatalogued individuals by measuring the distance between an embedded image and its closest class prototype. For the NDD AU SMRU dataset, a minimum distance of 4.0 was required before a warning was displayed. Analysis of the distances between SDRP image embeddings and class prototypes however determined that this value was too large to be used with this latent space, as warnings were displayed for only 25.37% of uncatalogued individuals when used. Through empirical examination of the distances between each image and the class prototypes, in agreement with the related literature [279], a threshold of 0.15 was determined to be

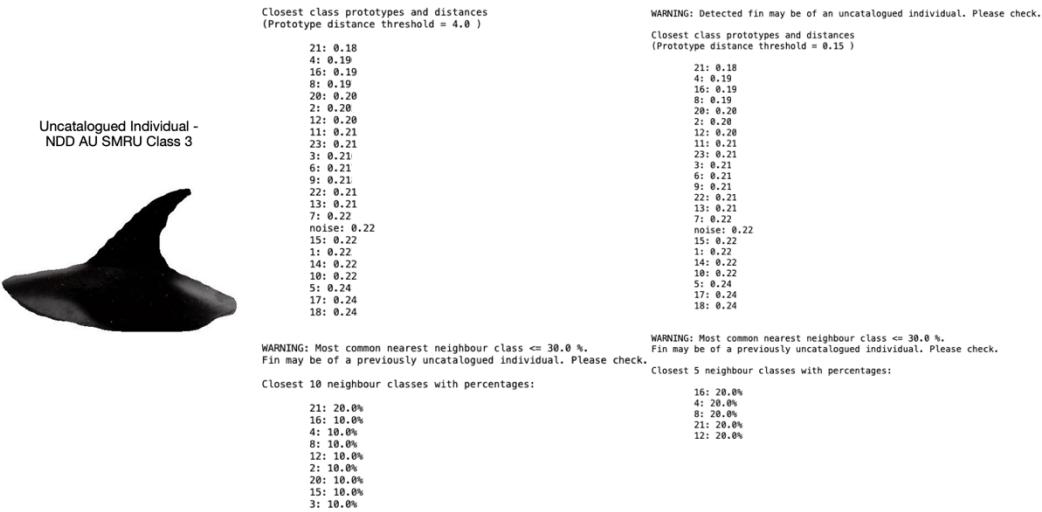


Figure 7.5 Example uncatalogued individual thresholding for the SDRP dataset using an individual not present during training. Left: the input image seen by the model, taken from the NDD AU SMRU dataset. Top Middle: The resultant Euclidean distances between the input image's embedding and the existing class prototypes, with the minimum distance threshold set to 4.0. No warning has been generated. Top Right: The resultant Euclidean distances between the input image's embedding and the existing class prototypes, with the minimum distance threshold set to 0.15. A warning has been generated. Bottom Middle: Uncertainty scores generated using K-Nearest Neighbours clustering, with  $K = 10$ . A Warning has been generated using an uncertainty threshold of  $\geq 30\%$ . Bottom Right: Uncertainty scores generated using K-Nearest Neighbours clustering, with  $K = 5$ . A Warning has been generated using an uncertainty threshold of  $\geq 30\%$ .

optimal for the SDRP dataset, being small enough to trigger a warning for uncatalogued individuals whilst still being large enough not to trigger in error.

Alongside prototype distance measurement, KNN can also be utilised for uncatalogued individual warning generation. Rather than measuring distances between embeddings as with prototype thresholding, here warnings are generated by calculating an uncertainty score based on the class labels of the  $K$  nearest embeddings. When processing the NDD AU SMRU dataset it was determined that setting  $K$  to 10 was sufficient, producing a warning when no single class made up 30% or less of the nearest class labels.

For the SDRP dataset, whilst an uncertainty of 30% was found to be sufficient for generating a warning, setting  $K$  to 10 was deemed too high. Utilising this value resulted in some catalogued individuals incorrectly producing a warning as a result of the latent space's more compact nature. Evaluation of each test image's neighbours in the latent space determined that setting  $K$  to 5 was optimal. Using the updated values, warnings were displayed for 39.06% of uncatalogued images, an increase of 13.69% over utilising the NDD AU SMRU dataset thresholds. This reduction in warning generation efficacy when compared

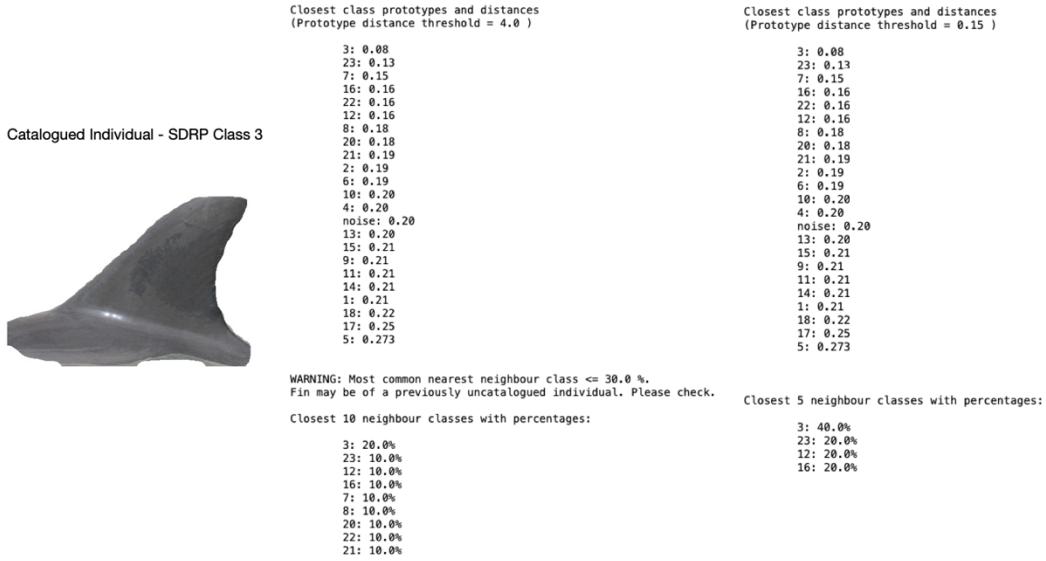


Figure 7.6 Example uncatalogued individual thresholding for the SDRP dataset using an individual present during training. Left: the input image seen by the model, taken from the SDRP dataset. Top Middle: the resultant Euclidean distances between the input image's embedding and the existing class prototypes, with the minimum distance threshold set to 4.0. Top Right: the resultant Euclidean distances between the input image's embedding and the existing class prototypes, with the minimum distance threshold set to 0.15. No warning has been generated. Bottom Middle: uncertainty scores generated using K-Nearest Neighbours clustering, with  $K = 10$ . A Warning has been generated using an uncertainty threshold of  $\geq 30\%$ . Bottom Right: uncertainty scores generated using K-Nearest Neighbours clustering, with  $K = 5$ . No warning has been generated using an uncertainty threshold of  $\geq 30\%$ .

to results in Section 6.4.2 is likely due to the small number of images in the SDRP dataset, resulting in a more compressed latent space.

Example warning generation using both prototype distance measurement and KNN for an uncatalogued individual can be seen in Figure 7.5. Here, an example from the NDD AU SMRU dataset has been processed by the SNN trained on the SDRP dataset, and so warnings should be generated both by prototype distance measurement and KNN. As can be seen, when utilising a minimum prototype distance of 4.0 no warning is generated. Changing this value to 0.15 however ensures a warning is generated. When checking using KNN, warnings are produced both when  $K$  is set to 10 and 5. Figure 7.6 shows example warning generation for an individual present during SNN training on the SDRP dataset. No warning has been generated with either minimum prototype distances, as expected. When  $K$  is set to 10 however, a warning is erroneously generated. This is prevented by setting  $K$  to 5.

Based on the experimentation described previously, it can be seen that whilst uncatalogued individual detection can be performed on the SDRP catalogue using both prototype distance

measurement and KNN as with the NDD AU SMRU catalogue, evidence suggests that threshold values for both techniques are catalogue dependent and require tuning before use. Whilst not tested here, it is thus likely that tuning is required if the model is retrained on an updated catalogue due to the change in data distribution. Further, whilst the KNN threshold values for flagging noise and uncatalogued individuals,  $\geq 70\%$  and  $\leq 30\%$  of an embedding's nearest neighbours respectively, were found to be sufficient for both catalogues examined in this work, there is no guarantee that this is the case for all catalogues globally. When adapting this work to a new catalogue, it is advised that consideration is given to all threshold values used by both prototype distance measurement and KNN for the task of uncatalogued individual detection. It may be the case that this tuning can be automated by utilising the properties of the latent space to determine sufficient thresholds based on the training data, an approach which should be explored in future.

### 7.3.4 Evaluating the NDD AU SMRU SNN Using the SDRP Dataset

As highlighted in Section 6.4.2, the SNN created using the NDD AU SMRU dataset is able to generate warnings for 67.86% of potentially previously uncatalogued individuals resident in the Northumberland survey area (see Section 3.2.2) through the use of prototype distance measurements and KNN. Using the SDRP dataset, experimentation was undertaken to examine if the Northumberland SNN was capable of generating warnings for the individuals and noise present in the SDRP dataset, without re-training or fine-tuning.

Using the threshold values for the NDD AU SMRU-trained SNN (see Section 6.4.2), warnings highlighting potentially previously uncatalogued individuals were generated for 79.25% of images containing dorsal fins of individuals present in the SDRP dataset. Further, warnings highlighting potential noise were generated for 85.19% of SDRP dataset images which were labelled as noise. These results highlight the strengths of utilising SNNs and their generated latent spaces to flag both potentially previously uncatalogued individuals and noise.

### 7.3.5 Effect of Training Set Size on Model Backbone Selection

As previously mentioned, the SDRP catalogue was provided pre-split which, once post-processed, produced a 35-65 train-test divide. As this training set is much smaller than what would normally be expected for developing a deep learning model, experimentation was undertaken to evaluate the hypothesis that a simpler model structure is best when low volumes of catalogue data are available during training. To this end, the dataset splits were reversed such that the test set was used for model training and the train set was used for

<b>Model Backbone</b>	<b>Data Augmentation Strategy</b>	<b>Accuracy (%)</b>		
		<b>Top-1</b>	<b>Top-5</b>	<b>Top-10</b>
EmbeddingNet	None	67.44	90.70	90.70
VarvaraNet	None	88.37	100.00	100.00
EmbeddingNet	Colour Jitter	69.77	88.37	90.70
VarvaraNet	Colour Jitter	72.09	88.37	97.67
EmbeddingNet	Perspective Shift	74.42	90.70	97.68
VarvaraNet	Perspective Shift	74.42	88.37	95.35
EmbeddingNet	Perspective Shift & Colour Jitter	60.47	88.37	90.70
<b>VarvaraNet</b>	<b>Perspective Shift &amp; Colour Jitter</b>	<b>90.70</b>	<b>100.00</b>	<b>100.00</b>

Table 7.4 Results of SNN training for the task of most likely catalogue matching on the reversed SDRP dataset. The best performing model is highlighted in bold.

evaluation. For consistency, the same model backbones and data augmentation strategies were utilised during training, as were the optimal NDD AU SMRU hyperparameters.

The top-1, top-5, and top-10 accuracies for the trained models can be seen in Table 7.4. By reversing the train-test split, a drop in performance is observed for all bar one of the models trained using an EmbeddingNet backbone. For the single model where performance improves, increases of 11.92% top-1 and 0.18% top-10 accuracies are observed, however a drop of 1.80% top-5 accuracy is also seen. For those models trained using a VarvaraNet backbone, increases in performance are observed for all models, with only two reporting slight drops in top-5 accuracy.

In general, it can be observed that the increase in training data has led to an overall performance boost for models trained using a VarvaraNet backbone whilst those with an EmbeddingNet backbone observe a performance drop. As such, the best performing model for the reversed split SDRP dataset now makes use of a VarvaraNet backbone alongside both Colour Jitter and Perspective Shift data augmentation strategies. This supports the hypothesis that backbone architecture selection is influenced by the size of the initial training set, a finding backed up by other works in the area [286, 287]. Further, it seems to be the case that more complex models are required to fully capture the fine-grained nature of larger photo-id catalogues. These findings also provide further evidence to support the idea that data augmentation strategies are catalogue dependent.

It should be noted however that the best performing model achieves 100% top-5 and top-10 accuracies, suggesting the possibility of model overfitting. Whilst the model may be capable of always providing a match within the first five results, this is likely a product of dataset size causing an inflation in model performance. It is expected that these accuracies

would decrease when utilising more data, either through an increase in the number of classes or examples per class, as this would provide an increase in data variation.

## 7.4 Considering Catalogue Matching as a Standard Classification Task

In order to better understand the performance of SNNs for the task of most likely catalogue matching, an evaluation was undertaken whereby the task was approached as a standard image classification problem. Using the NDD AU SMRU dataset, multiple backbone architectures (ResNet50, ResNet101 [103], VGG16 [152], EmbeddingNet, and VarvaraNet) were trained and evaluated against top-1, top-5, and top-10 accuracy metrics. This range was chosen to help identify whether model depth or setup has an effect on classification performance.

As these models only require a single example image per training step, rather than a triplet of images as required when training SNNs with Triplet Ranking Loss, Cross Entropy Loss was instead utilised. This function compares the predicted class probabilities outputted by the model to those of the ground truths. A loss,  $L$ , is computed using Equation 7.1 (where  $n$  is the number of classes,  $t_i$  is the ground truth label for the  $i^{th}$  class, and  $p_i$  is the Softmax probability for the  $i^{th}$  class) which in turn can be used to inform how the weights of the model should be changed. A model which performs perfectly would yield a Cross Entropy Loss of 0.

$$L = - \sum_{i=1}^n t_i \log(p_i) \quad (7.1)$$

Bayesian hyperparameter optimisation during model training was performed using the Optuna framework [277], as in Section 6.3.1. Due to the required change in loss function, the total number of hyperparameters required by the models was reduced. The search aimed to find an optimal `log uniform` learning rate between  $1 \times 10^{-6}$  and  $1 \times 10^{-3}$ , a dropout [72] `uniform` probability between 0.1 and 0.7, a `log uniform` weight decay between  $1 \times 10^{-6}$  and  $1 \times 10^{-1}$ , an `int` step size between 5 and 10, and a  $\gamma$  `log uniform` value between  $1 \times 10^{-3}$  and  $1 \times 10^{-1}$ . The learning rate optimiser was also tuned, allowing for a `categorical` choice between SGD or Adam [80]. When optimising ResNet50, ResNet101, and VGG16, hyperparameter optimisation included a `categorical` choice of starting from a pre-trained state (utilising ImageNet [169] weights) or not.

A total of 100 iterations were undertaken, with the optimal hyperparameters for each final model architecture decided by its best performing trial. The optimal hyperparameters for each model architecture can be seen in Appendix F. Due to memory constraints, the number

Model Architecture	Accuracy (%)		
	Top-1	Top-5	Top-10
ResNet50 [103]	3.46	11.79	27.03
ResNet101 [103]	4.07	14.43	27.85
VGG16 [152]	2.64	15.45	33.54
EmbeddingNet	16.06	35.16	50.41
VarvaraNet [265]	6.10	29.67	58.54

Table 7.5 Top- $N$  accuracies of the best performing image classification models on the NDD AU SMRU dataset.

of samples per class for each training batch was reduced from 9 for SNNs to 3 for ResNet50 and 2 for all other models.

The top-1, top-5, and top-10 accuracies for the best performing models can be seen in Table 7.5. Compared to the best performing SNN, these model architectures all struggle to achieve high top-1 accuracy on the NDD AU SMRU dataset. For comparison, the best performing SNN trained on this dataset achieves 36.78%, 54.47%, and 55.28% higher top-1, top-5, and top-10 accuracies respectively. The results provided suggest that shallower models generally perform best on this data, with the shallowest model, the EmbeddingNet, achieving the highest top-1 and top-5 accuracies, with its top-10 accuracy being only narrowly behind VarvaraNet model (the second shallowest architecture). The poor accuracy achieved here by the deeper models is hypothesised to be due to the extreme fine-grained nature and high intra-class variability of the dataset, as well as the relative lack of data to train such deep models causing overfitting.

Examining the Bayesian optimised hyperparameter tuning runs suggests that starting from a pre-trained state is the most important indicator of model performance when utilising a ResNet architecture for this task. An example of this can be seen in Figure 7.7 which shows the hyperparameter importances for the ResNet50 tuning run, including a pre-training importance of 0.81 – for ResNet101 this value was 0.72. Whilst it is expected that this hyperparameter would have high importance due to the relatively small size of the NDD AU SMRU dataset and deepness of the architectures, the lack of importance for all other hyperparameters may suggest that this model setup is incapable of identifying meaningful features which allow the models to consistently differentiate between classes.

When training the VGG16 model however, the optimal model found did not require pre-training, with a hyperparameter importance of 0.02. Instead, weight decay was the most important hyperparameter with an importance of 0.39 as seen in Figure 7.8. This lack of pre-training importance may be due to the shallower depth of the architecture. However,

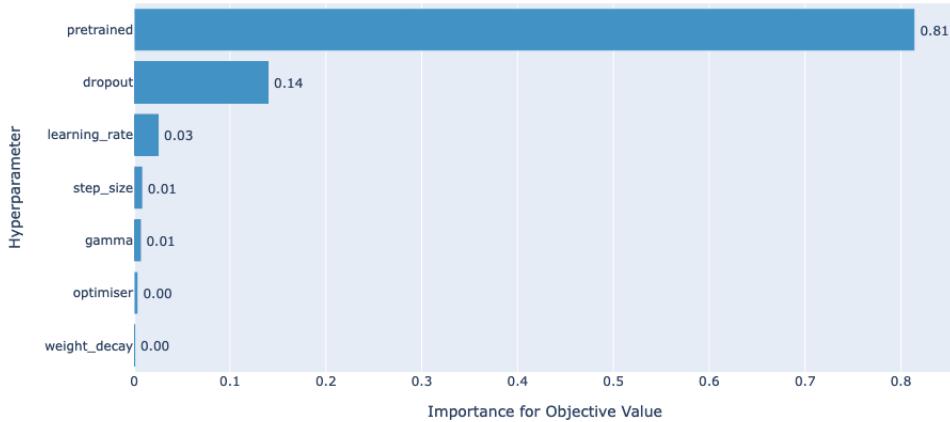


Figure 7.7 Importance scores for the hyperparameter tuning of the ResNet50 model.

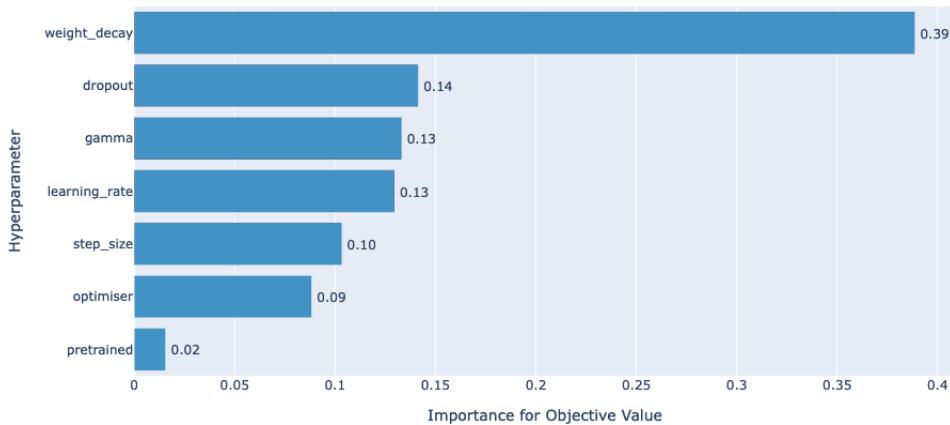


Figure 7.8 Importance scores for the hyperparameter tuning of the VGG16 model.

as seen in Table 7.5, the VGG model also did not achieve high accuracies on the NDD AU SMRU dataset.

These findings suggest that the use of triplets when training SNNs may allow for more meaningful features to be extracted, leading to more accurate most likely catalogue matching and top-1 accuracy. As such, this indicates that the use of an SNN to generate embeddings, plotting these into a latent space, and utilising clusters to provide most likely matches is a superior approach to training a standard image classifier, notwithstanding the difficulties which would arise if individuals not seen at train time were required to be classified.

## 7.5 Summary

This chapter examines the robustness of SNNs for the task of automatic most likely catalogue matching. To begin, the effect on model performance of additional photo-id data is examined. This work shows that while the additional data improves overall generalisability, a model capable of vastly reducing the search space can be obtained utilising only the data collected in a single fieldwork study, even when running inference on data from a different spatio-temporal environment.

Next, the effect of background retention on embedding generation is examined. By training a model using bounding box imagery rather than masks and examining the embeddings generated for various inputs, it can be seen that embeddings can be influenced more by feature heavy background surrounding a dorsal fin than the fin itself, at least when the model is trained on data collected over a small spatio-temporal scale. This emphasises the importance of detection masks over bounding boxes, further reinforcing the decision made in Section 4.2.1.

The generalisability of utilising SNNs for the task of automatic most likely catalogue matching is then examined through the use of a second smaller photo-id catalogue provided by the Chicago Zoological Society’s Sarasota Dolphin Research Program. Training an SNN on this data yields a model which is highly accurate, achieving 81.25% top-1, 95.00% top-5, and 97.50% top-10 accuracies. This provides evidence to suggest that SNNs are a viable approach to automatic most likely matching, regardless of catalogue size, species, or environment.

Finally, the task of most likely catalogue matching is framed as a standard image classification problem in order to provide a baseline with which to compare the previously developed SNN approach. Experiments show that the model architectures trained are incapable of classifying images in the NDD AU SMRU dataset with high accuracy. This is hypothesised to be a result of the extreme fine-grained nature and high intra-class variability rendering the models incapable of identifying meaningful features for classification, which could be negated thanks to the use of triplets during training and clustering generated embeddings into a latent space.

Experimentation throughout this chapter has shown that SNNs are an accurate and generalisable approach to the task of catalogue matching. When accompanied by a dorsal fin detector and post-processing methodology, a fully automatic photo-id aid can be created which is robust to changes in species of interest, geography, and time. A summary of the work undertaken throughout this thesis to develop an automatic photo-id aid is provided in Chapter 8. Possible avenues for further work are also explored in detail.

# **Chapter 8**

## **Conclusion**

In this chapter a summary of the work presented in this thesis is provided, with the work evaluated against the research aims outlined in Chapter 1. Next, potential avenues for future research are explored.

### **8.1 Thesis Summary**

Motivated by the problem space as outlined in Chapter 1, this thesis explores the use of computer vision and deep learning for the task of automated photo-id catalogue curation and most likely matching. Before beginning work to fulfil the thesis' aims, Chapter 2 provides the relevant background knowledge required for understanding work described in later Chapters; this includes an introduction to current photo-id methodology, deep learning and computer vision fundamentals, and the current state-of-the-art in photo-id aids. It is here where the problems with current approaches are evaluated in detail, providing further justification for the work undertaken in this thesis.

Before work on an automated photo-id aid could begin, two photo-id catalogues were transformed into useable computer vision datasets. The first is a coarse-grained dataset containing examples of Indo-Pacific bottlenose dolphins collected by Newcastle University's Marine MEGAfuna Lab whilst on expedition to Zanzibar, Tanzania in 2015 [10]. The second is a multi-granular dataset containing photo-id data from fieldwork undertaken during this thesis to create a catalogue of resident bottlenose and white-beaked dolphins present in the waters around Northumberland, UK, during Summer 2019. This catalogue was then converted into a dataset useful for computer vision model training, called the Northumberland Dolphin Dataset 2020 (NDD20). The creation of both datasets is outlined in Chapter 3.

In Chapter 4, the first step towards automatic photo-id catalogue management is tackled through the development of a coarse-grained detector capable of generating mask predictions for above water fieldwork imagery, focussing on dolphins as the cetacean species of interest. The environmental and technical requirements of the detector are outlined for use as success metrics, whilst the use of mask predictions rather than bounding boxes is justified. The use of approaches such as GrabCut [249] and SURF feature extraction [209] is explored, and their unsuitability outlined. To this end, a Mask R-CNN [36] detector is trialled, trained using the Zanzibar dataset created in Chapter 3.

During model development the use of transfer learning and various data augmentation strategies are explored, helping to mitigate the issues inherent with training deep computer vision models using dataset sizes typical of photo-id surveys. Model hyperparameter optimisation via a grid search is then performed, before the best performing model on the Zanzibar data is determined. This optimal model achieves high mean average precision (mAP) over a range of key intersection over union (IOU) thresholds, confirming coarse-grained detection of cetaceans is possible even in noisy environments and where large variation in the region of interest is present.

The developed detector's robustness to species of interest and spatio-temporal changes are evaluated using the NDD20 dataset. Using this data, it can be seen that the detector is capable of high mAP at a range of IOU thresholds, confirming the model is robust to species of interest and spatio-temporal changes. As such, this suggests that no model fine-tuning or re-training is required for use in future photo-id surveys provided the model is trained to a high accuracy using previously obtained imagery.

The optimal model is then utilised in Chapter 5 to explore the post-processing techniques required to allow for a reduction in the computational expense of downstream models and the removal of unneeded background noise, whilst at the same time ensuring no useful individually identifying markings are lost. The ability to achieve this highlights that it is possible to fully remove the need for the data pre-processing which is currently performed by cetacean researchers, either when performing photo-id matching manually or with the use of aids.

Detections from NDD20 are then post-processed to produce a second dataset representative of detector output, allowing for the training of the next model in the pipeline. Additional imagery of the individuals, provided by the University of Aberdeen and the University of St Andrews, are also included in the identification training dataset, called NDD AU SMRU, as a 23 individual overlap was determined between the catalogue created for Northumberland and those for Eastern Scotland held by the partner institutions. This additional data helps combat the issue of small dataset size when performing model training.

In Chapter 6 a model capable of fine-grained, few-shot cetacean re-identification is created, aiding researchers by vastly reducing their search space when performing catalogue matching. Beginning by outlining the requirements this model must adhere to, two possible approaches are evaluated. Through this, the use of a Siamese Neural Network (SNN) based approach is determined most suitable. Model training is performed using two distinct backbone architectures; a relatively simple one called EmbeddingNet and a more complex one known as VarvaraNet based on work by Vetrova *et al.* [265]. The use of pairwise or triplet ranking losses is explored, as well as online semi-hard triplet mining and hyperparameter tuning via Bayesian optimisation [277]. A range of models were generated, trained using the previously created NDD AU SMRU dataset. Evaluation of the models is undertaken using top-1, top-5, and top-10 accuracies. The best performing model, a VarvaraNet trained without the use of any data augmentation, is capable of vastly reducing the search required by a cetacean researcher when performing most likely catalogue matching.

The ability of SNNs to allow for the flagging of potentially previously uncatalogued individuals is also explored. Thanks to the model's capability to reduce a high dimensional input down to a low dimensional representation, or embedding, existing catalogue examples can be plotted into a latent space such that they create class clusters. By utilising class prototypes and Euclidean distance measurements, as well as the K-Nearest Neighbours algorithm, this chapter shows that the flagging of potentially previously unseen individuals can be achieved. Any new individuals that are added to the catalogue can also easily be added to the model without the need for re-training thanks to embedding clustering and class prototyping. Further experimentation using a modified version of NDD AU SMRU presents some evidence to suggest that splitting individual classes in two based on which side of the dorsal fin is captured may lead to improved model performance, but further tests are required to confirm this.

In Chapter 7 the robustness of SNNs for the task of most likely catalogue matching is explored. The effects of dataset variation on model performance are quantified, with experimentation suggesting that the retention of background noise through the use of bounding box detections leads to inflated model performance when data has been collected over a small temporal scale. This presents evidence to suggest that the initial decision to perform full background removal through the use of masks was the correct decision, even at the expense of greater computation, and may present an inherent flaw in other photo-id aids which do not remove all background.

The generalisability of catalogue matching using SNNs is further evaluated through the use of a second photo-id catalogue, provided by the Sarasota Dolphin Research Program. Work here shows that whilst catalogue matching can be performed on other datasets, rather

than there being something inherent to the NDD AU SMRU dataset which made this possible, a photo-id catalogue must first exist in order for training to take place. Further, the threshold values utilised in uncatalogued individual thresholding must be tuned. Finally in Chapter 8, the work undertaken in this thesis is summarised, its contributions outlined, and avenues for future research explored.

## 8.2 Evaluation Against Thesis Aims

As outlined in Chapter 1, the aim of this thesis was to **design, implement, and evaluate a system for fully automatic catalogue matching based on unprocessed photo-id field-work imagery**. This was then separated into four research questions (detailed in Section 1.1), each of which should be answered in an attempt to fulfil the overall research aim.

The first question asks whether it is possible to remove or greatly reduce the need for manual pre-processing of photo-id data in a fully automated way through the use of coarse-grained detection. Work undertaken in Chapter 4 answers this question, proving that it is indeed possible. The model developed during this chapter is capable of producing highly accurate detections when evaluated using a variety of mAP@IOU thresholds. Through training on unprocessed fieldwork data, the model is able to detect highly variant regions of interest in large-scale, noisy, above water photo-id imagery. When coupled with the post-processing methodology outlined in Chapter 5, the need for manual data pre-processing, such as cropping or rotation, is removed. If pre-processing of photo-id data was required for means other than most likely catalogue matching, or researchers did not wish to pass the data downstream, then the model could be used as a stand-alone system for quickly cropping down fieldwork imagery to only the dorsal fin.

The second question posits if it is possible to perform detection post-processing in such a way as to both reject likely false positives and remove noise whilst retaining identifiable markings present on the animals. The post-processing methodology outlined in Chapter 5 confirms this is possible. The use of a *dolphin-like* colour threshold, described in Section 5.4, allows for the removal of likely erroneous detections whilst keeping those that are over-exposed but correct, reducing the false-positive retention rate. Further, the inclusion of morphological transformations ensures that any identifying information on the dorsal fin which may have been missed is included in the detection, provided the information is surrounded by pixels that have been correctly classified. Coupled with background removal and cropping, this ensures unneeded noise is removed whilst identifiable markings are retained.

The third question asks if it is possible to perform highly accurate most likely photo-id catalogue matching based on extreme fine-grained information, even when operating on few-shot data. Work undertaken in Chapter 6 confirms this can be achieved through the use of an SNN trained using online semi-hard triplet mining. The model is capable of producing sufficiently distinct embeddings for classes within the extreme fine-grained NDD AU SMRU dataset, where there is small inter-class but high intra-class variation. Further, the dataset is also few-shot given the free roaming nature of the individual animals present.

High top-1, top-5, and top-10 accuracies are achieved on this dataset, confirming highly accurate automated most likely photo-id catalogue matching is possible. This is further confirmed through evaluation of the approach against the SDRP dataset, where high accuracies are once again obtained.

Finally, the fourth question asks how generalisable the models created throughout this work are to changes in species of interest and spatio-temporal shifts. In Chapter 4, the coarse-grained cetacean detector is shown to be highly robust to these changes. High mAP is observed at a range of IOU thresholds on previously unseen photo-id catalogue data collected in a different spatio-temporal area, and containing two new cetacean species, than that on which the model was trained. The approach to most likely catalogue matching is also shown to be generalisable, achieving high accuracies on both the NDD AU SMRU and SDRP datasets in Chapters 6 and 7 respectively. However, greater limitations on the generalisability of the catalogue matching model are observed when compared to the dorsal fin detector, as this model requires retraining on the new photo-id catalogue and new thresholds for potentially uncatalogued individual thresholding must be located.

In summary, as all research questions have been answered it can be stated that the work undertaken in this thesis has achieved its aims.

## 8.3 Future Research Directions

This section describes several interesting topics for future research which builds upon work undertaken in this thesis.

### 8.3.1 Effect of Initial Catalogue Size on Most Likely Matching

Performance of a model for the task of most likely matching has been shown to be influenced by initial catalogue size, most notably in Sections 7.1 and 7.3.5. Due to the data-hungry nature of training deep computer vision models, and the relatively small amounts of data available to this work, it was not feasible to explore in detail the effect of initial catalogue

size on model performance. Work by Wu *et al.* [288] has shown that training data can impact embedding generation as much as the chosen loss function. As such, it is highly likely that embedding generation is improved the more training data is provided, allowing the model to create better defined class clusters. This will in turn allow for improved detection of potentially uncatalogued individuals.

Due to the lack of data availability in this space, caused by an apprehension for cetacean researchers to release all data they have on resident populations (thanks to issues ranging from conservation of the animals and their habitat to safeguarding their own lab's future research), it was not possible to obtain a large scale, multi-year catalogue. Initial plans for this work included the provision for data collection around the Northumberland coastline with the goal of increasing the data available in the Northumberland Dolphin Dataset. These plans were scuppered due to the unforeseen health situation from 2020 onwards which prevented fieldwork surveys from occurring.

This lack of data availability meant that a meaningful study into the effect of initial catalogue size on most likely matching performance was not possible. A study like this is extremely important, as it would provide researchers with some minimum number of images they would need in order to make use of a system like the one presented in this work. As such, future research should endeavour to ensure that a study like the one described is undertaken. Understanding the relationship between embedding generation and catalogue size may also allow for the development of a metric alerting researchers to data drift (discussed in more detail in Section 6.4.3).

### 8.3.2 Effect of Background Over Large Temporal Scales

The provision of a large multi-year catalogue would also allow for a study into the effect of background noise over large temporal scales. Work undertaken in Section 7.2 suggests that, when utilising bounding box detections instead of pixel-wise masks, embedding generation is influenced more by the retained background than the region of interest (RoI), provided the background is feature heavy. This is hypothesised to be caused by the small temporal scale over which the images were collected; the background remains mostly consistent even though the RoI changes greatly due to the fast roaming nature of the animals.

If a dataset could be obtained which consisted of a large number of class examples with wide variation in the background, a property inherent of a multi-year dataset due to its broad temporal scale, then the effect of background on embedding generation could be fully quantified. At present, it is unclear whether embeddings will always be influenced regardless of background variation or whether the model will learn to generate embeddings which ignore this. If it is the former, then this has wide ranging implications for the development of

future photo-id aids, and may provide reason to further examine the performance of current aids which perform matching without full background removal such as finFindR [201] or work by Lee *et al.* [200].

Exploration into the use of domain generalisation techniques should also be explored to reduce the effect of background on embedding generation. Recent years have seen the publication of a wide range of generalisation techniques for image representation [289–295], and it may be the case that the use of one or more of these may help the SNN focus on features present in the RoI rather than the background. Further, the use of out of distribution training data may prove beneficial. Work undertaken by Lee *et al.* [296] showed that computer vision models may confuse background cues to be a foreground concept due to correlation, for example parsing railway tracks as part of trains as they are always shown together in training images. By providing explicit negative examples (e.g. images containing tracks only), the model is able to shift its understanding of what constitutes a train. Advances in the domain of person re-identification may also be applicable here, helping to reduce the bias introduced through background inclusion [285, 297, 298]. Utilising these techniques during model training may make it possible to shift embedding generation focus away from the background.

### 8.3.3 Further Per-Side Experimentation

Section 6.5 outlines experimentation performed using the NDD AU SMRU dataset to examine whether automated photo-id matching should, like its manual counterpart, split individual classes on a per-side basis. Results obtained suggest that training an SNN on a per-side basis improves model performance, reduces embedding dimensionality, and necessitates a less computationally expensive backbone architecture.

It was not possible to recreate this experiment however using the SDRP dataset due to the limited number of examples per class. As such it cannot be said with certainty whether splitting classes based on the side of the dorsal fin visible is optimal regardless of the data, or whether there is a property inherent to the NDD AU SMRU dataset that results in increased model performance and reduced complexity when split which is not observed in photo-id catalogues globally. If more data was obtained to facilitate the aforementioned future research, the generalisability of a per-side approach could also be validated.

### 8.3.4 Creating a Generalised Model for Catalogue Matching

One limitation of the model utilised for most likely matching currently is the need to re-train the SNN for each photo-id catalogue. As a result, initial manual curation must be performed

before the methodology can be applied. The threshold values required for uncatalogued individual detection through both Euclidean distance measurement between class prototypes and the K-Nearest Neighbours algorithm are also shown to be dataset dependent thanks to work outlined in Sections 6.4.2 and 7.3.3. It may be the case that determining these threshold values can be automated by utilising the properties of the latent space and the training data embeddings, or through clustering algorithms such as Mean Shift [299] or DBSCAN [300]. These approaches should be explored in future work.

The feasibility of a more general SNN capable of catalogue-agnostic photo-id should be examined in future research. There is little published work examining the feasibility of these models however, with only work by Melekhov *et al.* [301] suggesting it is possible for the task of landmark recognition. It may be the case that training a catalogue-agnostic model is not possible due to the extreme fine-grained nature of the classes, but future research should confirm this.

It should be noted however that these limitations do not apply to the detector model, which has been found in Section 4.5.1 not to require re-training when applied to new photo-id catalogues.

### 8.3.5 Verification of Dataset Ground Truth Labels

The photo-id catalogues utilised in this thesis are labelled by ecologists who are experts in the populations they survey, meaning there is little to no chance of someone not extremely familiar with the survey population detecting any mislabels present. Because of this, the generated computer vision datasets' ground truth labels (the individual ID classifications) are assumed to be correct.

Work presented by Tyson Moore *et al.* [35] shows that discrepancies in the number of individuals present in a photo-id catalogue can exist both between research groups and researchers within the same group. As a result, the accuracies presented in this work can be seen as an indication of how much the model agreed with the person who labelled the data, in line with other works in this area [302]. As any mislabelled examples present in the datasets will negatively impact overall system performance, future work should therefore aim to verify that no mislabels are present within the data utilised, making use of existing literature in this area [303, 304].

### 8.3.6 Expansion to Other Data Subjects

Work undertaken in this thesis has focussed on photo-id imagery of dolphins captured from above the waterline. Future work should endeavour to expand the developed pipeline for use

with other cetacean species, such as whales and porpoises, other marine life, and terrestrial species. Furthermore, the use of below water photo-id imagery [11] and video data should be explored. It is expected that a change in data subject would necessitate re-training of both the coarse-grained detector and the fine-grained catalogue matcher models although similarly high levels of performance as presented throughout this thesis are considered likely provided sufficient amounts of data are utilised.

### 8.3.7 Transforming the Model Pipeline

In 2017 the domain of Natural Language Processing (NLP) was revolutionised thanks to the introduction of the Transformer model architecture, introduced by Vaswani *et al.* [127]. Up until this point, tasks like sentiment analysis or text translation were dominated by models based around long short-term memory [305] or recurrent neural networks [306].

The Transformer architecture makes use of attention, a mechanism that allows a model to understand long-range interactions between values in a sequence. This removes the need for recurrence, allowing for the parallelisation of training. As a result, the state-of-the-art for a range of NLP tasks is now dominated by Transformers [307–310].

It was not until 2021 however that Transformers entered the vision space, thanks to work by Dosovitskiy *et al.* [311]. By splitting images into fixed size patches, analogous to words in a sentence, the authors show that it is possible to remove the need for convolution (the basis of convolutional neural networks (CNNs), see Section 2.4.1) and utilise attention for the task of image classification.

Since then Vision Transformers (ViTs) have been utilised for a variety of computer vision tasks such as object detection [116, 312, 313], segmentation [128, 314–316], and image similarity [317], often achieving state-of-the-art performance on large scale benchmark datasets. Some recent works have also made use of Siamese ViTs for tasks such as change detection [318] and image retrieval [319]. Future work should focus on examining potential performance improvements through conversion of the coarse-grained detector and fine-grained catalogue matcher models to ViTs.

It should be noted here however that this is very much a long term goal as Transformers currently require both massive amounts of data and time to train, with the original ViT models trained on the JFT-300M dataset [320], containing 300 million images, for up to 12.3k TPUv3-core-days – although recent works have shown it is possible to train Transformers using relatively smaller datasets [321].

Still, it is not currently possible to train a ViT for the work undertaken in this thesis. This however does not mean it will not be possible in the future through improvements in

architecture or training schedule, or via fine-tuning. The use of hybrid approaches should also be explored, which make use of elements from both ViTs and CNNs [125, 129].

### 8.3.8 The Creation of a Working Computer Program

Figure 1.1, shown at the beginning of this thesis, highlights the expected data flow between the proposed components. Throughout the course of this thesis, these components are created and it is shown how they can be connected together to form a fully automated photo-id curation system. However, the work undertaken stops short of actually implementing these connections and creating a working computer program which can be utilised by cetacean researchers.

As such, future work should focus heavily on program development. This would include not just Machine Learning Operations (MLOps) and data engineering work to connect the pipeline's components together [322], but the development of a backend database for the storage of photo-id imagery embeddings and a frontend graphical user interface (GUI) to facilitate researcher interaction with the pipeline in a familiar and friendly way – work considered beyond the scope of this thesis. Once created however, a working computer program would allow for an in-depth comparison study between work presented here and other related works as outlined in Table 2.1. Further, a study into the time savings provided by the developed system compared to performing catalogue matching manually could be undertaken. Once the program has been developed, cetacean researchers globally would be free to make full use of the techniques outlined in this thesis.

# Appendix

## A mAP@IOU[0.5:0.95] Scores for Mask R-CNN Grid Search Models

Model Name	mAP@IOU[x]									
	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
20190829T1458	0.774	0.748	0.724	0.709	0.633	0.555	0.423	0.265	0.092	0.000
20190829T2020	0.739	0.670	0.642	0.621	0.573	0.513	0.402	0.180	0.059	0.000
20190830T0145	0.843	0.815	0.778	0.739	0.711	0.586	0.442	0.245	0.052	0.000
20190830T0714	0.793	0.751	0.734	0.691	0.585	0.492	0.328	0.174	0.066	0.000
20190830T1443	0.732	0.724	0.702	0.630	0.592	0.531	0.379	0.199	0.075	0.000
20190830T2019	0.858	0.830	0.782	0.762	0.687	0.609	0.446	0.278	0.082	0.000
<b>20190902T0946</b>	<b>0.914</b>	<b>0.907</b>	<b>0.890</b>	<b>0.864</b>	<b>0.852</b>	<b>0.793</b>	<b>0.692</b>	<b>0.497</b>	<b>0.150</b>	<b>0.000</b>
20190904T2004	0.896	0.878	0.873	0.838	0.791	0.665	0.489	0.235	0.040	0.000
20190905T1813	0.937	0.917	0.907	0.904	0.851	0.775	0.713	0.452	0.113	0.000
20190905T1826	0.915	0.909	0.892	0.869	0.833	0.762	0.636	0.418	0.124	0.016
20190905T2202	0.848	0.823	0.814	0.773	0.712	0.619	0.555	0.281	0.116	0.000
20190905T2336	0.822	0.799	0.751	0.736	0.695	0.640	0.536	0.265	0.116	0.000
20190906T0332	0.902	0.895	0.877	0.867	0.811	0.738	0.630	0.417	0.124	0.000
20190906T0851	0.856	0.844	0.835	0.805	0.724	0.640	0.508	0.277	0.058	0.000
20190907T0932	0.834	0.805	0.780	0.718	0.649	0.567	0.391	0.239	0.063	0.000
20190907T0933	0.844	0.805	0.774	0.748	0.709	0.592	0.471	0.243	0.069	0.000
20190907T0934	0.921	0.908	0.902	0.887	0.848	0.796	0.663	0.457	0.130	0.000
20190907T1451	0.804	0.766	0.724	0.689	0.626	0.560	0.373	0.197	0.054	0.000
20190907T1500	0.837	0.814	0.790	0.780	0.739	0.649	0.497	0.289	0.101	0.000
20190907T1545	0.850	0.809	0.784	0.747	0.714	0.612	0.446	0.229	0.058	0.000
20190907T2026	0.919	0.919	0.901	0.861	0.843	0.782	0.628	0.412	0.118	0.000
20190907T2126	0.827	0.803	0.778	0.743	0.668	0.544	0.442	0.257	0.075	0.000
20190907T2215	0.928	0.917	0.917	0.889	0.855	0.789	0.660	0.452	0.137	0.000
20190908T0202	0.817	0.771	0.757	0.727	0.678	0.594	0.508	0.281	0.071	0.000
20190908T0352	0.836	0.790	0.731	0.710	0.678	0.566	0.415	0.259	0.069	0.000
20190908T0417	0.827	0.803	0.760	0.736	0.707	0.619	0.485	0.287	0.098	0.000
20190908T0957	0.884	0.880	0.863	0.835	0.801	0.681	0.577	0.324	0.048	0.000
20190908T1102	0.790	0.775	0.712	0.699	0.650	0.581	0.480	0.256	0.110	0.000
20190908T1204	0.901	0.890	0.872	0.855	0.825	0.733	0.674	0.453	0.185	0.000
20190908T1939	0.811	0.790	0.745	0.703	0.640	0.579	0.434	0.203	0.095	0.000
20190908T2043	0.929	0.913	0.904	0.881	0.841	0.804	0.680	0.482	0.126	0.000
20190908T2139	0.910	0.894	0.879	0.863	0.797	0.686	0.571	0.349	0.082	0.000
20190909T0723	0.798	0.778	0.770	0.738	0.690	0.644	0.516	0.298	0.099	0.000
20190911T1922	0.765	0.721	0.684	0.643	0.557	0.508	0.365	0.174	0.067	0.000
20190912T0045	0.780	0.745	0.710	0.663	0.596	0.539	0.425	0.214	0.080	0.000
20190912T0608	0.910	0.902	0.892	0.861	0.838	0.771	0.642	0.463	0.197	0.000
20191101T1633	0.909	0.909	0.902	0.883	0.864	0.780	0.662	0.419	0.135	0.000
20191101T2104	0.901	0.901	0.896	0.865	0.849	0.750	0.582	0.340	0.072	0.016
20191102T0140	0.916	0.911	0.896	0.880	0.858	0.813	0.685	0.442	0.190	0.000
20191102T0615	0.902	0.891	0.886	0.873	0.819	0.745	0.661	0.410	0.158	0.000
20191102T1051	0.914	0.895	0.864	0.861	0.836	0.812	0.601	0.461	0.132	0.000
20191102T1528	0.919	0.910	0.910	0.877	0.858	0.778	0.661	0.425	0.112	0.016
20191102T2006	0.919	0.897	0.897	0.865	0.836	0.747	0.583	0.391	0.125	0.016
20191103T0044	0.901	0.888	0.863	0.853	0.832	0.736	0.616	0.348	0.090	0.000
20191103T0520	0.913	0.904	0.876	0.868	0.844	0.778	0.652	0.443	0.129	0.000
20191103T0959	0.929	0.929	0.921	0.875	0.846	0.766	0.641	0.437	0.117	0.000
20191103T1441	0.882	0.873	0.854	0.835	0.763	0.739	0.646	0.401	0.129	0.000
20191103T1921	0.926	0.905	0.901	0.878	0.836	0.768	0.643	0.360	0.115	0.000
20191104T0011	0.912	0.897	0.887	0.874	0.844	0.788	0.706	0.391	0.083	0.000
20191104T0450	0.915	0.894	0.886	0.877	0.846	0.782	0.656	0.394	0.118	0.000

Table A mAP@IOU[0.5:0.95] scores for each Mask R-CNN model trained in the Zanzibar dataset grid search. See Section 4.4.3 for model hyperparameters. Model chosen for use highlighted in bold.

## B Data Collection Camera Settings

Setting	Value
ISO	Auto
Mode	Time Value (shutter priority)
Shutter Speed	1/1000 in high light 1/800 in low light
Exposure Compensation	0.5-1 depending on light level
Lens	Autofocus
Camera Autofocus	Spot Focus

Table B Camera settings used during data collection as outlined in Section 3.2.3. Any settings not displayed remained at default for a Canon EOS 550D Digital SLR with a Canon 70–200mm zoom lens.

## C NDD20 Above Water Example Images Class Labels

Filename	Labels		
	object	species	ID
911	dolphin	BND	-
421	dolphin	BND	-
241	dolphin	BND	-
	dolphin	BND	-
	dolphin	BND	-
177	dolphin	BND	30
296	dolphin	BND	7
1547	dolphin	BND	-
2078	dolphin	BND	-
	dolphin	BND	-
549	dolphin	BND	-
1891	dolphin	BND	-
1055	dolphin	BND	-
	dolphin	BND	-
797	dolphin	BND	-
61	dolphin	BND	-
1137	dolphin	BND	-
365	dolphin	WBD	-
1010	dolphin	BND	2
	dolphin	BND	-
450	dolphin	BND	-
	dolphin	BND	-
	dolphin	BND	-

Table C Class labels for the images shown in Figure 3.3. Table ordering is consistent with that of the figure, viewing left to right.

## D NDD20 Below Water Example Images Class Labels

Filename	Labels		
	object	out of focus	ID
26	dolphin	false	71
	dolphin	true	-
2116	dolphin	false	18
803	dolphin	false	11
1554	dolphin	false	49
1857	dolphin	false	41
	dolphin	false	-
1868	dolphin	false	30
	dolphin	true	-
	dolphin	true	-
509	dolphin	false	74
1065	dolphin	true	-
	dolphin	false	47
	dolphin	true	-
1136	dolphin	false	2
383	dolphin	false	7
311	dolphin	false	30
1602	dolphin	true	-
	dolphin	false	12
441	dolphin	true	-
	dolphin	true	-
	dolphin	false	50
522	dolphin	false	51
	dolphin	false	76
1341	dolphin	false	72
	dolphin	true	-
	dolphin	true	-
2116	dolphin	false	47

Table D Class labels for the images shown in Figure 3.4. Table ordering is consistent with that of the figure, viewing left to right.

## E Optimal Hyperparameters for Per-Side NDD AU SMRU Identification

Model Backbone	Data Augmentation Strategy	Network Blocks	Initial Convolutional Layer Size	Initial Convolutional Layer Kernel Size	Dropout	Learning Rate	Optimiser	Weight Decay	Step Size	$\gamma$	Embedding Size	Triplet Ranking Loss Margin
EmbeddingNet	None	2	19	8	0.162	$6.430 \times 10^{-5}$	SGD	0.010	9	0.018	38	0.818

Table E Optimal SNN hyperparameters for the best performing SNN trained on the per-side NDD AU SMRU dataset located using Bayesian optimisation over 100 iterations. Results given to three decimal places where applicable. See Section 6.5.

## F Optimal Hyperparameters for Standard Classification Task Architectures

Architecture	Dropout	Learning Rate	Optimiser	Weight Decay	Step Size	$\gamma$	Pre-Trained?
ResNet50 [103]	0.654	$1.225 \times 10^{-4}$	Adam	$3.948 \times 10^{-4}$	8	0.054	True
ResNet101 [103]	0.669	$1.408 \times 10^{-5}$	Adam	$1.471 \times 10^{-6}$	7	0.0185	True
VGG16 [152]	0.532	$1.771 \times 10^{-6}$	Adam	$9.045 \times 10^{-6}$	5	0.006	False
EmbeddingNet	0.545	$6.999 \times 10^{-6}$	Adam	$3.667 \times 10^{-3}$	5	0.0259	N/A
VarvaraNet [265]	0.371	$1.112 \times 10^{-4}$	Adam	$9.39 \times 10^{-2}$	6	0.019	N/A

Table F Optimal SNN hyperparameters for the best performing models trained on the NDD AU SMRU dataset for the task of image classification. Results given to three decimal places where applicable. See Section 7.4.



# Bibliography

- [1] B. Zhang, H. Wang, A. Alabri, K. Bot, C. McCall, D. Hamilton, and V. Růžička, “Unsupervised Wildfire Change Detection based on Contrastive Learning,” Nov. 2022. arXiv:2211.14654 [cs].
- [2] P. T. Jackson, Y. Wang, S. Knight, H. Chen, T. Dorval, M. Brown, C. Bendtsen, and B. Obara, “Phenotypic Profiling of High Throughput Imaging Screens with Generic Deep Convolutional Features,” in *2019 16th International Conference on Machine Vision Applications (MVA)*, pp. 1–4, May 2019.
- [3] A. Stankiewicz, T. Marciniak, A. Dabrowski, M. Stopa, E. Marciniak, and B. Obara, “Segmentation of Preretinal Space in Optical Coherence Tomography Images Using Deep Neural Networks,” *Sensors*, vol. 21, p. 7521, Nov. 2021.
- [4] S. Ghosh, P. Varakantham, A. Bhatkhande, T. Ahmad, A. Andheria, W. Li, A. Taneja, D. Thakkar, and M. Tambe, “Facilitating Human-Wildlife Cohabitation through Conflict Prediction,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 12496–12502, June 2022.
- [5] C. Robin, C. Requena-Mesa, V. Benson, L. Alonso, J. Poehls, N. Carvalhais, and M. Reichstein, “Learning to forecast vegetation greenness at fine resolution over Africa with ConvLSTMs,” Nov. 2022. arXiv:2210.13648 [cs].
- [6] M. Nasir, T. Sederholm, A. Sharma, S. R. Mallu, S. R. Ghatage, R. Dodhia, and J. L. Ferres, “Dwelling Type Classification for Disaster Risk Assessment Using Satellite Imagery,” Nov. 2022. arXiv:2211.11636 [cs].
- [7] B. G. Weinstein, “A computer vision for animal ecology,” *Journal of Animal Ecology*, vol. 87, no. 3, pp. 533–545, 2018.
- [8] R. Constantine, J. A. Jackson, D. Steel, C. S. Baker, L. Brooks, D. Burns, P. Clapham, N. Hauser, B. Madon, D. Mattila, M. Oremus, M. Poole, J. Robbins, K. Thompson, and C. Garrigue, “Abundance of humpback whales in Oceania using photo-identification and microsatellite genotyping,” *Marine Ecology Progress Series*, vol. 453, pp. 249–261, May 2012.
- [9] M. Bigg, “An Assessment of Killer Whale (*Orcinus orca*) Stocks off Vancouver Island, British Columbia,” *Report of the International Whaling Commission*, vol. 32, no. 65, p. 12, 1982.

- [10] M. Sharpe and P. Berggren, “Indian Ocean humpback dolphin in the Menai Bay off the south coast of Zanzibar, East Africa is Critically Endangered,” *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 29, no. 12, pp. 2133–2146, 2019.
- [11] M.-F. Van Bressem, B. Burville, M. Sharpe, P. Berggren, and K. Van Waerebeek, “Visual health assessment of white-beaked dolphins off the coast of Northumberland, North Sea, using underwater photography,” *Marine Mammal Science*, vol. 34, pp. 1119–1133, Oct. 2018.
- [12] M. Arso Civil, N. J. Quick, B. Cheney, E. Pirotta, P. M. Thompson, and P. S. Hammond, “Changing distribution of the east coast of Scotland bottlenose dolphin population and the challenges of area-based management,” *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 29, pp. 178–196, Sept. 2019.
- [13] B. Cheney, R. Corkrey, J. W. Durban, K. Grellier, P. S. Hammond, V. Islas-Villanueva, V. M. Janik, S. M. Lusseau, K. M. Parsons, N. J. Quick, B. Wilson, and P. M. Thompson, “Long-term trends in the use of a protected area by small cetaceans in relation to changes in population status,” *Global Ecology and Conservation*, vol. 2, pp. 118–128, Dec. 2014.
- [14] K. S. Norris and K. W. Pryor, “A Tagging Method for Small Cetaceans,” *Journal of Mammalogy*, vol. 51, p. 609, Aug. 1970.
- [15] L. J. Hobbs, “Bowhead Whale Radio Tagging Feasibility Study and Review of Large Cetacean Tagging,” p. 74, 1982.
- [16] R. D. Andrews, R. W. Baird, J. Calambokidis, C. E. C. Goertz, F. M. D. Gulland, M. P. Heide-Jørgensen, S. K. Hooker, M. Johnson, B. Mate, Y. Mitani, D. P. Nowacek, K. Owen, L. T. Quakenbush, S. Raverty, J. Robbins, G. S. Schorr, O. V. Shpak, F. I. Townsend Jr., M. Uhart, R. S. Wells, and A. N. Zerbini, “Best practice guidelines for cetacean tagging,” *IWC Journal of Cetacean Research and Management*, vol. 20, pp. 27–66, Jan. 2019.
- [17] M.-F. VanBressem, B. Burville, M. Sharpe, P. Berggren, and K. VanWaerebeek, “Visual health assessment of white-beaked dolphins off the coast of Northumberland, North Sea, using underwater photography,” *Marine Mammal Science*, 2018.
- [18] K. Uriarte, A. Gorgone, A. Read, B. Balmer, R. S. Wells, P. Berggren, J. Durban, T. Eguchi, W. Rayment, and P. S. Hammond, “Recommendations for photo-identification methods used in capture-recapture models with cetaceans,” *Marine Mammal Science*, vol. 31, pp. 298–321, Jan. 2015.
- [19] J. Reisser, M. Proietti, P. Kinas, and I. Sazima, “Photographic identification of sea turtles: method description and validation, with an estimation of tag loss,” *Endangered Species Research*, vol. 5, pp. 73–82, Sept. 2008.
- [20] C. A. Langtimm, C. A. Beck, H. H. Edwards, K. J. Fick-Child, B. B. Ackerman, S. L. Barton, and W. C. Hartley, “Survival Estimates for Florida Manatees from the Photo-Identification of Individuals,” *Marine Mammal Science*, vol. 20, no. 3, pp. 438–463, 2004.

- [21] J. Holmberg, B. Norman, and Z. Arzoumanian, “Estimating population size, structure, and residency time for whale sharks *Rhincodon typus* through collaborative photo-identification,” *Endangered Species Research*, vol. 7, pp. 39–53, Apr. 2009.
- [22] P. S. Hammond, S. A. Mizroch, and G. P. Donovan, “Individual recognition of cetaceans: use of photo-identification and other techniques to estimate population parameters: incorporating the proceedings of the symposium and workshop on individual recognition and the estimation of cetacean population parameters,” *International Whaling Commission*, vol. 12, 1990.
- [23] P. G. H. Evans and P. S. Hammond, “Monitoring cetaceans in European waters,” *Mammal Review*, vol. 34, no. 1-2, pp. 131–156, 2004.
- [24] R. Payne, “Long term behavioral studies of the southern right whale (*Eubalaena australis*),” Tech. Rep. 10, International Whaling Commission, 1986.
- [25] K. A. Forney and J. Barlow, “Seasonal Patterns in the Abundance and Distribution of California Cetaceans, 1991–1992,” *Marine Mammal Science*, vol. 14, no. 3, pp. 460–489, 1998.
- [26] B. Würsig and T. A. Jefferson, “Methods of photo-identification for small cetaceans,” *Reports of the International Whaling Commission*, vol. 12, pp. 43–52, 1990.
- [27] A. Miragliuolo, B. Mussi, and G. Bearzi, “Risso’s dolphin harassment by pleasure boaters off the island of Ischia, central Mediterranean Sea,” *European Research on Cetaceans*, vol. 15, pp. 168–171, 2004.
- [28] L. J. Feyrer, M. Stewart, J. Yeung, C. Soulier, and H. Whitehead, “Origin and Persistence of Markings in a Long-Term Photo-Identification Dataset Reveal the Threat of Entanglement for Endangered Northern Bottlenose Whales (*Hyperoodon ampullatus*),” *Frontiers in Marine Science*, vol. 8, 2021. Publisher: Frontiers.
- [29] V. R. Goswami, M. D. Madhusudan, and K. U. Karanth, “Application of photographic capture–recapture modelling to estimate demographic parameters for male Asian elephants,” *Animal Conservation*, vol. 10, no. 3, pp. 391–399, 2007.
- [30] M. Clapham, E. Miller, M. Nguyen, and C. T. Darimont, “Automated facial recognition for wildlife that lack unique markings: A deep learning approach for brown bears,” *Ecology and Evolution*, vol. 10, no. 23, 2020.
- [31] R. W. Baird, A. M. Gorgone, D. J. McSweeney, A. D. Ligon, M. H. Deakos, D. L. Webster, G. S. Schorr, K. K. Martien, D. R. Salden, and S. D. Mahaffy, “Population structure of island-associated dolphins: Evidence from photo-identification of common bottlenose dolphins (*Tursiops truncatus*) in the main Hawaiian Islands,” *Marine Mammal Science*, vol. 25, no. 2, pp. 251–274, 2009.
- [32] B. Würsig and M. Würsig, “The Photographic Determination of Group Size, Composition, and Stability of Coastal Porpoises (*Tursiops truncatus*),” *Science*, vol. 198, pp. 755–756, Nov. 1977.

- [33] C. Lockyer and R. Morris, “Some observations on wound healing and persistence of scars in *Tursiops truncatus*,” *Reports of the International Whaling Commission*, no. 12, pp. 113–118, 1990.
- [34] J. Mann, R. C. Connor, P. Tyack, and H. Whitehead, *Cetacean Societies: Field Studies of Dolphins and Whales*. University of Chicago Press, 2000.
- [35] R. B. Tyson Moore, K. W. Urián, J. B. Allen, C. Cush, J. R. Parham, D. Blount, J. Holmberg, J. W. Thompson, and R. S. Wells, “Rise of the Machines: Best Practices and Experimental Evaluation of Computer-Assisted Dorsal Fin Image Matching Systems for Bottlenose Dolphins,” *Frontiers in Marine Science*, vol. 9, p. 849813, Apr. 2022.
- [36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *arXiv:1703.06870 [cs]*, Mar. 2017. arXiv: 1703.06870.
- [37] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal, “SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification,” Sept. 2017. Number: arXiv:1707.02131 arXiv:1707.02131 [cs].
- [38] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.
- [39] C. Trotter, G. Atkinson, M. Sharpe, A. S. McGough, N. Wright, and P. Berggren, “The Northumberland Dolphin Dataset: A Multimedia Individual Cetacean Dataset for Fine-Grained Categorisation,” *arXiv:1908.02669 [cs]*, Aug. 2019. arXiv: 1908.02669.
- [40] C. Trotter, G. Atkinson, M. Sharpe, K. Richardson, A. S. McGough, N. Wright, B. Burville, and P. Berggren, “NDD20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation,” *arXiv:2005.13359 [cs]*, May 2020. arXiv: 2005.13359.
- [41] C. Trotter, N. Wright, A. Stephen McGough, M. Sharpe, B. Cheney, M. A. Civil, R. Tyson Moore, J. Allen, and P. Berggren, “Towards Automatic Cetacean Photo-Identification: A Framework for Fine-Grain, Few-Shot Learning in Marine Ecology,” in *2022 IEEE International Conference on Big Data (Big Data)*, (Osaka, Japan), pp. 1942–1949, IEEE, Dec. 2022.
- [42] R. Curry, C. Trotter, and A. S. McGough, “Application of deep learning to camera trap data for ecologists in planning / engineering – Can captivity imagery train a model which generalises to the wild?,” in *2021 IEEE International Conference on Big Data (Big Data)*, (Orlando, FL, USA), pp. 4011–4020, IEEE, Dec. 2021.
- [43] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.
- [44] L. Zhang, J. Tan, D. Han, and H. Zhu, “From machine learning to deep learning: progress in machine intelligence for rational drug discovery,” *Drug Discovery Today*, vol. 22, pp. 1680–1685, Nov. 2017.

- [45] A. Korotcov, V. Tkachenko, D. P. Russo, and S. Ekins, “Comparison of Deep Learning With Multiple Machine Learning Methods and Metrics Using Diverse Drug Discovery Data Sets,” *Molecular Pharmaceutics*, vol. 14, pp. 4462–4475, Dec. 2017.
- [46] N. G. Paterakis, E. Mocanu, M. Gibescu, B. Stappers, and W. van Alst, “Deep learning versus traditional machine learning methods for aggregated energy demand prediction,” in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1–6, Sept. 2017.
- [47] R. B. Tyson Moore, A. Barleycorn, C. Cush, A. Honaker, S. McBride, C. Toms, and R. Wells, “Final Report: Abundance and distribution of common bottlenose dolphins (*Tursiops truncatus*) near Naples and Marco Island, Florida, USA, 2018-2019,” tech. rep., The Bachelor Foundation, 2020.
- [48] C. E. Gibson, D. Williams, R. Dunlop, and S. Beck, “Using social media as a cost-effective resource in the photo-identification of a coastal bottlenose dolphin community,” *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 30, no. 8, pp. 1702–1710, 2020.
- [49] B. Cheney, P. M. Thompson, S. N. Ingram, P. S. Hammond, P. T. Stevick, J. W. Durban, R. M. Culloch, S. H. Elwen, L. Mandleberg, V. M. Janik, N. J. Quick, V. ISLAS-Villanueva, K. P. Robinson, M. Costa, S. M. Eisfeld, A. Walters, C. Phillips, C. R. Weir, P. G. H. Evans, P. Anderwald, R. J. Reid, J. B. Reid, and B. Wilson, “Integrating multiple data sources to assess the distribution and abundance of bottlenose dolphins *Tursiops truncatus* in Scottish waters,” *Mammal Review*, vol. 43, no. 1, pp. 71–88, 2013.
- [50] D. K. Caldwell, “Evidence of Home Range of an Atlantic Bottlenose Dolphin,” *Journal of Mammalogy*, vol. 36, pp. 304–305, May 1955.
- [51] W. E. Schevill and R. H. Backus, “Daily Patrol of a Megaptera,” *Journal of Mammalogy*, vol. 41, p. 279, May 1960.
- [52] F. Alves, S. Quéroutil, A. Dinis, C. Nicolau, C. Ribeiro, L. Freitas, M. Kaufmann, and C. Fortuna, “Population structure of short-finned pilot whales in the oceanic archipelago of Madeira based on photo-identification and genetic analyses: implications for conservation,” *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 23, no. 5, pp. 758–776, 2013.
- [53] T. Franklin, F. Smith, N. Gibbs, S. Childerhouse, D. Paton, W. Franklin, S. Baker, and P. Clapham, “Migratory movements of humpback whales (*Megaptera novaeangliae*) between eastern Australia and the Balleny Islands, Antarctica, confirmed by photo-identification,” p. 5, 2008.
- [54] M. Pedersen, M. Nyegaard, and T. B. Moeslund, “Finding Nemo’s Giant Cousin: Keypoint Matching for Robust Re-Identification of Giant Sunfish,” *Journal of Marine Science and Engineering*, vol. 11, p. 889, Apr. 2023.
- [55] S. Véronique, “Underwater photo-identification of sperm whales (*Physeter macrocephalus*) off Mauritius,” p. 17, 2022.

- [56] R. G. Vernazzani and E. Cabrera, “Eastern South Pacific southern right whale photoidentification catalog reveals behavior and habitat use patterns,” *MARINE MAMMAL SCIENCE*, p. 10, 2013.
- [57] T. Arnblom, “Individual identification of sperm whales,” *Report of the International Whaling Commission*, vol. 37, no. 20, pp. 201–204, 1987.
- [58] W. Perrin, B. Würsig, and J. Thewissen, *Encyclopedia of marine mammals*. Academic Press, 2009.
- [59] T. Cheeseman, “Happywhale Presentation - World Marine Mammal Conference 2019 - Rise of the Machines Workshop (available at: drive.google.com/file/d/1yRtNKagANZOgpY6852zzUpZmiFXuZmLM),” 2019.
- [60] M. Mariani, A. Miragliuolo, B. Mussi, G. F. Russo, G. Ardizzone, and D. S. Pace, “Analysis of the natural markings of Risso’s dolphins (*Grampus griseus*) in the central Mediterranean Sea,” *Journal of Mammalogy*, vol. 97, pp. 1512–1524, Dec. 2016.
- [61] M. Rosso, A. Moulins, and M. Würtz, “Colour patterns and pigmentation variability on striped dolphin *Stenella coeruleoalba* in north-western Mediterranean Sea,” *Marine Biological Association of the United Kingdom. Journal of the Marine Biological Association of the United Kingdom*, vol. 88, pp. 1211–1219, Sept. 2008. Num Pages: 9 Place: Cambridge, United Kingdom Publisher: Cambridge University Press.
- [62] P. S. Hammond, T. B. Francis, D. Heinemann, K. J. Long, J. E. Moore, A. E. Punt, R. R. Reeves, M. Sepúlveda, G. M. Sigurðsson, M. C. Siple, G. Víkingsson, P. R. Wade, R. Williams, and A. N. Zerbini, “Estimating the Abundance of Marine Mammal Populations,” *Frontiers in Marine Science*, vol. 8, p. 735770, Sept. 2021.
- [63] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A K-Means Clustering Algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [64] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, Dec. 1943.
- [65] Z. Luo, H. Liu, and X. Wu, “Artificial neural network computation on graphic process unit,” in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 1, pp. 622–626 vol. 1, July 2005. ISSN: 2161-4407.
- [66] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, “Tensorflow: a system for large-scale machine learning.,” in *OSDI*, vol. 16, pp. 265–283, 2016.
- [67] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in PyTorch,” p. 4, 2017.
- [68] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, pp. 2278–2324, 1998.
- [69] G. Griffin, A. Holub, and P. Perona, “Caltech-256 Object Category Dataset,” Mar. 2007.

- [70] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (Miami, FL), pp. 248–255, IEEE, June 2009.
- [71] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [72] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, p. 30, 2014.
- [73] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *arXiv:1502.03167 [cs]*, Feb. 2015. arXiv: 1502.03167.
- [74] I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts,” *arXiv:1608.03983 [cs, math]*, Aug. 2016. arXiv: 1608.03983.
- [75] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond Empirical Risk Minimization,” *arXiv:1710.09412 [cs, stat]*, Oct. 2017. arXiv: 1710.09412.
- [76] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv:1609.04747 [cs]*, Sept. 2016. arXiv: 1609.04747.
- [77] L. Bottou and O. Bousquet, “The Tradeoffs of Large Scale Learning,” in *Advances in Neural Information Processing Systems 20* (J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds.), pp. 161–168, Curran Associates, Inc., 2008.
- [78] N. Qian, “On the momentum term in gradient descent learning algorithms,” *Neural Networks*, vol. 12, pp. 145–151, Jan. 1999.
- [79] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.
- [80] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv:1412.6980 [cs]*, Dec. 2014. arXiv: 1412.6980.
- [81] S. J. Reddi, S. Kale, and S. Kumar, “On the Convergence of Adam and Beyond,” *arXiv:1904.09237 [cs, math, stat]*, Apr. 2019. arXiv: 1904.09237.
- [82] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization,” in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2933–2941, Curran Associates, Inc., 2014.
- [83] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, “The Loss Surfaces of Multilayer Networks,” p. 13, 2015.
- [84] M. Alber, I. Bello, B. Zoph, P.-J. Kindermans, P. Ramachandran, and Q. Le, “Backprop Evolution,” *arXiv:1808.02822 [cs, stat]*, Aug. 2018. arXiv: 1808.02822.

- [85] S. Linnainmaa, “The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors,” *Master’s Thesis (in Finnish)*, Univ. Helsinki, pp. 6–7, 1970.
- [86] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, “Learning representations by back-propagating errors,” p. 4, 1986.
- [87] S. Bengio, Y. Bengio, and J. Cloutier, “Use of genetic programming for the search of a new learning rule for neural networks,” pp. 324–327, IEEE, 1994.
- [88] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, “Random feedback weights support learning in deep neural networks,” *arXiv:1411.0247 [cs, q-bio]*, Nov. 2014. arXiv: 1411.0247.
- [89] D.-H. Lee, S. Zhang, A. Fischer, and Y. Bengio, “Difference Target Propagation,” in *Machine Learning and Knowledge Discovery in Databases* (A. Appice, P. P. Rodrigues, V. Santos Costa, C. Soares, J. Gama, and A. Jorge, eds.), Lecture Notes in Computer Science, pp. 498–515, Springer International Publishing, 2015.
- [90] A. Nøkland, “Direct feedback alignment provides learning in deep neural networks,” pp. 1037–1045, 2016.
- [91] Q. Liao, J. Z. Leibo, and T. Poggio, “How important is weight symmetry in backpropagation?”, 2016.
- [92] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A Brief Review,” *Computational Intelligence and Neuroscience*, vol. 2018, p. e7068349, Feb. 2018. Publisher: Hindawi.
- [93] P. Kim, “Convolutional Neural Network,” in *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, pp. 121–147, Berkeley, CA: Apress, 2017.
- [94] V. Dumoulin and F. Visin, “[1603.07285] A guide to convolution arithmetic for deep learning,” 2018.
- [95] Y.-L. Boureau, J. Ponce, and Y. LeCun, “A Theoretical Analysis of Feature Pooling in Visual Recognition,” p. 8, 2010.
- [96] M. D. Zeiler and R. Fergus, “Stochastic Pooling for Regularization of Deep Convolutional Neural Networks,” *arXiv:1301.3557 [cs, stat]*, Jan. 2013. arXiv: 1301.3557.
- [97] H. Gholamalinezhad and H. Khosravi, “Pooling Methods in Deep Neural Networks, a Review,” *arXiv:2009.07485 [cs.CV]*, 2020.
- [98] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.
- [99] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” pp. 580–587, 2014.

- [100] T. Elsken, J. H. Metzen, and F. Hutter, “Neural Architecture Search: A Survey,” *arXiv:1808.5377 null*, Aug. 2018. arXiv: 1808.5377.
- [101] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *arXiv:1502.01852 [cs]*, Feb. 2015. arXiv: 1502.01852.
- [102] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper With Convolutions,” pp. 1–9, 2015.
- [103] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385 [cs]*, Dec. 2015. arXiv: 1512.03385.
- [104] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.
- [105] R. Girshick, “Fast R-CNN,” *arXiv:1504.08083 [cs]*, Apr. 2015. arXiv: 1504.08083.
- [106] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *arXiv:1506.01497 [cs]*, June 2015. arXiv: 1506.01497.
- [107] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving Into High Quality Object Detection,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (Salt Lake City, UT), pp. 6154–6162, IEEE, June 2018.
- [108] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV, USA), pp. 779–788, IEEE, June 2016.
- [109] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv:2004.10934 [cs, eess]*, Apr. 2020. arXiv: 2004.10934.
- [110] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, Imyhxy, Lorna, Yifu), C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, Tkianai, YxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, “github.com/ultralytics/yolov5/tree/v7.0 - YOLOv5 SOTA Realtime Instance Segmentation,” Nov. 2022.
- [111] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications,” Sept. 2022. arXiv:2209.02976 [cs].
- [112] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021,” Aug. 2021. arXiv:2107.08430 [cs].
- [113] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, and X. Chu, “YOLOv6 v3.0: A Full-Scale Reloading,” Jan. 2023. arXiv:2301.05586 [cs].

- [114] S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du, and B. Lai, “PP-YOLOE: An evolved version of YOLO,” Dec. 2022. arXiv:2203.16250 [cs].
- [115] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” *arXiv:1512.02325 [cs]*, vol. 9905, pp. 21–37, 2016. arXiv: 1512.02325.
- [116] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, “DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection,” July 2022. arXiv:2203.03605 [cs].
- [117] J. Royo-Miquel, S. Tolu, F. E. T. Schöller, and R. Galeazzi, “RetinaNet Object Detector based on Analog-to-Spiking Neural Network Conversion,” Sept. 2021. arXiv:2106.05624 [eess].
- [118] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and Efficient Object Detection,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Seattle, WA, USA), pp. 10778–10787, IEEE, June 2020.
- [119] H. Noh, S. Hong, and B. Han, “Learning Deconvolution Network for Semantic Segmentation,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, (Santiago, Chile), pp. 1520–1528, IEEE, Dec. 2015.
- [120] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, “Simultaneous Detection and Segmentation,” *arXiv:1407.1808 [cs]*, July 2014. arXiv: 1407.1808.
- [121] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *arXiv:1411.4038 [cs]*, Nov. 2014. arXiv: 1411.4038.
- [122] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *arXiv:1511.00561 [cs]*, Nov. 2015. arXiv: 1511.00561.
- [123] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs,” *arXiv:1412.7062 [cs]*, Dec. 2014. arXiv: 1412.7062.
- [124] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, Apr. 2018.
- [125] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” Mar. 2022. arXiv:2201.03545 [cs].
- [126] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid Scene Parsing Network,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Honolulu, HI), pp. 6230–6239, IEEE, July 2017.
- [127] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, “Attention is All you Need,” p. 11, 2017.

- [128] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment Anything,” Apr. 2023. arXiv:2304.02643 [cs].
- [129] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” Aug. 2021. arXiv:2103.14030 [cs].
- [130] M. Buhrmester, T. Kwang, and S. D. Gosling, “Amazon’s Mechanical Turk: A new source of inexpensive, yet high-quality, data?,” *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.
- [131] S. Maji, “Large Scale Image Annotations on Amazon Mechanical Turk,” p. 12, 2011.
- [132] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, “Part-Based R-CNNs for Fine-Grained Category Detection,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), Lecture Notes in Computer Science, pp. 834–849, Springer International Publishing, 2014.
- [133] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, “Hierarchical Part Matching for Fine-Grained Visual Categorization,” in *2013 IEEE International Conference on Computer Vision*, (Sydney, Australia), pp. 1641–1648, IEEE, Dec. 2013.
- [134] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars, “Fine-Grained Categorization by Alignments,” pp. 1713–1720, 2013.
- [135] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, “Deformable part descriptors for fine-grained recognition and attribute prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 729–736, 2013.
- [136] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, “Dog breed classification using part localization,” in *European conference on computer vision*, pp. 172–185, Springer, 2012.
- [137] K. Chen, W. Ouyang, C. C. Loy, D. Lin, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, and J. Shi, “Hybrid Task Cascade for Instance Segmentation,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, CA, USA), pp. 4969–4978, IEEE, June 2019.
- [138] D. Neven, B. De Brabandere, M. Proesmans, and L. Van Gool, “Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth,” *arXiv:1906.11109 [cs]*, Aug. 2019. arXiv: 1906.11109.
- [139] P. Soviany and R. T. Ionescu, “Optimizing the Trade-off between Single-Stage and Two-Stage Object Detectors using Image Difficulty Prediction,” *arXiv:1803.08707 [cs]*, Aug. 2018. arXiv: 1803.08707.
- [140] X. Zhou, J. Zhuo, and P. Krähenbühl, “Bottom-up Object Detection by Grouping Extreme and Center Points,” *arXiv:1901.08043 [cs]*, Apr. 2019. arXiv: 1901.08043.
- [141] W. Xu, H. Wang, F. Qi, and C. Lu, “Explicit Shape Encoding for Real-Time Instance Segmentation,” *arXiv:1908.04067 [cs]*, Aug. 2019. arXiv: 1908.04067.

- [142] E. Xie, P. Sun, X. Song, W. Wang, D. Liang, C. Shen, and P. Luo, “PolarMask: Single Shot Instance Segmentation with Polar Representation,” *arXiv:1909.13226 [cs]*, Feb. 2020. arXiv: 1909.13226.
- [143] H. u. M. Riaz, N. Benbarka, and A. Zell, “FourierNet: Compact mask representation for instance segmentation using differentiable shape decoders,” *arXiv:2002.02709 [cs, eess]*, Oct. 2020. arXiv: 2002.02709.
- [144] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLOCT: Real-time Instance Segmentation,” Oct. 2019. arXiv:1904.02689 [cs].
- [145] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv:1804.02767 [cs]*, Apr. 2018. arXiv: 1804.02767.
- [146] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, “BlendMask: Top-Down Meets Bottom-Up for Instance Segmentation,” *arXiv:2001.00309 [cs]*, Apr. 2020. arXiv: 2001.00309.
- [147] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully Convolutional One-Stage Object Detection,” *arXiv:1904.01355 [cs]*, Aug. 2019. arXiv: 1904.01355.
- [148] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, “SOLO: Segmenting Objects by Locations,” *arXiv:1912.04488 [cs]*, July 2020. arXiv: 1912.04488.
- [149] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, “SOLOv2: Dynamic and Fast Instance Segmentation,” *arXiv:2003.10152 [cs]*, Oct. 2020. arXiv: 2003.10152.
- [150] F. Li, H. Zhang, H. xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, “Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation,” Dec. 2022. arXiv:2206.02777 [cs].
- [151] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, “EVA: Exploring the Limits of Masked Visual Representation Learning at Scale,” Dec. 2022. arXiv:2211.07636 [cs].
- [152] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556 [cs]*, Apr. 2015. arXiv: 1409.1556 version: 6.
- [153] K. Rohit Malhotra, A. Davoudi, S. Siegel, A. Bihorac, and P. Rashidi, “Autonomous Detection of Disruptions in the Intensive Care Unit Using Deep Mask R-CNN,” pp. 1863–1865, 2018.
- [154] J.-Y. Chiao, K.-Y. Chen, K. Y.-K. Liao, P.-H. Hsieh, G. Zhang, and T.-C. Huang, “Detection and classification the breast tumors using mask R-CNN on sonograms,” *Medicine*, vol. 98, May 2019.
- [155] M. Liu, J. Dong, X. Dong, H. Yu, and L. Qi, “Segmentation of Lung Nodule in CT Images Based on Mask R-CNN,” in *2018 9th International Conference on Awareness Science and Technology (iCAST)*, pp. 1–6, Sept. 2018. ISSN: 2325-5994.
- [156] R. Anantharaman, M. Velazquez, and Y. Lee, “Utilizing Mask R-CNN for Detection and Segmentation of Oral Diseases,” in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2197–2204, Dec. 2018.

- [157] Y. Qiao, M. Truman, and S. Sukkarieh, “Cattle segmentation and contour extraction based on Mask R-CNN for precision livestock farming,” *Computers and Electronics in Agriculture*, vol. 165, p. 104958, Oct. 2019.
- [158] T. Zhao, Y. Yang, H. Niu, D. Wang, and Y. Chen, “Comparing U-Net convolutional network with mask R-CNN in the performances of pomegranate tree canopy segmentation,” in *Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques and Applications VII*, vol. 10780, p. 107801J, International Society for Optics and Photonics, Oct. 2018.
- [159] H.-S. Lee and B.-S. Shin, “Potato Detection and Segmentation Based on Mask R-CNN,” *Journal of Biosystems Engineering*, Oct. 2020.
- [160] P. Chu, Z. Li, K. Lammers, R. Lu, and X. Liu, “DeepApple: Deep Learning-based Apple Detection using a Suppression Mask R-CNN,” *arXiv:2010.09870 [cs]*, Oct. 2020. arXiv: 2010.09870.
- [161] M. Buric, M. Pobar, and M. Ivasic-Kos, “Ball Detection Using Yolo and Mask R-CNN,” in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 319–323, Dec. 2018.
- [162] M. Pobar and M. Ivašić-Kos, “Detection of the leading player in handball scenes using Mask R-CNN and STIPS,” in *Eleventh International Conference on Machine Vision (ICMV 2018)*, vol. 11041, p. 110411V, International Society for Optics and Photonics, Mar. 2019.
- [163] D. Nguyen, T. Le, T. Tran, H. Vu, T. Le, and H. Doan, “Hand segmentation under different viewpoints by combination of Mask R-CNN with tracking,” in *2018 5th Asian Conference on Defense Technology (ACDT)*, pp. 14–20, Oct. 2018.
- [164] C. J. Burke, P. D. Aleo, Y.-C. Chen, X. Liu, J. R. Peterson, G. H. Sembroski, and J. Y.-Y. Lin, “Deblending and classifying astronomical sources with Mask R-CNN deep learning,” *Monthly Notices of the Royal Astronomical Society*, vol. 490, pp. 3952–3965, Dec. 2019. Publisher: Oxford Academic.
- [165] S. Nie, Z. Jiang, H. Zhang, B. Cai, and Y. Yao, “Inshore Ship Detection Based on Mask R-CNN,” in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 693–696, July 2018. ISSN: 2153-7003.
- [166] J. Hong, M. Fulton, and J. Sattar, “TrashCan: A Semantically-Segmented Dataset towards Visual Detection of Marine Debris,” *arXiv:2007.08097 [cs]*, July 2020. arXiv: 2007.08097.
- [167] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” Dec. 2020. original-date: 2019-09-05T21:30:20Z.
- [168] A. Waleed, “Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow,” 2017. original-date: 2017-10-19T20:28:34Z.
- [169] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” p. 60, 2009.

- [170] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-Aware Minimization for Efficiently Improving Generalization,” *arXiv:2010.01412 [cs, stat]*, Dec. 2020. arXiv: 2010.01412.
- [171] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, “Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs,” p. 2, 2011.
- [172] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, “Caltech-UCSD Birds 200,” Sept. 2010.
- [173] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset - CaltechAUTHORS,” 2011.
- [174] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, “Fine-Grained Visual Classification of Aircraft,” *arXiv:1306.5151 [cs]*, June 2013. arXiv: 1306.5151.
- [175] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer, “Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna,” *Scientific Data*, vol. 2, p. 150026, June 2015. Number: 1 Publisher: Nature Publishing Group.
- [176] S. Beery, D. Morris, and P. Perona, “The iWildCam 2019 Challenge Dataset,” *arXiv:1907.07617 [cs]*, July 2019. arXiv: 1907.07617.
- [177] M. A. Tabak, M. S. Norouzzadeh, D. W. Wolfson, S. J. Sweeney, K. C. Vercauteren, N. P. Snow, J. M. Halseth, P. A. D. Salvo, J. S. Lewis, M. D. White, B. Teton, J. C. Beasley, P. E. Schlichting, R. K. Boughton, B. Wight, E. S. Newkirk, J. S. Ivan, E. A. Odell, R. K. Brook, P. M. Lukacs, A. K. Moeller, E. G. Mandeville, J. Clune, and R. S. Miller, “Machine learning to classify animal species in camera trap images: Applications in ecology,” *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 585–590, 2019.
- [178] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning,” *Proceedings of the National Academy of Sciences*, vol. 115, pp. E5716–E5725, June 2018. Publisher: National Academy of Sciences Section: PNAS Plus.
- [179] M. Willi, R. T. Pitman, A. W. Cardoso, C. Locke, A. Swanson, A. Boyer, M. Veldthuis, and L. Fortson, “Identifying animal species in camera trap images using deep learning and citizen science,” *Methods in Ecology and Evolution*, vol. 10, no. 1, pp. 80–91, 2019.
- [180] S. Beery, D. Morris, and S. Yang, “Efficient Pipeline for Camera Trap Image Review,” *arXiv:1907.06772 [cs]*, July 2019. arXiv: 1907.06772.
- [181] M. S. Norouzzadeh, D. Morris, S. Beery, N. Joshi, N. Jojic, and J. Clune, “A deep active learning system for species identification and counting in camera trap images,” *arXiv:1910.09716 [cs, eess, stat]*, Oct. 2019. arXiv: 1910.09716.

- [182] K. Katija, E. Orenstein, B. Schlining, L. Lundsten, K. Barnard, G. Sainz, O. Boulais, M. Cromwell, E. Butler, B. Woodward, and K. L. C. Bell, “FathomNet: A global image database for enabling artificial intelligence in the ocean,” *Scientific Reports*, vol. 12, p. 15914, Sept. 2022.
- [183] R. Bogucki, M. Cygan, C. B. Khan, M. Klimek, J. K. Milczek, and M. Mucha, “Applying deep learning to right whale photo identification,” *Conservation Biology*, vol. 33, no. 3, pp. 676–684, 2019.
- [184] P. C. Gray, K. C. Bierlich, S. A. Mantell, A. S. Friedlaender, J. A. Goldbogen, and D. W. Johnston, “Drones and convolutional neural networks facilitate automated and accurate cetacean species identification and photogrammetry,” *Methods in Ecology and Evolution*, vol. 10, no. 9, pp. 1490–1500, 2019.
- [185] C. Centellegher, L. Carraro, J. Gonzalvo, M. Rosso, E. Esposti, C. Gili, M. Bonato, D. Pedrotti, B. Cardazzo, M. Povinelli, and S. Mazzariol, “The use of Unmanned Aerial Vehicles (UAVs) to sample the blow microbiome of small cetaceans,” *PLOS ONE*, vol. 15, p. e0235537, July 2020. Publisher: Public Library of Science.
- [186] L. Fiori, E. Martinez, M. B. Orams, and B. Bolland, “Using Unmanned Aerial Vehicles (UAVs) to assess humpback whale behavioral responses to swim-with interactions in Vava’u, Kingdom of Tonga,” *Journal of Sustainable Tourism*, vol. 28, pp. 1743–1761, Nov. 2020.
- [187] A. B. Giles, P. A. Butcher, A. P. Colefax, D. E. Pagendam, M. Mayjor, and B. P. Kelaher, “Responses of bottlenose dolphins (*Tursiops spp.*) to small drones,” *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. n/a, no. n/a, 2020.
- [188] E. Bevan, S. Whiting, T. Tucker, M. Guinea, A. Raith, and R. Douglas, “Measuring behavioral responses of sea turtles, saltwater crocodiles, and crested terns to drone disturbance to define ethical operating thresholds,” *PLOS ONE*, vol. 13, p. e0194460, Mar. 2018. Publisher: Public Library of Science.
- [189] E. A. Ramos, B. Maloney, M. O. Magnasco, and D. Reiss, “Bottlenose Dolphins and Antillean Manatees Respond to Small Multi-Rotor Unmanned Aerial Systems,” *Frontiers in Marine Science*, vol. 5, 2018. Publisher: Frontiers.
- [190] P. Pomeroy, L. O’Connor, and P. Davies, “Assessing use of and reaction to unmanned aerial systems in gray and harbor seals during breeding and molt in the UK1,” *Journal of Unmanned Vehicle Systems*, Sept. 2015. Publisher: NRC Research Press <http://www.nrcresearchpress.com>.
- [191] G. Hillman, N. Kehtarnavaz, B. Wursig, B. Araabi, G. Gailey, D. Weller, S. Mandava, and H. Tagare, ““Finscan”, a computer system for photographic identification of marine animals,” in *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering SocietyJ [Engineering in Medicine and Biology*, vol. 2, pp. 1065–1066 vol.2, Oct. 2002. ISSN: 1094-687X.
- [192] J. D. Adams, T. Speakman, E. Zolman, and L. H. Schwacke, “Automating Image Matching, Cataloging, and Analysis for Photo-Identification Research,” *Aquatic Mammals*, vol. 32, pp. 374–384, Sept. 2006.

- [193] S. A. Hale, “Unsupervised Threshold for Automatic Extraction of Dolphin Dorsal Fin Outlines from Digital Photographs in DARWIN (Digital Analysis and Recognition of Whale Images on a Network),” *arXiv:1202.4107 [cs]*, Feb. 2012. arXiv: 1202.4107.
- [194] E. M. Keen, J. Wren, O’Mahony, and J. Wray, “catRlog: a photo-identification project management system based in R,” *Mammalian Biology*, Aug. 2021.
- [195] H. J. Weideman, Z. M. Jablons, J. Holmberg, K. Flynn, J. Calambokidis, R. B. Tyson, J. B. Allen, R. S. Wells, K. Hupman, K. Urian, and C. V. Stewart, “Integral Curvature Representation and Matching Algorithms for Identification of Dolphins and Whales,” *arXiv:1708.07785 [cs]*, Aug. 2017. arXiv: 1708.07785.
- [196] J. Karnowski, E. Hutchins, and C. Johnson, “Dolphin Detection and Tracking,” in *2015 IEEE Winter Applications and Computer Vision Workshops*, (Waikoloa, HI, USA), pp. 51–56, IEEE, Jan. 2015.
- [197] Y. Quiñonez, O. Zatarain, C. Lizarraga, and J. Peraza, “Using Convolutional Neural Networks to Recognition of Dolphin Images,” in *Trends and Applications in Software Engineering* (J. Mejia, M. Muñoz, A. Rocha, A. Peña, and M. Pérez-Cisneros, eds.), pp. 236–245, Springer International Publishing, 2019.
- [198] E. Morteo, A. Rocha-Olivares, R. Morteo, and D. W. Weller, “Phenotypic variation in dorsal fin morphology of coastal bottlenose dolphins (*Tursiops truncatus*) off Mexico,” *PeerJ*, vol. 5, p. e3415, June 2017. Publisher: PeerJ Inc.
- [199] S. Bouma, M. D. Pawley, K. Hupman, and A. Gilman, “Individual Common Dolphin Identification Via Metric Embedding Learning,” in *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1–6, Nov. 2018. ISSN: 2151-2191.
- [200] Y.-C. Lee, H.-W. Hsu, J.-J. Ding, W. Hou, L.-S. Chou, and R. Y. Chang, “Backbone Alignment and Cascade Tiny Object Detecting Techniques for Dolphin Detection and Classification,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. advpub, 2020.
- [201] J. W. Thompson, V. H. Zero, L. H. Schwacke, T. R. Speakman, B. M. Quigley, J. S. Morey, and T. L. McDonald, “finFindR: Automated recognition and identification of marine mammal dorsal fins using residual convolutional neural networks,” *Marine Mammal Science*, vol. 38, no. 1, pp. 139–150, 2022.
- [202] G. University, “Is That ‘Jimmy Carter’ or ‘Barbara Bush?’ Google Designs, Professor Uses Artificial Intelligence to Track Wildlife,” Oct. 2018.
- [203] R. Maglietta, V. Renò, G. Cipriano, C. Fanizza, A. Milella, E. Stella, and R. Carlucci, “DolFin: an innovative digital platform for studying Risso’s dolphins in the Northern Ionian Sea (North-eastern Central Mediterranean),” *Scientific Reports*, vol. 8, p. 17185, Nov. 2018. Number: 1 Publisher: Nature Publishing Group.
- [204] P. Kulits, J. Wall, A. Bedetti, M. Henley, and S. Beery, “ElephantBook: A Semi-Automated Human-in-the-Loop System for Elephant Re-Identification,” *arXiv:2106.15083 [cs]*, June 2021. arXiv: 2106.15083.

- [205] A. Bedetti, C. Greyling, B. Paul, J. Blondeau, A. Clark, H. Malin, J. Horne, R. Makukule, J. Wilmot, T. Eggeling, J. Kern, and M. Henley, “System for Elephant Ear-pattern Knowledge (SEEK) to identify individual African elephants,” *Pachyderm*, vol. 61, pp. 63–77, Oct. 2020.
- [206] D. Weller, *Global and regional variation in the biology and behavior of bottlenose dolphins*. PhD thesis, Texas A&M University, College Station, 1998.
- [207] J. H. Ward, “Hierarchical Grouping to Optimize an Objective Function,” *Journal of the American Statistical Association*, vol. 58, pp. 236–244, Mar. 1963.
- [208] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *arXiv:1505.04597 [cs]*, May 2015. arXiv: 1505.04597.
- [209] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, June 2008.
- [210] V. Renò, G. Dimauro, G. Labate, E. Stella, C. Fanizza, G. Cipriano, R. Carlucci, and R. Maglietta, “A SIFT-based software system for the photo-identification of the Risso’s dolphin,” *Ecological Informatics*, vol. 50, pp. 95–101, Mar. 2019.
- [211] Kaggle, “Humpback Whale Identification Challenge,” 2018.
- [212] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” *arXiv:1801.07698 [cs]*, Feb. 2019. arXiv: 1801.07698.
- [213] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the Gap to Human-Level Performance in Face Verification,” 2014.
- [214] T. Y. Berger-Wolf, D. I. Rubenstein, C. V. Stewart, J. A. Holmberg, J. Parham, S. Menon, J. Crall, J. Van Oast, E. Kiciman, and L. Joppa, “Wildbook: Crowdsourcing, computer vision, and data science for conservation,” *arXiv:1710.08880 [cs]*, Oct. 2017. arXiv: 1710.08880.
- [215] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 vol.2, Sept. 1999.
- [216] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, “The inaturalist species classification and detection dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8769–8778, 2018.
- [217] A. Dutta and A. Zisserman, “The VIA Annotation Software for Images, Audio and Video,” in *Proceedings of the 27th ACM International Conference on Multimedia*, (Nice France), pp. 2276–2279, ACM, Oct. 2019.
- [218] Tzutalin, “LabelImg,” Jan. 2021. original-date: 2015-09-17T01:33:59Z.
- [219] C. Nita and M. Vandewal, “CNN-based object detection and segmentation for maritime domain awareness,” in *Artificial Intelligence and Machine Learning in Defense Applications II* (J. Dijk, ed.), (Online Only, United Kingdom), p. 4, SPIE, Sept. 2020.

- [220] L. Yang, M. Sharpe, A. J. Temple, N. Jiddawi, X. Xu, and P. Berggren, “Description and classification of echolocation clicks of Indian Ocean humpback (*Sousa plumbea*) and Indo-Pacific bottlenose (*Tursiops aduncus*) dolphins from Menai Bay, Zanzibar, East Africa,” *PLOS ONE*, vol. 15, p. e0230319, Mar. 2020.
- [221] A. J. Temple, S. M. Stead, N. Jiddawi, N. Wambiji, N. K. Dulvy, E. Barrowclift, and P. Berggren, “Life-history, exploitation and extinction risk of the data-poor Baraka’s whipray (*Maculabatis ambigua*) in small-scale tropical fisheries,” *Journal of Fish Biology*, vol. 97, no. 3, pp. 708–719, 2020.
- [222] A. J. Temple, N. Wambiji, C. N. Poonian, N. Jiddawi, S. M. Stead, J. J. Kiszka, and P. Berggren, “Marine megafauna catch in southwestern Indian Ocean small-scale fisheries from landings data,” *Biological Conservation*, vol. 230, pp. 113–121, Feb. 2019.
- [223] A. J. Temple, J. J. Kiszka, S. M. Stead, N. Wambiji, A. Brito, C. N. S. Poonian, O. A. Amir, N. Jiddawi, S. T. Fennessy, S. Pérez-Jorge, and P. Berggren, “Marine megafauna interactions with small-scale fisheries in the southwestern Indian Ocean: a review of status and challenges for research and management,” *Reviews in Fish Biology and Fisheries*, vol. 28, pp. 89–115, Mar. 2018.
- [224] S. Weigmann, O. Gon, R. H. Leeney, E. Barrowclift, P. Berggren, N. Jiddawi, and A. J. Temple, “Revision of the sixgill sawsharks, genus *Pliotrema* (Chondrichthyes, Pristiophoriformes), with descriptions of two new species and a redescription of *P. warreni* Regan,” *PLOS ONE*, vol. 15, p. e0228791, Mar. 2020.
- [225] E. Barrowclift, A. Temple, S. Stead, N. Jiddawi, and P. Berggren, “Social, economic and trade characteristics of the elasmobranch fishery on Unguja Island, Zanzibar, East Africa,” *Marine Policy*, vol. 83, pp. 128–136, Sept. 2017.
- [226] A. J. Temple, E. Westmerland, and P. Berggren, “By-catch risk for toothed whales in global small-scale fisheries,” *Fish and Fisheries*, vol. 22, no. 6, pp. 1155–1159, 2021.
- [227] L. Yang, X. Xu, P. Zhang, J. Han, B. Li, and P. Berggren, “Classification of underwater vocalizations of wild spotted seals (*Phoca largha*) in Liaodong Bay, China,” *The Journal of the Acoustical Society of America*, vol. 141, pp. 2256–2262, Mar. 2017.
- [228] L. Yang, X. Xu, and P. Berggren, “Influence of ice concentration and thickness on under-ice ambient noise levels in shallow coastal waters of Liaodong Bay, China,” *Estuarine, Coastal and Shelf Science*, vol. 266, p. 107706, Mar. 2022.
- [229] L. Yang, M. Sharpe, A. J. Temple, and P. Berggren, “Characterization and comparison of echolocation clicks of white-beaked dolphins (*Lagenorhynchus albirostris*) off the Northumberland coast, UK,” *The Journal of the Acoustical Society of America*, vol. 149, pp. 1498–1506, Mar. 2021.
- [230] “Marine and Coastal Access Act,” Nov. 2009.
- [231] F. Stephenson, N. V. C. Polunin, A. C. Mill, C. Scott, P. Lightfoot, and C. Fitzsimmons, “Spatial and temporal changes in pot-fishing effort and habitat use,” *ICES Journal of Marine Science*, vol. 74, pp. 2201–2212, Oct. 2017.

- [232] G. Araujo, S. Snow, C. L. So, J. Labaja, R. Murray, A. Colucci, and A. Ponzo, “Population structure, residency patterns and movements of whale sharks in Southern Leyte, Philippines: results from dedicated photo-ID and citizen science,” *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 27, no. 1, pp. 237–252, 2017.
- [233] J. J. Currie, S. H. Stack, and G. D. Kaufman, “Conservation and Education Through Ecotourism: Using Citizen Science to Monitor Cetaceans in the Four-Island Region of Maui, Hawaii,” *Tourism in Marine Environments*, vol. 13, pp. 65–71, Aug. 2018.
- [234] A. O. Armstrong, A. J. Armstrong, M. B. Bennett, A. J. Richardson, K. A. Townsend, and C. L. Dudgeon, “Photographic identification and citizen science combine to reveal long distance movements of individual reef manta rays *Mobula alfredi* along Australia’s east coast,” *Marine Biodiversity Records*, vol. 12, p. 14, Dec. 2019.
- [235] G. Araujo, A. Agustines, B. Tracey, S. Snow, J. Labaja, and A. Ponzo, “Photo-ID and telemetry highlight a global whale shark hotspot in Palawan, Philippines,” *Scientific Reports*, vol. 9, p. 17209, Dec. 2019.
- [236] W. M. Society, “The Beaufort Scale of Wind Force:(Technical and Operational Aspects),” vol. 3, 1970.
- [237] J. Mann, “Behavioral Sampling Methods for Cetaceans: A Review and Critique,” *Marine Mammal Science*, vol. 15, no. 1, pp. 102–122, 1999.
- [238] B. Bessesen, L. Oviedo, L. Burdett Hart, D. Herra-Miranda, J. Pacheco-Polanco, L. Baker, G. Saborío-Rodríguez, L. Bermúdez-Villapol, and A. Acevedo-Gutiérrez, “Lacaziosis-like disease among bottlenose dolphins *Tursiops truncatus* photographed in Golfo Dulce, Costa Rica,” *Diseases of Aquatic Organisms*, vol. 107, pp. 173–180, Jan. 2014.
- [239] M. A. Silva, L. Steiner, I. Cascão, M. J. Cruz, R. Prieto, T. Cole, P. K. Hamilton, and M. Baumgartner, “Winter sighting of a known western North Atlantic right whale in the Azores,” p. 6, 2012.
- [240] R. A. Smolker, A. F. Richards, R. C. Connor, and J. W. Pepper, “Sex Differences in Patterns of Association Among Indian Ocean Bottlenose Dolphins,” *Behaviour*, vol. 123, pp. 38–69, Jan. 1992. Publisher: Brill.
- [241] S. H. Shane, R. S. Wells, and B. Wursig, “ECOLOGY, BEHAVIOR AND SOCIAL ORGANIZATION OF THE BOTTLENOSE DOLPHIN: A REVIEW,” *MARINE MAMMAL SCIENCE*, vol. 2, no. 1, p. 30, 1986.
- [242] S. Beery and E. Bondi, “Can poachers find animals from public camera trap images?,” *arXiv:2106.11236 [cs, q-bio]*, June 2021. arXiv: 2106.11236.
- [243] J. Kay and M. Merrifield, “The Fishnet Open Images Database: A Dataset for Fish Detection and Fine-Grained Categorization in Fisheries,” *arXiv:2106.09178 [cs]*, June 2021. arXiv: 2106.09178.
- [244] O. Boulais, S. Y. Alaba, J. E. Ball, M. Campbell, A. T. Iftekhar, R. Moorehead, and J. Primrose, “SEAMAPD21: a large-scale reef sh dataset for ne-grained categorization,” p. 7, 2021.

- [245] J. Gao, T. Burghardt, W. Andrew, A. W. Dowsey, and N. W. Campbell, “Towards Self-Supervision for Video Identification of Individual Holstein-Friesian Cattle: The Cows2021 Dataset,” *arXiv:2105.01938 [cs]*, May 2021. arXiv: 2105.01938.
- [246] A. Shirke, A. Saifuddin, A. Luthra, J. Li, T. Williams, X. Hu, A. Kotnana, O. Kocabalkanli, N. Ahuja, A. Green-Miller, I. Condotta, R. N. Dilger, and M. Caesar, “Tracking Grow-Finish Pigs Across Large Pens Using Multiple Cameras,” *arXiv:2111.10971 [cs]*, Nov. 2021. arXiv: 2111.10971.
- [247] M. Timm, S. Maji, and T. Fuller, “Large-Scale Ecological Analyses of Animals in the Wild Using Computer Vision,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (Salt Lake City, UT), pp. 1977–19772, IEEE, June 2018.
- [248] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Honolulu, HI), pp. 3296–3297, IEEE, July 2017.
- [249] C. Rother, V. Kolmogorov, and A. Blake, ““GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts,” *ACM transactions on graphics*, vol. 23, no. 3, pp. 309–314, 2004.
- [250] A. O. Vuola, S. U. Akram, and J. Kannala, “Mask-RCNN and U-net Ensembled for Nuclei Segmentation,” *arXiv:1901.10170 [cs]*, Jan. 2019. arXiv: 1901.10170.
- [251] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft COCO: Common Objects in Context,” *arXiv:1405.0312 [cs]*, May 2014. arXiv: 1405.0312.
- [252] Y. Yu, K. Zhang, L. Yang, and D. Zhang, “Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN,” *Computers and Electronics in Agriculture*, vol. 163, p. 104846, Aug. 2019.
- [253] V. Couteaux, S. Si-Mohamed, O. Nempong, T. Lefevre, A. Popoff, G. Pizaine, N. Vilain, I. Bloch, A. Cotten, and L. Boussel, “Automatic knee meniscus tear detection and orientation classification with Mask-RCNN - ScienceDirect,” *Diagnostic and Interventional Imaging*, vol. 100, no. 4, pp. 235–242, 2019.
- [254] H. Fujita, M. Itagaki, K. Ichikawa, Y. K. Hooi, K. Kawahara, and A. Sarlan, “Fine-tuned Surface Object Detection Applying Pre-trained Mask R-CNN Models,” in *2020 International Conference on Computational Intelligence (ICCI)*, pp. 17–22, Oct. 2020.
- [255] O. Pantazis, G. J. Brostow, K. E. Jones, and O. M. Aodha, “Focus on the Positives: Self-Supervised Learning for Biodiversity Monitoring,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Montreal, QC, Canada), pp. 10563–10572, IEEE, Oct. 2021.
- [256] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random Erasing Data Augmentation,” *arXiv:1708.04896 [cs]*, Nov. 2017. arXiv: 1708.04896.

- [257] A. Atapour-Abarghouei, S. Bonner, and A. S. McGough, “A Kings Ransom for Encryption: Ransomware Classification using Augmented One-Shot Learning and Bayesian Approximation,” *arXiv:1908.06750 [cs]*, Aug. 2019. arXiv: 1908.06750.
- [258] A. Karpathy, “A peek at trends in machine learning,” 2017.
- [259] J. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” p. 39, 2011.
- [260] A. Krogh and J. A. Hertz, “A Simple Weight Decay Can Improve Generalization,” p. 8, 1991.
- [261] P. Berggren, O. A. Amir, A. Guissamulo, N. S. Jiddawi, Z. Ngazy, E. Stensland, A. Särnblad, and V. G. Cockcroft, “Sustainable Dolphin Tourism in East Africa,” *WIOMSA Book Series*, 2007. Publisher: Newcastle University.
- [262] F. Christiansen, D. Lusseau, E. Stensland, and P. Berggren, “Effects of tourist boats on the behaviour of Indo-Pacific bottlenose dolphins off the south coast of Zanzibar,” *Endangered Species Research*, vol. 11, pp. 91–99, Mar. 2010.
- [263] Y. Gal, “Uncertainty in Deep Learning,” p. 174, 2016.
- [264] Q. Wang, W. Min, D. He, S. Zou, T. Huang, Y. Zhang, and R. Liu, “Discriminative fine-grained network for vehicle re-identification using two-stage re-ranking,” *Science China Information Sciences*, vol. 63, p. 212102, Nov. 2020.
- [265] V. Vetrova, S. Coup, E. Frank, and M. J. Cree, “Hidden Features: Experiments with Feature Transfer for Fine-Grained Multi-Class and One-Class Image Categorization,” in *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, (Auckland, New Zealand), pp. 1–6, IEEE, Nov. 2018.
- [266] V. M. Araújo, A. S. Britto Jr., L. S. Oliveira, and A. L. Koerich, “Two-view fine-grained classification of plant species,” *Neurocomputing*, vol. 467, pp. 427–441, Jan. 2022.
- [267] O. Brookes, M. Mirmehdi, H. Kühl, and T. Burghardt, “Triple-stream Deep Metric Learning of Great Ape Behavioural Actions,” Jan. 2023. arXiv:2301.02642 [cs].
- [268] D. A. Maharani, C. Machbub, P. H. Rusmin, and L. Yulianti, “Improving the Capability of Real-Time Face Masked Recognition using Cosine Distance,” in *2020 6th International Conference on Interactive Digital Media (ICIDM)*, pp. 1–6, Dec. 2020.
- [269] C. Pan, J. Huang, J. Hao, and J. Gong, “Towards zero-shot learning generalization via a cosine distance loss,” *Neurocomputing*, vol. 381, pp. 167–176, Mar. 2020.
- [270] Y. Feng, H. Wang, H. R. Hu, L. Yu, W. Wang, and S. Wang, “Triplet Distillation For Deep Face Recognition,” in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 808–812, Oct. 2020. ISSN: 2381-8549.
- [271] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, “Learning to rank using gradient descent,” in *Proceedings of the 22nd international conference on Machine learning - ICML '05*, (Bonn, Germany), pp. 89–96, ACM Press, 2005.

- [272] E. Hoffer and N. Ailon, “Deep metric learning using Triplet network,” *arXiv:1412.6622 [cs, stat]*, Dec. 2018. arXiv: 1412.6622.
- [273] A. Hermans, L. Beyer, and B. Leibe, “In Defense of the Triplet Loss for Person Re-Identification,” Nov. 2017. arXiv:1703.07737 [cs].
- [274] O. Moindrot, “Triplet Loss and Online Triplet Mining in TensorFlow,” Mar. 2018.
- [275] L. Yang, P. Luo, C. C. Loy, and X. Tang, “A large-scale car dataset for fine-grained categorization and verification,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Boston, MA, USA), pp. 3973–3981, IEEE, June 2015.
- [276] C.-A. Brust, T. Burghardt, M. Groenenberg, C. Kading, H. S. Kühl, M. L. Manguette, and J. Denzler, “Towards Automated Visual Monitoring of Individual Gorillas in the Wild,” in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2820–2830, Oct. 2017. ISSN: 2473-9944.
- [277] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” July 2019. Number: arXiv:1907.10902 arXiv:1907.10902 [cs, stat].
- [278] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for Hyper-Parameter Optimization,” p. 9, 2011.
- [279] M. L. Battle, A. Atapour-Abarghouei, and A. S. McGough, “Siamese Neural Networks for Skin Cancer Classification and New Class Detection using Clinical and Dermoscopic Image Datasets,” Dec. 2022. arXiv:2212.06130 [cs].
- [280] P. Siva and T. Xiang, “Weakly supervised object detector learning with model drift detection,” in *2011 International Conference on Computer Vision*, pp. 343–350, Nov. 2011. ISSN: 2380-7504.
- [281] A. Suprem, J. Arulraj, C. Pu, and J. Ferreira, “ODIN: Automated Drift Detection and Recovery in Video Analytics,” Sept. 2020. arXiv:2009.05440 [cs, eess].
- [282] P. Nagar, M. Khemka, and C. Arora, “Concept Drift Detection for Multivariate Data Streams and Temporal Segmentation of Daylong Egocentric Videos,” in *Proceedings of the 28th ACM International Conference on Multimedia*, (Seattle WA USA), pp. 1065–1074, ACM, Oct. 2020.
- [283] O. Cobb and A. V. Looveren, “Context-Aware Drift Detection,” in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162, (Baltimore, Maryland, USA), p. 25, PMLR, 2022.
- [284] Z. Birenbaum, H. Do, L. Horstmyer, H. Orff, K. Ingram, and A. Ay, “SEALNET: Facial recognition software for ecological studies of harbor seals,” *Ecology and Evolution*, vol. 12, May 2022.
- [285] M. Tian, S. Yi, H. Li, S. Li, X. Zhang, J. Shi, J. Yan, and X. Wang, “Eliminating Background-bias for Robust Person Re-identification,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (Salt Lake City, UT), pp. 5794–5803, IEEE, June 2018.

- [286] S. Dutta and E. Gros, “Evaluation of the impact of deep learning architectural components selection and dataset size on a medical imaging task,” in *Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications* (J. Zhang and P.-H. Chen, eds.), (Houston, United States), p. 36, SPIE, Mar. 2018.
- [287] H. Sug, “The Effect of Training Set Size for the Performance of Neural Networks of Classification,” vol. 9, no. 11, p. 10, 2010.
- [288] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, “Sampling Matters in Deep Embedding Learning,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, (Venice), pp. 2859–2867, IEEE, Oct. 2017.
- [289] R. Zheng, Z. Luo, and B. Yan, “Exploiting Time-Series Image-to-Image Translation to Expand the Range of Wildlife Habitat Analysis,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 825–832, July 2019.
- [290] X. Yan, I. Misra, A. Gupta, D. Ghadiyaram, and D. Mahajan, “ClusterFit: Improving Generalization of Visual Representations,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Seattle, WA, USA), pp. 6508–6517, IEEE, June 2020.
- [291] E. Creager, J.-H. Jacobsen, and R. Zemel, “Environment Inference for Invariant Learning,” p. 12, 2021.
- [292] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, “Domain Generalization: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022. arXiv:2103.02503 [cs].
- [293] T. Tommasi, N. Patricia, B. Caputa, and T. Tuytelaars, “Domain Generlization: A Survey,” in *Pattern Recognition: 37th German Conference, GCPR 2015*, vol. 9358 of *Lecture Notes in Computer Science*, (Aachen, Germany), Springer International Publishing, 2015.
- [294] D. Cohn, L. Atlas, and R. Ladner, “Improving generalization with active learning,” *Machine Learning*, vol. 15, pp. 201–221, May 1994.
- [295] B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro, “Exploring Generalization in Deep Learning,” p. 10, 2017.
- [296] J. Lee, S. J. Oh, S. Yun, J. Choe, E. Kim, and S. Yoon, “Weakly Supervised Semantic Segmentation Using Out-of-Distribution Data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (New Orleans, LA, USA), p. 10, 2022.
- [297] T. B. Nguyen, P. Van Phu, T.-L. Le, and C. Vo Le, “Background removal for improving saliency-based person re-identification,” in *2016 Eighth International Conference on Knowledge and Systems Engineering (KSE)*, pp. 339–344, Oct. 2016.
- [298] Y. Huang, Q. Wu, J. Xu, and Y. Zhong, “SBSGAN: Suppression of Inter-Domain Background Shift for Person Re-Identification,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Seoul, Korea (South)), pp. 9526–9535, IEEE, Oct. 2019.

- [299] Y. Cheng, “Mean shift, mode seeking, and clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 790–799, Aug. 1995.
- [300] M. Ester, H.-P. Kriegel, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” *KDD-96 Proceedings*, pp. 226–231, 1996.
- [301] I. Melekhov, J. Kannala, and E. Rahtu, “Siamese network features for image matching,” in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 378–383, Dec. 2016.
- [302] J. Boulet, B. Charry, M. M. Kennedy, E. Tissier, R. Fan, M. Marcoux, C. A. Watt, and A. Gagné-Turcotte, “Scaling whale monitoring using deep learning: A human-in-the-loop solution for analyzing aerial datasets,” *Frontiers in Marine Science*, vol. 10, p. 1099479, Mar. 2023.
- [303] N. M. Müller and K. Markert, “Identifying Mislabeled Instances in Classification Datasets,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, July 2019. ISSN: 2161-4407.
- [304] R. Ekambaram, D. B. Goldgof, and L. O. Hall, “Finding label noise examples in large scale datasets,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2420–2424, Oct. 2017.
- [305] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.
- [306] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Internal Representations by Error Propagation,” technical Report, CALIFORNIA UNIV SAN DIEGO LA JOLLA INST FOR COGNITIVE SCIENCE, 1985.
- [307] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language Models are Few-Shot Learners,” July 2020. arXiv:2005.14165 [cs].
- [308] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” May 2019. arXiv:1810.04805 [cs].
- [309] N. Ng, K. Yee, A. Baevski, M. Ott, M. Auli, and S. Edunov, “Facebook FAIR’s WMT19 News Translation Task Submission,” July 2019. arXiv:1907.06616 [cs].
- [310] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Fun-to-wicz, J. Davison, S. Shleifer, and R. Louf, “Transformers: State-of-the-Art Natural Language Processing,” p. 8, 2020.

- [311] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” June 2021. arXiv:2010.11929 [cs].
- [312] Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, and W. Liu, “You Only Look at One Sequence: Rethinking Transformer in Vision through Object Detection,” Oct. 2021. arXiv:2106.00666 [cs].
- [313] Y. Li, H. Mao, R. Girshick, and K. He, “Exploring Plain Vision Transformer Backbones for Object Detection,” p. 21, 2022.
- [314] J. Hu, L. Cao, Y. Lu, S. Zhang, Y. Wang, K. Li, F. Huang, L. Shao, and R. Ji, “ISTR: End-to-End Instance Segmentation with Transformers,” May 2021. arXiv:2105.00637 [cs].
- [315] T. Prangemeier, C. Reich, and H. Koepll, “Attention-Based Transformers for Instance Segmentation of Cells in Microstructures,” in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 700–707, Dec. 2020.
- [316] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, “End-to-End Video Instance Segmentation with Transformers,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Nashville, TN, USA), pp. 8737–8746, IEEE, June 2021.
- [317] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, “Training Vision Transformers for Image Retrieval,” Feb. 2021. arXiv:2102.05644 [cs].
- [318] W. G. C. Bandara and V. M. Patel, “A Transformer-Based Siamese Network for Change Detection,” in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pp. 207–210, July 2022. ISSN: 2153-7003.
- [319] A. Shabanov, A. Tarasov, and S. Nikolenko, “STIR: Siamese Transformer for Image Retrieval Postprocessing,” Apr. 2023. arXiv:2304.13393 [cs].
- [320] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era,” Aug. 2017. arXiv:1707.02968 [cs].
- [321] H. Bellafkir, M. Vogelbacher, J. Gottwald, M. Mühling, N. Korfhage, P. Lampe, N. Frieß, T. Nauss, and B. Freisleben, “Bat Echolocation Call Detection and Species Recognition by Transformers with Self-attention,” in *Intelligent Systems and Pattern Recognition* (A. Bennour, T. Ensari, Y. Kessentini, and S. Eom, eds.), vol. 1589, pp. 189–203, Cham: Springer International Publishing, 2022. Series Title: Communications in Computer and Information Science.
- [322] Y. Zhou, Y. Yu, and B. Ding, “Towards MLOps: A Case Study of ML Pipeline Platform,” in *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pp. 494–500, Oct. 2020.