Developing Deep Learning Methods for Aquaculture Applications

Alzayat Saleh

A thesis submitted for the degree of
Master of Philosophy
James Cook University

June 2020

Declaration

Except where otherwise indicated, this thesis is my own original work.

Alzayat Saleh
29 June 2020

This research is supported by an
Australian Research Training Program (RTP)
Scholarship.

To Sally, Ahmed, Cady

Acknowledgments

I would like to express my deepest gratitude to my supervisory panel, Dr. Dmitry A. Konovalov for his patience, his encouragements and time. I learned a lot from him especially in the area of computer vision. As well, Prof Hossein Ghodosi for providing me the opportunity to study under his supervision.

I would like to express my deepest gratitude to Prof Trina Myers. She provided me with tremendous support and encouraged me to continue my research when things got difficult.

Exceptional thanks to Dr. Mostafa Rahimi Azghadi and Prof Mohan Jacob for their continued support of me.

I would also like to thank the GRS team here at the JCU for their enthusiastic support. Especially, I am indebted to Dr. Lauretta Grasso for always being there for me and her moral support during a difficult times and also special thanks to Mrs. Jodie Wilson.

I would like to thank my close collaborators, Dr. Issam Laradji and Dr. David Vázquez, from ElementAI, who I've had the distinct pleasure of working with and learning from. I'd like to acknowledge the support from and useful discussions with my fellow research students in the IT group. Special thanks to Timothy Hart, Randy Zhu, Louis Cianciullo, Karl Mohring, Xiaonan LI.

I would also like to extend thanks to the many people who I've had the pleasure of interacting with and learning from, including Prof Marcus sheaves, Dr Mangalam Sankupellay, and Dr Amin Roushdy.

Kevin Bairos-Novak, thanks bro for you support throughout this journey.

Finally, I would like to express my deepest thanks to my family. Especially, Dr Alhamalawy Saleh, Yasser Hamad, and Abdelrahman Mostafa. Sally, Ahmed, and Cady thank you for your love and trust. You have encouraged me the entire way and have always been my biggest fans and strongest supporters. Without you, I could never be where I am today.

This thesis is dedicated to the memory of my father and my mother.

Abstract

Assessing fish habitats is an important step for maintaining sustainable fisheries. Counting and sizing fish in real-time across habitats requires significant human efforts. Reducing the labour cost in maintaining and improving fish productivity could lead to enormous economic impact. In this work, a computer vision framework has been developed that can aid experts in analyzing such fish habitats. Also, an efficient labelling method is developed for training a CNN-based fish-detector on a relatively small number of underwater fish/no-fish images (4,000), combined with 17,000 known negatives such as images with no fish in them or above-water general-domain images. Moreover, a benchmark based on a large image data-set of remote underwater video collected from tropical Australia is presented. The purpose of this benchmark is to facilitate specialized algorithms that can automate the task of fish image analysis. The benchmark dataset consists of approximately 40,000 labelled images representing 20 different fish habitats around Australia. A model that estimates the fish weight directly from its image is also developed. For this model, fish masks were automatically segmented and fit using simple mathematical models to achieve a low mean absolute percent error (MAPE) of 4-10% while using 1,400 test images. This thesis presents a practical and easily reproducible weight-from-image approach and tests the use of a segmentation convolutional neural network (CNN) for estimating fish weight. The methods outlined in this thesis are one step towards the development of valuable practical computer vision applications. The methods described herein will aid in the classification of various fish species habitats, determining the presence or absence of fishes, and quantifying fish weights and sizes. Thus, the thesis represents an important technological step towards better fisheries management, ecosystem management, and fish stock conservation programs.

Contents

Declaration iii

Acknowledgments vii

Abstract ix

List of Abbreviations xxi

Statement of the Contribution by Others xxiii

List of Publications Related to this Thesis xxv

1 Introduction 1

1.1 Overview 1

1.2 Contributions and Outline 4

2 Technical Background 9

2.1 Neural Networks and Backpropagation 9

2.1.1 Activation Functions 10

2.1.2 Bias Node 10

2.1.3 Cost Function 10

2.1.4 Optimization 11

2.1.5 Backpropagation 12

2.2 Convolutional Neural Network 13

2.2.1 Convolutional Layer 13

2.2.2 Pooling Layer 14

2.3 Supervised Learning 14

2.4 Neural Network Workflow 16

2.4.1 Data preparation 16

2.4.2 Data preprocessing 17

2.4.3 Architecture design. 17
2.4.4 Optimization. 18
2.4.5 Hyperparameter optimization. 18
2.4.6 Evaluation. 19

3 Underwater Fish Detection **21**
3.1 Introduction 22
3.2 Related Work 24
3.3 Materials And Methods 30
3.3.1 Standardized Procedure for Dataset Preparation 30
3.3.2 The Dataset 32
3.3.3 The CNN model and training setup 34
3.3.4 General-Domain Image Datasets 34
3.3.5 The Heatmap 35
3.4 Results and Discussion 36
3.4.1 Training Baseline 36
3.4.2 General-Domain Image Datasets Supervision 37
3.4.3 Heatmap Localization 37
3.5 Conclusion 40

4 Benchmark for Analyzing Fish Habitats **41**
4.1 Introduction 42
4.2 Data and Methods 44
4.2.1 Dataset description 44
4.2.2 Deep Learning Methods 48
4.3 Results and Discussion 52
4.3.1 Classification 52
4.3.2 Counting 53
4.3.3 Localization 54
4.3.4 Segmentation 55
4.4 Conclusion 55

5 Mass Estimation of Fish From Images **59**
5.1 Introduction and Related work 60

5.2 Materials and Methods 63
5.2.1 Harvested Barramundi Datasets 63
5.2.2 Segmentation of Fish Surface Area 63
5.3 Results and Discussion 69
5.4 Conclusion 71

6 Automatic Weight Estimation of Fish from Images **73**
6.1 Introduction and Related work 74
6.2 Materials And Methods 78
6.2.1 Harvested Barramundi Datasets 78
6.2.2 Segmentation of Fish Surface Area 79
6.2.3 Training Pipeline 80
6.3 Results and Discussion 82
6.3.1 Estimating weight-from-area by mathematical models . 82
6.3.2 Direct weight-from-image estimation 86
6.3.3 Models predictive performance 88
6.4 Conclusion 90

7 Conclusion **91**
7.1 Summary and contributions 91
7.2 Future Work 94

Bibliography **97**

List of Figures

1.1 visual recognition tasks. Left: a fish in an underwater habitat. Middle: localization of the fish. Right: human segmentation of the fish from the background. 2

1.2 Thesis structure and interconnection of chapters. 5

2.1 **Left:** A drawing of a biological neuron and myelinated axon, with signal flow from inputs at dendrites to outputs at axon terminals. **Right:** A diagram of the biological inspiration behind a single CNN neuron. Inputs x_i interact multiplicatively with the synapses w_i. The cell body accumulates the sum of all inputs and then fires an output signal according to the activation function. If the activation is the sigmoid non-linearity (with output range in [0,1]), then the output can be interpreted as the average firing rate of the neuron. Figure from [1] 14

2.2 Architecture of LeNet-5, an old convolutional neural network for digits recognition. Each layer is a feature map, i.e a set of different neuron convolutions, whose weights are constrained to be identical. [2] 15

3.1 Sample fish-containing video frames from the 20 considered habitats. 25

3.2 Examples of frames from mangroves habitat with: one *Lutjanus argentimaculatus* adult (top row), one *Chaetodon vagabundus* (bottom row), and multiple *Caranx sexfasciatus* juveniles (middle row). Top-left, middle-right and the bottom-row images were all histogram equalized via CLAHE. 29

3.3 Locations of video clip capture via unbaited remote underwater video (UBRUV) along the near-shore island chain of the Great Barrier Reef (Queensland, Australia). 30
3.4 The distribution of all images in the dataset, across 20 different aquatic habitats with the relative proportion of images containing fish ('valid') and those without fish ('empty'). 31
3.5 Typical examples of fish correctly detected and localized by XFishHm (trained as in XFishHmMp), where the left two subfigures are the padded and re-scaled original images in grayscale, the middle two subfigures are the same images overlapped with the prediction heatmaps, and right-most two subfigures being the generated prediction heatmaps. 39

4.1 **A comparison of fish datasets.** (a) image from QUT [3], (b) image from Fish4Knowledge [4] and (c) image from Rockfish [5], (d) image from the proposed dataset DeepFish. Other datasets are acquired from constrained environments, whereas DeepFish has more realistic and challenging environments. 43
4.2 **Locations where the DeepFish images were acquired.** DeepFish has been acquired from the Hinchinbrook/Palm Islands region in North Eastern Australia. 45
4.3 DeepFish image samples across 20 different habitats. 46
4.4 **Additional annotations.** (a) original images, (b) images with count and point-level annotations (c) the segmentation masks. . 46
4.5 **Deep learning methods.** Four different deep learning models used for 4 different tasks. These tasks are from top classification, counting, localization, and segmentation. 51
4.6 Qualitative results on counting and localization with Reg-2 and Loc-2. (a) Test images obtained from DeepFish, (b) Prediction results using localization-based loss function. (c) Annotations represent the x and y coordinates of each fish within the images and they are placed around the centroid of the corresponding fish. 54

4.7 Qualitative results for segmentation with Seg-2. (a) Test images obtained from DeepFish, (b) Prediction results using the focal loss. (c) Annotations represent the full segmentation masks of the corresponding fish. 55
4.8 Thesis structure and interconnection of chapters. 57

5.1 (left) Drawing of how fish total length (L) is measured; (right) drawing of fish surface area (S) not including fins or the tailfin. 60
5.2 Examples of images from the BR445 (left images) and BA600 (right) datasets. 64
5.3 Examples of the 200 fish images that were manually-segmented into fish-body and background. 65
5.4 Relation between the measured fish weight (M in g) and the manually-segmented fish body image area (S in cm^2) fitted by: Equation 5.4 as $M = 0.1695 \times S^{1.5}$, $R^2 = 0.9819$, $MARE = 5.13\%$; and Equation 5.3 as $M = 0.1622 \times S^{1.5073}$, $R^2 = 0.9819$, $MARE = 5.06\%$. Higher densities of data points are denoted by lighter coluors. 66
5.5 Example of image augmentation: (left) the augmented image, (centre) the corresponding augmented binary mask of the fish body (without fins), (right) both the image and the superimposed mask. 67
5.6 Diagram summarizing the full process from training the FAS model to fitting of model results. 70
5.7 Relation between the measured fish weight (M in g) and the automatically-segmented fish-body image area (S in cm^2) as the red line using Equation 5.4: $M = 0.17 \times S^{1.5}$, $R^2 = 0.9804$, $MARE = 5.128\%$; the dotted line is the fitted Equation 5.3, $M = 0.12 \times S^{1.5}$, $R^2 = 0.9808$, $MARE = 4.84\%$. Higher densities of data points are denoted by lighter colours. 71

6.1 Samples of original images from the used datasets: BR445 (a) and (b), BW1400 (c) and (d), BA600 (e) and (f). 78

6.2 The same samples as in Fig. 6.1 converted to grayscale and enhanced by Enhance Local Contrast (CLAHE) [6]. 78
6.3 An example of weight measuring error in the BW1400 dataset: (a-c) the correctly measured reference images with y weight values; (d) the identified recording/measuring error (predicted $p = 751$g); (e) the mask without fins (including tailfin) for the fish in (d); (f) the whole fish mask for the fish in (d); (g) the mask without fins and tail; (h) the whole fish mask. 79
6.4 Relation between the measured fish weight and the automatically segmented fish area for the combined BR445 and BA600 datasets: no-fins (fish masks without fins or tailfins) on top and whole fish (fish masks including fins and tailfins) on the bottom. 85
6.5 Normalized distributions of automatically-segmented mask areas in the BW1400 images 87

7.1 Thesis structure and interconnection of chapters 92

List of Tables

3.1 Confusion Matrix for Model Testing Fold Results using the FD10-Test Dataset 38

4.1 **DeepFish Dataset Statistics.** Number of images annotated for each sub-dataset: FishClf for classification, FishLoc for counting/localization, and FishSeg for semantic segmentation 49
4.2 Counting and Localization results on FishLoc. 53
4.3 Classification results on FishClf. 53
4.4 Segmentation results on FishSeg. 53

6.1 Mass estimation models 83

List of Abbreviations

ADAM Adaptive Moment Estimation
ANN Artificial Neural Networks
CNN Convolutional Neural Networks
IoU Intersection over Union
ML Machine Learning
MSE Mean Squared Error
MAPE Mean Absolute Percentage Error
PCA Principal Component Analysis
ReLU Rectified Linear Unit
ROI Region of Interest
SGD Stochastic Gradient Descent
SVM Support Vector Machine
TP True Positives
FP False Positives
VGG Visual Geometry Group

Statement of the Contribution by Others

- Dr Dmitry A. Konovalov supervised the two projects used in this thesis, offered suggestions, and helped revise the final version of this thesis.
- A/Prof Hossein Ghodosi - offered suggestions, and helped revise the final version of this thesis.
- Dr Issam H. Laradji - offered suggestions on chapter 4, and helped revise the final version.
- Dr David Vazquez - offered suggestions on chapter 4, and helped revise the final version.
- Dr Michael Bradley - collected the dataset used in chapter 3 for 20 habitats from remote coastal marine environments of tropical Australia.
- Prof Marcus Sheaves - supervised dataset collection used in chapter 3.
- Dr Jose A. Domingos - collected the images of Asian seabass (barramundi, *Lates calcarifer*) used in chapter 5 and chapter 6.
- Prof Dean R. Jerry - supervised dataset collection used in chapter 5 and chapter 6.
- Prof Ronald D. White - offered suggestions on chapter 5, and helped revise the final version.
- Dr Simone Marini - offered suggestions on chapter 3, and helped revise the final version.
- Dr Mangalam Sankupellay - offered suggestions on chapter 3, and helped revise the final version.

List of Publications

- Alzayat Saleh, Issam H. Laradji, Dmitry A. Konovalov, Michael Bradley, David Vazquez, and Marcus Sheaves, "Where is the Fish: a Benchmark for Analyzing Fish Habitats" accepted at Nature Scientific Reports journal.

- D. A. Konovalov, A. Saleh, M. Bradley, M. Sankupellay, S. Marini, and M. Sheaves, "Underwater fish detection with weak multi-domain supervision," Institute of Electrical and Electronics Engineers (IEEE), 2019. [Online]. Available: http://dx.doi.org/10.1109/IJCNN.2019.8851907

- D. A. Konovalov, A. Saleh, J. A. Domingos, R. D. White, and D. R. Jerry, "Estimating mass of harvested asian seabass lates calcarifer from images," World Journal of Engineering and Technology, vol. 6, no. 03, p. 15, 2018. [Online]. Available: http://dx.doi.org/10.4236/wjet.2018.63B0

- D. A. Konovalov, A. Saleh, D. B. Efremova, J. A. Domingos, and D. R. Jerry, "Automatic weight estimation of harvested fish from images," Institute of Electrical and Electronics Engineers (IEEE), 2019. [Online]. Available: http://dx.doi.org/10.1109/DICTA47822.2019.8945971

Chapter 1

Introduction

1.1 Overview

Empowering machines to see the world and recognize visual objects around us is one of the primary challenges often addressed using Artificial Intelligence. For humans, it is relatively easy to perform several tasks that involve identifying, localizing and segmenting objects in our perceived worlds, using multiple visual recognition systems while differentiating between different recognized objects. One look at an image is adequate for a human to identify and explain many of its details. For example, a human can easily spot and identify a fish in the first image in Figure 1.1, and can quickly describe it as a single fish in an underwater habitat. Moreover, a human can segment the fish from the background (Figure 1.1, right), identifying where the fish begins and where it ends. However, these processes (e.g., identifying there is a fish, segmenting the fish, classifying the habitat) are much more difficult for a computer to do. However, other tasks such as sizing and weighing the fish in a photo are much easier to do; the majority of the work lies in correctly identifying the location and size of the fish.

Fisheries research often involves deploying remote underwater video stations and compiling thousands of hours of video footage to be analyzed by volunteers or paid workers. While easy to obtain, identifying and measuring fish within videos can take even longer than recordings themselves [7]. Thus, there is a need to address these challenges using computer vision applications. Below, I discussed the two main challenges my thesis will aim to address in the context of underwater cameras used for fisheries management and

conservation: identifying fish, and after identification, weighing and sizing fish.

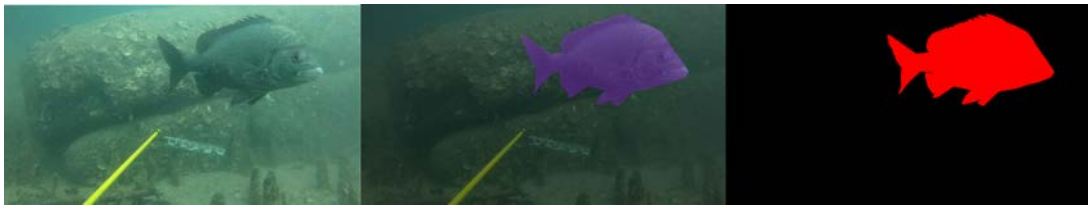

Figure 1.1: visual recognition tasks. Left: a fish in an underwater habitat. Middle: localization of the fish. Right: human segmentation of the fish from the background.

Thesis aims. This thesis has two broad aims, which when deployed in the future, may provide extremely valuable information to fisheries management, ecosystem management and conservation to aid in maintaining the integrity of fish stocks while saving thousands of man-hours that can be better spent on ground operations within fisheries.

- The first aim is to develop practical computer vision applications capable of detecting fishes from various aquatic habitats using a large image dataset. The image database *DeepFish* consists of approximately 40 thousand labelled images representing 20 fish habitats collected from remote coastal marine-environments within tropical climates from around Australia. My thesis uses *DeepFish* to evaluate a variety of deep learning methods that can be grouped according to four main tasks: (1) classification, (2) counting, (3) localization, and (4) segmentation of fishes.

- The second aim of my thesis is to measure harvested fish weights and sizes from their images and develop a practical and easily reproducible weight-from-image approach. This involves using basic statistical models to determine approximate size and weights of fish.

Challenges. Visual recognition and scene comprehension tasks appear natural and simple for us as humans; it is often easy to overlook how difficult these tasks are for a computer to replicate. In the 'mind' of a computer, images are stored as an array of integers representing the brightness at every pixel (i.e. grey-scale image). A real-life image might have millions of pixels, and the computer must convert these patterns of brightness values into higher-level

concepts (e.g. the outline of a fish). The same fish seen under various lighting conditions from another camera perspective, or in a different posture might still represent a “fish”, although the "pattern" of these pixels could change dramatically. In contrast, patterns that have similar low-level representations (e.g. fish scale-like patterns) could also be confused for a different object (e.g., ground debris, aquatic plants, etc.) [8; 9; 10]. The process of segmentation (determining where the fish begins and ends relative to the background, to obtain the fish’s contour) is even more challenging. In a computer, this task is done by assigning a label (integer) to each pixel, which marks if it is part of the ’fish’ or part of the ’background’. Hence, the very natural tasks of visual perception (i.e. describing the image), classifying images (fish or no fish), and segmenting different elements of a single object (the fish contour) in an image requires the annotation of millions of pixels with sequences of integers that share a complex pattern. In contrast to visual recognition and scene understanding tasks, the task of predicting fish weight from its image is not an easy task for humans. However, for computers that easily can recall the size vs. weight relationships derived from thousands of fish images, the task is made relatively easier. The computer vision tasks in this thesis require localizing, detecting and segmenting multiple high-level features in images in order to predict weight, which depends more on the morphological features of the fish. In this thesis, images of harvested fish have been used as inputs to a convolutional neural network (CNN) developed to predict the weight of fish.

Promising progress. Despite the fact that the complexity of fish identification and segmentation tasks, these areas of computer vision have recently experienced rapid progress in terms of accuracy. In particular, state-of-the-art image classification deep CNN models have seen significant improvements in recent times [2] and can now classify thousands of optical objects at accuracies equivalent to an average human, and sometimes even beating them during some specific classification tasks (e.g. classifying dog breeds [11]). Another advancement is object detection and segmentation, which have both improved drastically [12; 13]. Concurrently, these advancements have promoted many real-world applications (examples include self-driving cars, face detection and

recognition, perception in robotics [14]).

Motivation. The motivation behind this work is the development of valuable practical computer vision applications capable of classifying, counting, localizing, and segmenting underwater fishes in various aquatic habitats, and measuring fish weights and sizes to provide valuable information to fisheries management, ecosystem management and conservation to maintain the integrity of fish stocks. Monitoring the abundances and biomass of fish is paramount to monitoring fish population health. Therefore, classifying various underwater habitats, determining the fish species that use them, and analyzing their abundance and size can all give valuable insights regarding the health of the ecological system, and can be used for monitoring environmental changes [8; 15; 10]. Underwater fish species classification can also be used to understand fish distribution patterns and identify seasonal trends in species migrations to provide a deeper insight into the behaviour of a species as a whole [16; 17]. These tasks are part of a large scale project to improve understanding of habitat requirements for tropical coastal fisheries species and is critical to fisheries and conservation management, as they provide (a) knowledge of juvenile habitats that need to be protected, (b) knowledge of the extent and direction of population change, (c) the ability to predict the size of future harvestable fish stocks, and (d) an understanding of the impact of habitat/environmental change on fish recruitment and survival.

1.2 Contributions and Outline

In this thesis, a Deep Learning CNN models have been developed to leverage real-world applications for marine habitats. These neural network architectures can be used to detect the presence of underwater fish species and predict the weight of harvested fish from images of fish, using datasets of either images or videos. The content of chapters 3,4,5 and 6 are based on the following papers: Where is the Fish: a Benchmark for Analyzing Fish Habitats, Underwater fish detection with weak multi-domain supervision [18], Estimating mass of harvested asian seabass lates calcarifer from images [19], Automatic weight estimation of harvested fish from images[20]. Figure 1.2 provides the

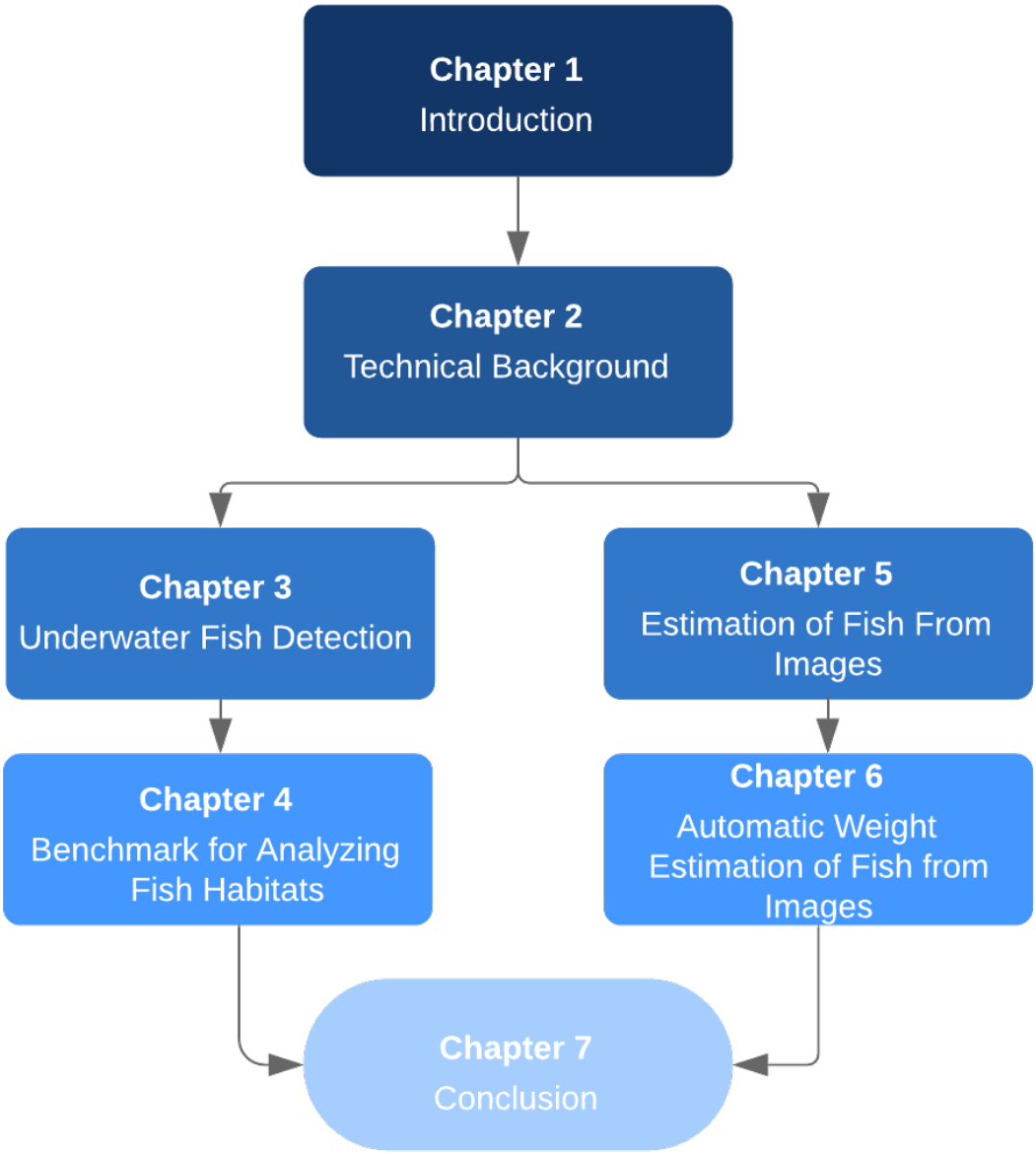

Figure 1.2: Thesis structure and interconnection of chapters.

thesis structure and the interconnections among chapters.

The content of this thesis was written by myself, Alzayat Saleh. My primary supervisor Dmitry A. Konovalov supervised the project, offered suggestions, and helped revise drafts and the final version of the thesis. The use of 'we' throughout this thesis is to acknowledge the contribution of others noted on page xxiii: the Statement of the Contribution by Others.

Chapter 2 presents brief technical background concepts on neural networks and deep learning relating to this thesis.

Chapter 3 developed a labelling-efficient method of training a CNN-based fish-detector (the *Xception* [21] CNN was used as the base) on relatively small numbers (4,000) of project-domain underwater fish/no-fish images from 20 different habitats. Additionally, 17,000 known negative (i.e. missing fish) general-domain (VOC2012) above-water images were used. Two publicly available fish-domain datasets supplied an additional 27,000 above-water and underwater positive/fish images. By using multi-domain collection of images, the trained *Xception*-based binary (fish/not-fish) classifier achieved 0.17% false-positives and 0.61% false-negatives on the project's 20 thousand negative and 16 thousand positive holdout test images.

Chapter 4 presents a benchmark called *DeepFish* that is based on a large image dataset of remote underwater video collected from remote coastal marine-environments of tropical Australia. The dataset consists of approximately 40 thousand labelled images representing 20 fish habitats across Australia. As baselines, I also evaluate a variety of deep learning methods across four tasks: (1) classification, (2) counting, (3) localization, and (4) segmentation of fishes.

Chapter 5 developed a Segmentation Convolutional Neural Network trained on 200 images which were used to automatically segment fish-body from the background in all of this study's 1,072 digital images of Asian seabass (barramundi, *Lates calcarifer*). The automatically-extracted fish body areas were used to predict the corresponding manually-measured weights for each fish, yielding highly accurate single- and two-factor mass-from-area estimation models.

Chapter 6 developed a practical and easily reproducible approach to estimating weight from an image, and details how a standard "*off-the-shelf*"

segmentation CNN such as *LinkNet-34* [22] could be trained efficiently using: (i) only 100-200 training image-mask pairs; (ii) a linear learning rate annealing schedule; and (iii) reduced learning rate for the ImageNet-trained encoder (ResNet-34). With- or without-fins fish masks were automatically segmented and fit by one-factor and two-factor weight-from-area models. Then they were fit using 1,072 area-weight pairs from two locations, where area values were extracted from the automatically segmented masks. When applied to 1,400 test images (from a third location), the one-factor whole-fish mask model achieved the best mean absolute percentage error (MAPE), MAPE $= 4.36\%$. Direct weight-from-image regression CNNs were also trained, where the no-fins based CNN performed best on the test images, with MAPE $= 4.28\%$.

Finally, **Chapter 7** concludes the thesis, where the remaining challenges have been identified, and potential future work have been discussed.

Chapter 2

Technical Background

This chapter provides a brief technical background on concepts such as neural networks and deep machine learning. The following chapters involve work that was developed for a larger project on processing underwater images, which involves real-world applications for marine habitats. The details of each part are discussed herein.

2.1 Neural Networks and Backpropagation

A 'Neural Network' is a computer system originally conceived by mimicking actual cerebral neural networks that make up brain's grey matter. A computer's neural network, a.k.a. an artificial neural network, "learns" to do a specific task by using a large amount of data, usually through network training that does not involve any task-specific rules. A neural network is constructed from three types of layers: an input layer, hidden or latent layers, and an output layer (e.g. see Figure 2.2). Where in the Input layer, input features are acceptable in this layer. It provides information to the network from outside, no calculation is made on that layer and nodes pass the information(s) on to the hidden layer. In the Hidden Layer, this layer of nodes is not exposed to the outside world, but it is the abstraction that comes with a neural network. Hidden layer performs all kinds of calculations on the functionality entered via the input layer and transfers the result to the output layer. Finally the Output Layer, This layer brings the information that the network has learned into the outside world. In this section, I will discuss some of the main components of artificial neural networks.

2.1.1 Activation Functions

The activation function in a neural network defines whether a given node is "activated" or not based on the weighted sum of input features. The sigmoid function is one of the most commonly used activation functions today. The sigmoid function is defined as:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

where S(x) is the sigmoid function output that will be used as the input for the following node and x is the weighted sum of input features from the previous layer. The sigmoid function is non-linear, and ranges in value between 0 and 1. This simplicity of interpretation (between 0 and 1) makes it fairly popular, and because the function enables the nodes to take any values between 0 and 1 because of these properties. In the output layer, for multiple output classes, the probability of "activating" each output class will be different. Then, the function with the largest "activation" value is selected, thus improving the network's ability to classify the image.

2.1.2 Bias Node

Another important component in successful neural networks are the "bias" nodes. A bias value enables the activation function to be shifted to the left or to the right and aids the model to find a better fit faster. In other words, bias nodes are added in order to increase the model's flexibility (see Figure 2.1). In particular, when all input features equal to 0, the network can adjust to the data and decrease the distance between the fitted values in other data spaces.

2.1.3 Cost Function

Cost functions measure the performance of a data-based machine learning model. The cost function is important to consider, as it measures and presents error in the form of a single real number between predicted values and expected values. In other words, it maps a value of one or more of these variables into a real number which represents an event's cost. As an example, the cost

function for linear regression is defined as:

$$J = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y)^2 \tag{2.2}$$

, where m is the number of training example, $\hat{y}$ is the predicted value of the model, and y is the true value of the inputs of the training data.

For classification tasks, the loss function L is generally a cross-entropy loss function. **Cross-entropy loss** measures the performance of a classification model with a probability value ranging from 0 to 1. The loss of cross-entropy functions will increase as the predicted probability differs from the ground truth. Another classification loss is **Hinge Loss**. In Hinge Loss, the correct category score should by some safety margin be higher than the sum of values for all incorrect categories. The regression task is a predictive modelling technique which examines the causal effect relationship between the variables, independent (ground-truth) and dependent (predicted) variables. There are various regression techniques that can be applied to predict dependent variables, both for linear and non-linear responses (e.g. linear regression, polynomial regression), and for binary outcomes (logistic regression).

2.1.4 Optimization

In a supervised learning problem, the learning task can be reduced to an optimization problem in the form of $\theta^* = \arg\min_\theta g(\theta)$, where θ is a parameter vector and g usually combines a regularization penalty and the average loss for all examples.

Given a function $f(x)$, an 'optimization' method helps in either minimizing or maximizing the value of $f(x)$. In deep learning, the optimization methods is used to train the neural network by optimizing the error function E. The error function is defined as:

$$E(W, b) = \sum_{i=1}^{m} L\left(\hat{y}_i, y_i\right) \tag{2.3}$$

where W and b are the weights and biases of the network, respectively. The value of the error function E is thus the sum of the mean squared loss L be-

tween the predicted value $\hat{y}$ and true value y. The value of $\hat{y}$ is obtained during the forward propagation step and makes use of the previously-mentioned weights and biases of the network. Optimization minimizes the value of error function E by updating the values of the trainable parameters W and b.

Optimization methods fall into two main categories [23]: **1) First-order optimization**: this method minimizes a loss function E by using its gradient slopes for the parameters. The most commonly used method is *Gradient Descent* [23]. The gradient is optimized by calculating a matrix of partial derivatives (computed using backpropagation - as detailed below), which provides the slope of g simultaneously at each dimension of θ. The gradient enables first-order approximation via a Taylor series of g, hence the name "first-order optimization". Therefore, the gradient is used to determine the next direction to search for Global Optima. In order to enhance θ and reach a lower g, a small quantity is subtracted from θ in the optimal direction (since the gradient provides the direction of the rise and conversely the descent in g), such that the global optimum is eventually reached and g is minimized. **2) Second-order optimization**: this optimization adopts the second-order derivative (*Hessian* [24]) to minimize the loss function. The Hessian is a square matrix of second-order partial derivatives of a scalar-valued function, or scalar field. The Hessian describes the local curvature of a function of many variables [24]. The Hessian specifies if the first derivative is decreasing or increasing, which implies the function's curvature and provides a quadratic surface that touches the curvature of the *Error Surface*. As the second derivative is computationally expensive, second-order optimizations are not used as often as first-order optimization methods.

2.1.5 Backpropagation

The backpropagation algorithm mentioned previously refers to the process of efficiently computing gradients of the functions using the inputs from the last layer. Put simply, backpropagation involves the recursive use of the chain rule formula to calculate partial derivatives. Using the function g, the data (x_i, y_i) as input, and the parameter θ. Backpropagation probably is the most important building block in the neural network

To address this, in applications of the neural networks, the CNN initially does a 'forward pass' by taking a batch of the dataset $\{(x_i, y_i)\}_{i=1}^{m}$ and the current parameter θ. A forward pass is when the network is used to calculate all the intermediate values, by passing the data input to the hidden layer to process according to the activation function, then passing the values on to the next layer, then do a "forward" passing. The network then calculates the intermediate values for the batch of the dataset and stores them for later. The cost function g at the end is then displayed using a "computational graph", (a computational graph is a directed graph where the nodes correspond to operations or variables). After the forwards pass, a backwards pass is done through all intermediate stages, and these derivatives are then "chained" by the local gradients. For each backwards pass, the matrix is multiplied by the next Jacobian matrix $\frac{\partial y_i}{\partial x_i}$ in the full product.

2.2 Convolutional Neural Network

A convolutional neural network (CNN) [2] is a type of feed-forward artificial neural network, specifically designed for dealing with datasets that have some spatial or topological features (e.g. images, videos), where each of the neurons are placed in such a manner that they overlap and thus react to multiple spots in the visual field, and each spot has redundant neurons connecting. CNNs are broadly designed after the neuronal architecture of the human cortex but on much smaller scales [25]. A CNN neuron is a simple mathematical design of the human brain's neuron that is utilized to transform nonlinear relationships between inputs and outputs in parallel (see Figure 2.1). There are two primary layers in a CNN: convolutional layers and pooling layers.

2.2.1 Convolutional Layer

In this layer, the convolutional processes (i.e., overlap among neuron inputs) used on limited fields to avoid learning billions of weights (parameters) in the case of a fully connected layer. This excessive computation is avoided because of both weight-sharing via the convolutional layers combined with filters for the corresponding feature map. Further, parameter sharing is a way

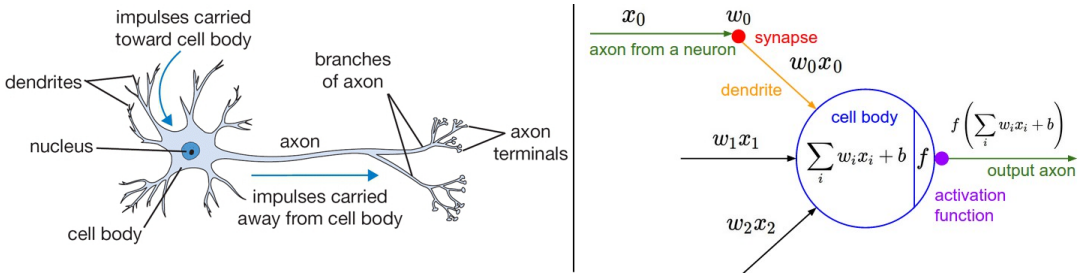

Figure 2.1: Left: A drawing of a biological neuron and myelinated axon, with signal flow from inputs at dendrites to outputs at axon terminals. **Right:** A diagram of the biological inspiration behind a single CNN neuron. Inputs x_i interact multiplicatively with the synapses w_i. The cell body accumulates the sum of all inputs and then fires an output signal according to the activation function. If the activation is the sigmoid non-linearity (with output range in [0,1]), then the output can be interpreted as the average firing rate of the neuron. Figure from [1]

of controlling for model overfitting [26], in addition to reducing computing memory requirements and enhancing CNN performance [27].

2.2.2 Pooling Layer

Pooling layers are used to further control for overfitting by reducing the amount of the representations with a specified in Max Pooling, think of scaling an image in size, by taking the maximum old pixel of the map to the same new pixel (i.e. with no trainable parameters). Pooling layers are systematically-implemented between convolutional layers in a traditional CNN architecture. The pooling layers work on each channel (activation map) individually, and downsample them spatially.

When combined together, convolutional and pooling layers make up the convolutional network, and together, aid in managing the computational difficulty of CNN architecture (e.g. see Figure 2.2).

2.3 Supervised Learning

Supervised learning is a deep learning task used to enable the computer to quickly learn a function that maps to an input-output object pair, also known as a training example, where each input object is paired with the desired output value. This function uses a set of training examples based

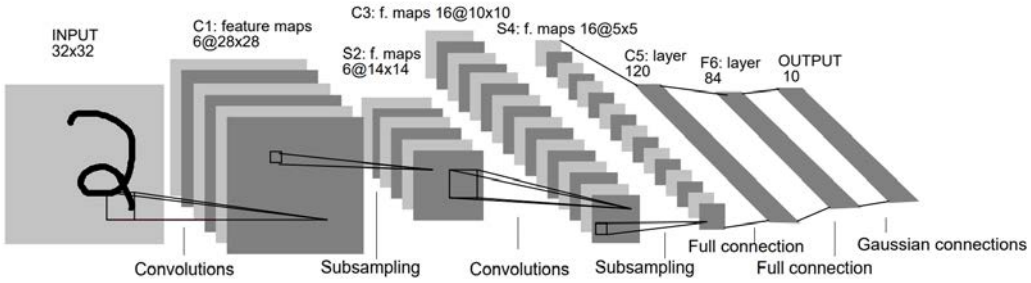

Figure 2.2: Architecture of LeNet-5, an old convolutional neural network for digits recognition. Each layer is a feature map, i.e a set of different neuron convolutions, whose weights are constrained to be identical. [2]

on manually-labelled training data, done by human observers or supervisors, hence the name for the learning method.

Supervised learning analyses and generates an inferred function that maps to the training examples, which then can be used to map to new examples outside of the training set.

A computer can be programmed to map ($f : X \mapsto Y$) to formulate several practical problems, where X is an input domain and Y is an output domain. For example, in the classification task, the object in the image is visually recognized to classify it, where X was the dataset of images and Y is a set of corresponding values of a fish existing in each image or not.

Unfortunately, most of the time, it is difficult to manually feature engineer (i.e implementing domain knowledge of a dataset to create 'features' or reduced dimensionalization of the pixel values that allows deep learning algorithms to be more efficient) in order to determine a function f that can recognize a fish in the image. Comparatively, it is often more feasible to collect a large dataset of $(x, y) \in X \times Y$ for the mapping process, and this affords supervised learning advantage as an alternative mapping technique when solving these kinds of problems. Specifically, in the fish classification task, I collected a dataset of fish images, each image being manually labelled for the presence or absence of a fish.

When conducting supervised learning on a dataset of n example images $\{(x_1, y_1), \ldots (x_n, y_n)\}$, a mapping function ($f : X \mapsto Y$) can be identified from a set of functions by searching and selecting for the function that is

most consistent relative to the other functions within the training dataset. More specifically, consider a class of functions $\mathcal{F}$ that map $X \mapsto Y$, and a scalar-valued loss function ($L(\hat{y}, y)$) that computes the difference between the predicted label $\hat{y}_i = f(x_i)$ for $f \in \mathcal{F}$ and the true label y_i to find a mapping function f^* that minimizes the loss over the training dataset. Upon finding this function, the original training dataset can then be discarded but the learned function f^* is retained, which is used to map elements of X to Y. To ensure that the function f^* *generalizes* outside of training data, the process of regularization is applied.

Regularization is a technique that makes small changes to the learning algorithm to improves the performance of the model on testing or out-of-sample data. In other words, it avoids the risk of over-fitting to the training data by discouraging learning of more complex models. Model regularization involves a regularization term being added to the general cost function, which makes more complex models more costly (increases g). This means that the relative weight matrices for CNNs with simpler models are smaller and thus favoured over CNNs using more complex models and higher relative weight matrices. The most common forms of regularization are L1 and L2.

Thus, using regularization, the supervised learning task becomes an optimization problem taking the general form of: $\theta^* = \arg\min_\theta g(\theta)$, where θ is a parameter vector and $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(f_\theta(x_i), y_i) + R(f_\theta)$. Note the added regularization function, $R(f_\theta)$ within the cost function g. The θ parameter is a part of the mapping function f, while the functions L and R do not require any training.

2.4 Neural Network Workflow

In this section, I present the standard workflow for implementing a neural network in practice.

2.4.1 Data preparation

The first step is to obtain a dataset. For this thesis, the data were represented as a collection of pairs (x, y), where x is an input (i.e. an image of a habitat

that potentially contains a fish) and y is a label (a binary mask or integer [0,1] indicating if there is a fish present or if this pixel is part of the fish). After this, the data were split into three different folds/subsets: a training fold, a validation fold and a test fold (80%, 10%, and 10% of the complete dataset, respectively). The training fold is used for optimizing parameters through backpropagation, the validation fold for optimizing hyperparameters, and the test fold for the final step of evaluating the model's accuracy.

2.4.2 Data preprocessing

Data preprocessing can help improve the convergence of neural networks [2]. As the datasets used in this thesis are images, a common preprocessing technique for normalizing and standardizing images was used. This technique subtracts the mean of the image to normalize it and divides it by the standard deviation to standardize it independently for every input dimension of x (i.e. each image). This standardization was done during the train, validation and test phases, and the application of these statistics in preparing the validation and test dataset as it helps with the deployment of the definite model into a real-world application.

2.4.3 Architecture design.

In this step, the neural network architecture (the function of f, as notated above) is determined. This step is sometimes said to be more of an art than a science [28]; however, there are some simple heuristics that can be used to set up an appropriate architecture in practice. It is common to process images (pixel data) with convolution layers, because of the parameter-sharing benefits to convoluted neurons (weights) in determining broader patterns in the images. Recall that the adjacent neurons in one activation map share the same weights (filters). This greatly reduces the number of trainable parameters in each convolutional layer relative to a layer with one neuron per each pixel or a group of non-overlapping pixels. A very rough rule of thumb for deciding the scale of the CNN architecture is that the full model should have a roughly comparable number of parameters as the number of examples (i.e. images) in

the training dataset [28].

2.4.4 Optimization.

In this thesis, I applied the optimization algorithm *Adam* [29]. *Adam* is an adaptive learning rate optimization algorithm, which means that it calculates unique learning rates for various parameters in a dynamic or changing way. To adapt the learning rate for each weight of the neural network, Adam utilises estimates of the first and the second moments of the gradient. Adam's initial learning-rate (lr) was set to 1×10^{-4}, where the rate was halved every time the *validation* accuracy did not increase after 10 epochs (the number of passes of the entire training dataset the CNN model has completed during training). The training was done in batches of 4 images at a time, and training was restarted twice from an initialization based on the highest-accuracy model if the validation accuracy did not increase after 32 epochs, where upon each restart the initial lr was multiplied by 0.9. As a sanity check for code debugging and to compare to other batches for overfitting, I intentionally overfit one individual batch of the dataset (approaching zero training loss) before optimizing the whole training dataset.

2.4.5 Hyperparameter optimization.

Stochastic gradient descent optimization can be seen as an *inner loop* of the optimization, or an iterative method for optimizing an objective function, while hyperparameter optimization is the *outer loop* that defines proper values of hyperparameters in which the chain rules applied during backpropagate are hard to exploit (e.g., it is difficult to determine the number of units in each hidden layer or the appropriate learning rate). This process is done by first searching for optimal hyperparameters, next optimizing the model, and finally evaluating the model. Eventually, the model that achieves the best validation accuracy (and least validation loss) is accepted as the final model.

2.4.6 Evaluation.

Once the final model is determined, I tested the model on the test fold of the complete dataset and documented the prediction accuracy on this data that had never been seen by the model before. Additional enhancements to model performance are also achieved by using model ensembles, which average the outputs of various models trained from different initializations or with different hyperparameters.

Using the computational techniques described above, the rest of this thesis aims to address my two main aims: (first aim) training CNN models to determine fish presence/absence (Chapter 3) and classification, counting, localization, and segmentation of fishes (Chapter 4). (second aim) generating weight estimates for these fish (Chapter 5 and 6).

Chapter 3

Underwater Fish Detection

This chapter and the following chapter are devoted to the **first part of the thesis project**: underwater Fish Detection. Specifically, a CNN-based binary image labelling efficient procedure was developed to train for fish-detection within images (i.e. fish/no-fish) and fish-localization. The goal of this chapter is to develop a deep learning method capable of significantly reducing the amount of human effort needed to analyze fish habitats. The structure of this chapter is as follows: Section 3.1 introduces the procedure. Section 3.2 examines recent developments in the detection and classification of underwater fish. Section 3.3.1 describes the labelling-efficient training and testing data preparation protocol. Section 3.3.3 presents the training pipeline. Section 3.3.4, the most novel feature of this work, demonstrates the weakly-supervised training of CNN fish detectors using external-to-project image domains. Section 3.4 presents the results on the project test images not used in the fish-detector training.

The content of this chapter is written by the candidate, and he also wrote the code, annotated the dataset, and performed the experiments.

3.1 Introduction

Remote underwater video (RUV) is a tool used in fisheries monitoring, as well as for the management of ecosystems, and conservation programs [30; 31]. RUVs aid in the collection of valuable information about fish and habitats used by fish at both low cost of deployment (relative to direct surveys of fish) and with less human interference. The application of remote underwater video (RUV) is divided into *baited*[31] (BRUV) and *unbaited* (UBRUV) cameras. UBRUV processing (UBRUV) was used for the purposes of this project, as it offers the following advantages: a) UBRUV provides more information regarding early life history of fishes, b) UBRUV provides more information about the spatial distribution and temporal dynamics of juveniles.

For fisheries and conservation management, this information is important because it provides (a) the ability to estimate the scale of potential stocks, (b) an understanding of the scope and direction of population change, (c) knowledge of juvenile fish habitats that can be prioritized for conservation purposes, and (d) understanding of early life-history of fish and differences in fish abundances across habitats with different levels of environmental change.

Over time, the quality of cameras used in BRUVs and UBRUVs has continued to improve. As with most electronic technologies, this trend is likely to continue, where the same technical specifications are likely to decrease in price, while more advanced video recording options become available. Even at the current RUV unit prices of $100-$1,000 USD, it is still financially viable for governments and monitoring organisations to deploy a large number of RUVs that will generate a large amount of video. However, this video must be processed from the deployed RUVs, which usually requires humans to watch and annotate thousands of hours of videos before a large-enough dataset is obtained for which population inferences can be made.

Additionally, BRUVs and UBRUVs are often deployed in sometimes visually-difficult environments with high turbidity or discolouration (Figs. 3.1). Here, RUV footage often becomes at best unreliable and at worst unusable for the detection of fish by human observers (see Section 3.2). Since fishes are often missing in many sections of the unbaited videos, it is here that modern Deep

Learning [32] though Convolutional Neural Networks (CNNs) can aid tremendously by automatic sorting underwater video clips into segments where fish are present vs. segments where fish are missing altogether [33].

3.2 Related Work

The Fish4Knowledge or F4K [34; 4; 35; 36] project is the first large-scale automatic fish detection and classification study carried out over five years between 2010 and 2015. The project is based on images and videos. F4K was first collected in Taiwan, with tens of thousands of hours of submarine coral reef video clips.

Based on the Fish4Knowledge project, [37] published the LCF-14 manual with annotated dataset of 30,000 fish images and 1,000 video clips of 10 different species of fish. The pictures and videos were used for the LifeCLEF2014 challenge dataset [34] competition. For fish classification, the VLfeat-BoW [38; 39] was applied as a baseline for the identification of fish in still pictures with 97% average precision (AP) and 91% average recall (AR), defined as [40]

$$AP = \frac{1}{c}\sum_{j=1}^{c} TP_j/(TP_j + FP_j), \tag{3.1}$$

$$AR = \frac{1}{c}\sum_{j=1}^{c} TP_j/(TP_j + FN_j), \tag{3.2}$$

where TP_j, FP_j and FN_j are the numbers of true-positives, false-positives and false-negatives within the classified results for the jth species, respectively, and where c is the number of species. The video algorithm ViBe [41] was used first for background subtractions, then followed by the VLfeat-BoW [38; 39] to achieve only $AP = AR = 54\%$. The same LCF-14 test videos have been reported using the [42] Support Vector Machine (SVM) Classifier, with very similar $AP \approx AR \approx 50\%$. The slightly lower recall (91%) from (3.1) was attributed to the higher number of false-negatives FNj (as against FP_j, there was notably higher overall accuracy: 97%). In addition, [37; 42] showed significantly worse outcomes ($AP \approx AR = 50 - 54\%$) for video data. More specific fish *detection* methods are the subject of this section of the project.

From the fish task competition of LifeCLEF2014 [34; 37], [43] used the Fast Regional Convolutional Network, Fast-R-CNN, together with the AlexNet

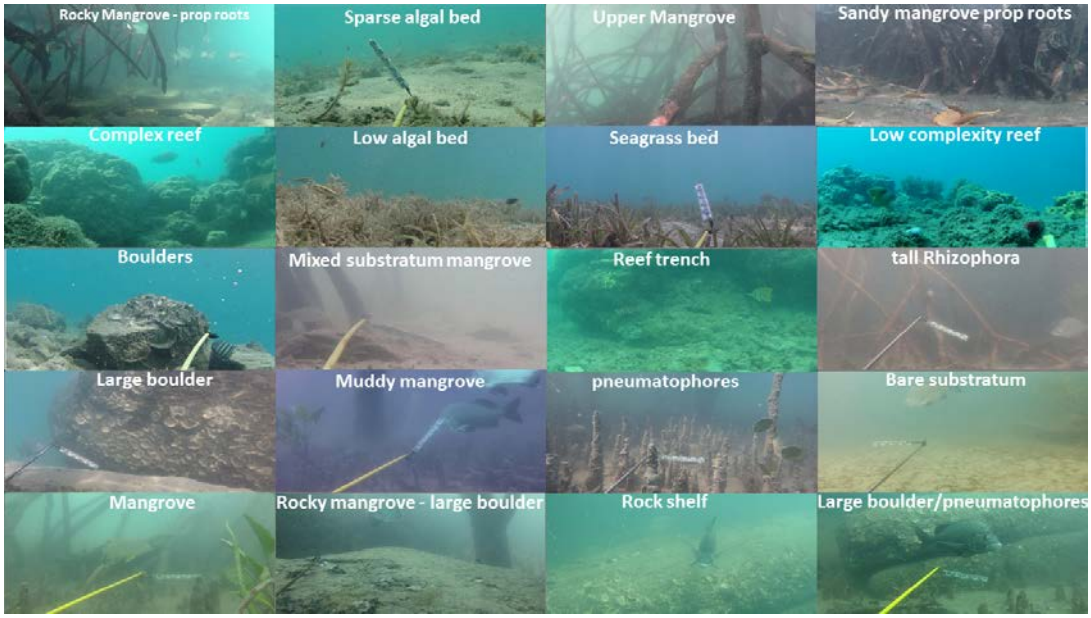

Figure 3.1: Sample fish-containing video frames from the 20 considered habitats.

for rapid object detection and classification in images, as has also been used in [33]. 24,272 image sets were manually curated for training and testing to classify 12 different fish species. The fish task pictures of LifeCLEF2014 were derived from the F4K collection [36]. For the 12 species, a mean average precision (mAP; the total area under the precision-recall curve) of 81.4% was achieved [33] (see [33] for the exact definition of the total area under the precision-recall curve). The most important aspect of [33] to manually pick one of the 12 species for each train and test file. In this case, the Convolution Neural Network (CNN) knew during training that any picture could also have unrelated fish features that were not related to the species being trained/tested upon. In submerged video system, the ability to *detect* each species is often unknown.

For the purposes of classifying test images by looking for the most related species type, the face recognition algorithm of [44] was applied in [30]. The LCF-14 [37] dataset used to build 32×32 grayscale training and testing images for [30]. The average accuracy of classification was 94.6% classification, which was an important improvement compared to a similar study, [45], using sparse image representation. The [30] approach for face-recognition, and the likely cause of the improvement in accuracy, relied on an external process [46], which

makes it less realistic to extract and crop fish sub-images from any particular video.

In some experiments, the MoG [47] (Gaussian Mixture-based Background/-Foreground Segmentation Algorithm) used to segment moving fish from the stationary subsurface are used as context substraction in [46]. Variations in pixel intensities for fish are on average less than fluctuations seen on real fish habitats due to the fish being the *foreground* target. In the case of real fish habitats, MoG [47] context subtraction works extremely well. The clean and debris-free water in the lower right corner, for example of Fig 3.1 exemplifies this. The MoG algorithm is readily available in many popular software packages such as OpenCV [48] and Matlab (TM). The typical motion detection methods of fish (for example, [46]) in complex under-water fish environments could not, by nature, differentiate between floating waste and comparably-sized juvenile fish, or when the fish were stationary. Fish are also inseparable from the ground debris when they remain stationary for long periods of time, as seen in the middle of Fig 3.1's sparse algal bed (top row, second image from right), and fish in the mid-left (similar subfigure). Moreover, background variations of pixels are comparable to pixel variations on slow moving fish when using the MOG method [47], causing it to fail to differentiate the fish.

To train CNNs and use them in different environmental sites and/or detect unknown species, only three CNN layers are used in [40]. CNN has been tested for LCF-14 using 20,000 labelled images as a training set, as well as 93 videos for 15 fish species, with training accuracies as high as $AP = 97.18\%$ [49; 50]. In [40], videos were used in both [37] and LCF-15 data sets, which processed frames made into separate frames and then all images were redesigned to the 32×32 form and converted to gray scale. However, the CNN performance degraded to $AP = 65.36\%$ when trained on LCF-14, but tested with the noisy and low-quality images from the LCF-15 dataset. In addition, [31] reported that the CNN classification precision of [40] declined from 87.46% when trained on LCF-15 data, to 53.5% when used on a completely different dataset [31]. This drop in output highlights the technical limitations of any fisheries or ecology monitoring project. It is difficult to train CNNs for generic fish detection and to use them in different environmental locations

and/or to identify previously-unknown organisms. Therefore, the financial and human costs of setting up and developing a CNN project have become a central consideration to be discussed in this section.

Images from habitats in West Australia including kelp, sand, seagrass and coral reef habitats [51] were recorded via UBRUVs in [31]. The videos have been processed and resulted in 2,209 images containing 16 labels for 16 different species of fish and a 17th label indicating *other* organisms apart from fish. The images extracted and resized to a $224 \times 224 \times 3$ shape, where the three colour channels were kept (hence the additional $\times 3$). This resulted in a large improvement in vision processing performance relative to the greyscale images [40; 30]. AlexNet[52], VGGNet[53], and ResNet[54], were used in [31]. A comprehensive image collection using ImageNet [55] weights, to initialise initial CNNs layers were loaded for the pre-trained weights during setup [56; 57], and this is commonly known as *transfer learning* or *knowledge transfer*. An ImageNet-trained CNN often performs better than an image class [57] CNN. Without further training in image extraction, the three ImageNet-trained CNNs were applied and then used as an input in the Support Vector Machine classification model [31]. Out of the three CNNs, the Support Vector Machine (SVM) classification using the ImageNet pre-trained ResNet [54] achieved the best accuracy of 89% on the 663-image trial sub-subset. However, a 96.73% higher accuracy was achieved on the testing set relative to Fast-R-CNN [33] and the face-recognition-method [30]. A ResNet+SVM combination can obtain even better results, when examining the LCF-15 dataset [49; 50]. ResNet and SVM [31] focuses on classifying photos that have been detected and correctly cropped externally (and manually). Only when combined with automated fish detection and using comparatively accurate bounding-box methods, can the high accuracy precision of ResNet and SVM be achieved. At present, only [37; 42] has been calculated at 50% accurate.

The Seafloor Observatory (OBSEA) has documented an automated method of counting fish in real-world videos to track remote coastal ecosystems [58]. In order to detect fish, 11,920 images obtained from the OBSEA test site [59] in 2012 were used for training binary (fish/non-fish) classification, then tested on 10,961 images taken at that site in 2013. Two separate steps were taken in

the OBSEA method:

- In the first step, all training and testing images were taken from the extracted Regions of Interest (RoI), i.e. the part of the image containing a fish. The RoI stage is very similar to the MoG-type [47] method, so the limitations of RoI are similar to those previously discussed for MoG, specifically: a) there are a high percentage of false-negatives for fish if the fish is stationary. b) the detection threshold is relatively low. Figures and supplementary videos from [58] showcased this second argument, in which the RoI step did not segment several fish in the images. In the RoI step, sequential images are sorted by their retrieval time and the image differences are extracted as RoIs.

- The second step is to use [58] to generate binary fish/no-fish classifications to ensure that the manually marked RoIs are correct for every segmented RoIs. RoIs accuracy was 92% by the first step. In the same way, as in the ResNet and SVM results [31], if the previous RoI/bounding box segmentation process results in a low false-negative and false-positive rate, then the accuracy of this method is on a per-fish/per-image basis; however, [58] did not report false-negative and false-positive rates to be compared between the two methods.

The previous paragraphs reviewed the available literature of recent studies and provided intuition for the following working hypotheses:

- The labelling/annotation of bounding boxes/RoI for each training picture is required in all the checked classifiers.

- RoI extraction procedures in complicated reef habitats are just 50% -80%, significantly less accurate than those of similar studies using different methods in the same habitats.

- Due to a RoI or bounding boxes in a picture, a number of methods for the correct classification of fish or fish/not-fish detection have been reported, achieving 85%-95% precision.

- The accuracy of RoI/segmentation methods depends heavily on the image's background relative to the fish.

- Trained CNN is extremely specific for fish and/or environmental habitat and are unlikely to work as well on other species or backgrounds.

Figure 3.2: Examples of frames from mangroves habitat with: one *Lutjanus argentimaculatus* adult (top row), one *Chaetodon vagabundus* (bottom row), and multiple *Caranx sexfasciatus* juveniles (middle row). Top-left, middle-right and the bottom-row images were all histogram equalized via CLAHE.

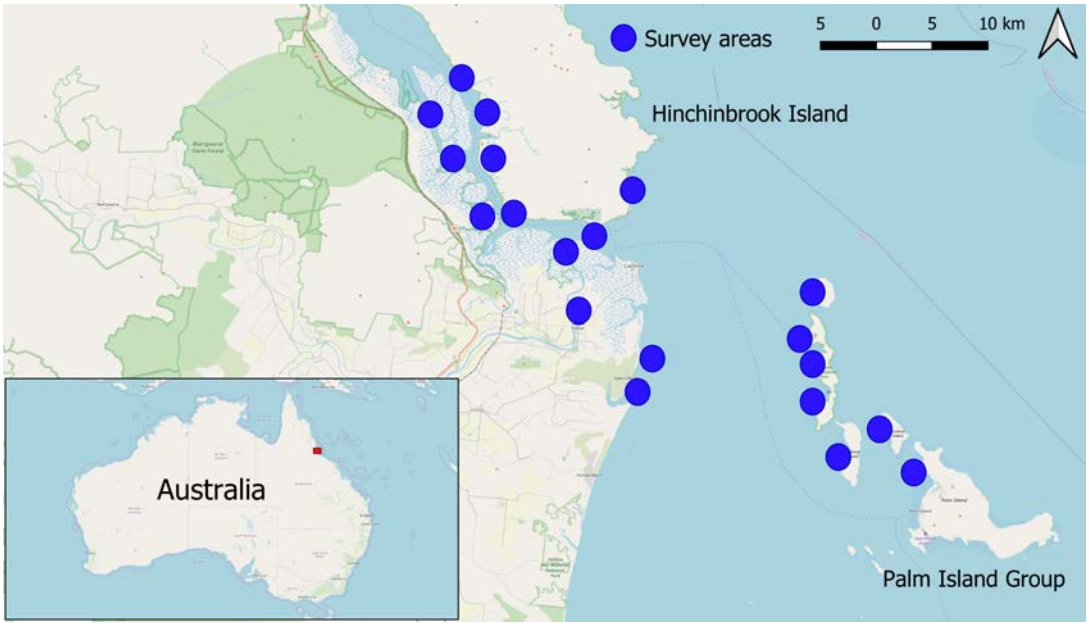

Figure 3.3: Locations of video clip capture via unbaited remote underwater video (UBRUV) along the near-shore island chain of the Great Barrier Reef (Queensland, Australia).

3.3 Materials And Methods

3.3.1 Standardized Procedure for Dataset Preparation

A useful and labelling-efficient data preparation method that can be employed in future fish surveys is the primal goal of this chapter. The following processes were performed: 1) Video clips were chosen from 20 different locations (see the typical Figs. 3.1 examples. 2) Video clips of the Great Barrier Reef near-shore island chains have been recorded during a series of different environmental habitats and conditions (see Fig. 3.3). 3) These videos depict the variety of conditions faced in a typical survey of tropical fish and belong to field data provided during an evaluation of juvenile fish habitat [60]. 4) Locations often include various 3D ecosystem architecture, natural lighting, oceanic currents, suspended sediment levels (turbidity), and organic particles ("sea snow").

The video was annotated by an experienced and qualified marine biologist

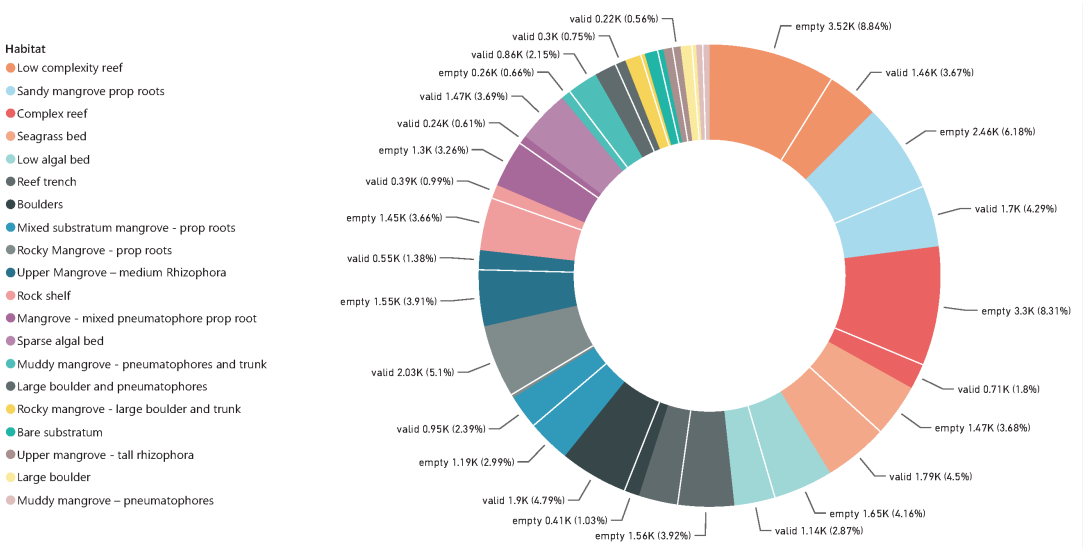

Figure 3.4: The distribution of all images in the dataset, across 20 different aquatic habitats with the relative proportion of images containing fish ('valid') and those without fish ('empty').

who was previously familiar with the habitats and videos, and they took about two days (10 hours) to complete the annotations. The biologist checked to see if there was at least one fish within the image. All clips containing fish were then placed in a *valid* sub-folder, while all clips without fish were placed in the *empty* sub-folder. At least one valid fish recording clip was recorded in all but one habitat (the Sparse Algal Bed, see Fig. 3.4. The (first, 11th, 21st etc.) frames (intervals of 10) were used for the training dataset (referred to as FD10), wherein the remaining frames were saved for the testing set (referred to as FD10-test). All clippings were then converted into individual frame images. Overall, 40,000 frames were generated, with 1,764 (fish) positive and 2,253 (no-fish) negative images contained in the FD10 data-set. There were 16,000 positive and 20,000 negative images on the FD10-test dataset. This video-level labelling is very useful for the creation of thousands of image-level annotations for projects. However, only when video clips containing fish and no-fish are recorded using the same RUVs is the suggested protocol for labelling valid. It is necessary to know the fish features of a CNN model (specific to 'valid' video clips containing fish).

The FD10 and FD10-test collections of unprocessed original frames have

been further processed with the use of the OpenCV [48] library and the CLAHE [61] algorithm, to generate the FD10c and FD10c-Test data sets of processed images. A limit value for the CLAHE standard clip was preserved at 2 and 16 column tiles and 8 row tiles were set for the grid sizes of the CLAHE tile. Each picture was first transformed by the associated OpenCV function from RGB colour space to CIELAB space [62], and then the CLAHE algorithm was used to determine the luminosity channel (L). The resulting frames processed by CLAHE were visually better than untreated, unprocessed frames, as shown in Fig. 3.2.

The initial videos in the project resolution were 1080×1920, and the total length was approximately 200 hours (600 clips, 20 mins/clip), demanding at least a 200-hour time commitment/payment by dedicated marine biologists to analyze all videos (or an compensation payment from a missed opportunity cost). There are no fish species in more than half of the video frames, which means that the original videos contain no fish recording in at least 100 hours of the footage. The inefficiency of the work for biologists is further amplified when they must also determine which segments of video contain fish vs. which can be ignored. Since these types of surveys using UBRUVs are frequently carried out, the method of fish detection can be made more efficient using an application of computer vision that is presented below.

3.3.2 The Dataset

This project is designed to learn a function which maps an input to an output using input pairs as examples. It calibrates this function from labelled training data in the process known as supervised learning. The best human-efficient labelling for supervised learning is achieved by the [63] image-level class labels. If the class label is annotated, this means that the computer will infer that one or more class instances are visible somewhere within the image scene.

Xception [21] has been chosen as the base CNN of the current ImageNet-trained model, and is available for Keras [64] CNNs. To create the expected binary fish/non-fish classifier (referred as XFishMp), I replaced the Xception 1,000 class top by one spatial/global maximum pooling layer (hence the "Mp" abbrevation for XFishMp), then added a 0.5 probability drop-out layer, and

finally a one-class dense layer using the previously-described *sigmoid* activation function. XFishMp has the smallest number of trainable parameters (20.8 million parameters), compared with 23.5 million for ResNet50 and 21.7 million for InceptionV3-based XFish-based equivalents. Additional CNN configurations denote XFishHmMp, which has shifted a final layer of XFishMp's global max pool and converts a dense one-class layer into a convolution layer. XFishHmMp adds the "Hm" to its name because of the one-class convolution layer yielding $[0, 1]$-ranged values that make up a two-dimensional *heatmap*.

The data set consisted of 4,017 colour pictures (1,764 with fish and 2,253 without fish), each with $(1,080 \times 1,920$ shape) 1,080 pixel rows and 1,920 columns. The fish measurements were mainly within the range of $[30, 300]$-pixels.

The FD10 data set is very small (4,017 images), in contrast to the over one million frames in the ImageNet dataset used to train Xception. Additional action was therefore anticipated to limit the overfitting of the XFish CNNs.

1) Training used only the greyscale versions of images for this project. This is because fish species usually have a colour change larger than the fish shape, the XFish CNNs were designed to learn (i.e. generalise) the fish forms instead of memorising (i.e. overfitting) the pixel colours, so colour features were removed. Furthermore, the colours in the underwater background vary considerably (Figs. 3.1), so that the 4,017 images studied in FD10 could be more efficiently classified when the coloured channels were added. Such a colour fixture will, however, have a minimal to no overall value outside the studied training data. Because the Xception trained models using ImageNet expected three coloured channels for input, the XFish CNNs have been added in order to accept single channel grey-to-RGB converted images.

2) In addition to the grey-scale input images, the following augmentations were taken to reduce over-fitting (i.e. further regularisation), so that training picture dimensions were limited to 512×512 to achieve practical training and to fit training to common GPUs (Nvidia GTX 1080 Ti were used). Each $1,080 \times 1,920$ original image has been converted to a grey scale and then a 5% border with a $1,188 \times 2,112$ shaped image has been zero-padded. The image was then downsized to the form of 512×512, randomly flipped horizontally,

normalized to [0,1] range, randomly rotated, and then Gaussian normalized.

3.3.3 The CNN model and training setup

The training was conducted in 4-picture batches and twice restarted from the highest model, if after 32 epochs, the exact validation did not increase and the initial lr was multiplied with 0.9 upon each restart. The image validation subset was not augmented, but was pre-processed, with 512×512 resized, $[0, 1]$ standardised, and 5% zero-padded, as per the training set. The Tensorflow [65] back-end trained all the considered models in Keras, where the Adam [66] algorithm was used as a training optimiser. For training XFishMp, Adam had a first learning rate (lr) at $lr = 1 \times 10^{-5}$ and at 1×10^{-4} for XFishHmMp, where each time the accuracy *validation* increased after ten epochs, the rate was halved.

3.3.4 General-Domain Image Datasets

Two listed fish-domains datasets were used for the weak supervision: the classification challenge dataset LCF-15 [49; 50], with 22,400 fish images, and the [67; 68] dataset. QUT2014 contains 4,000 fish images. Due to the lack of public fishery domain/video datasets for the fish species examined and due to the requirements for datasets of high quality images and quantity, I decided to use the projects-domain datasets. There are however a number of general-domain image data sets in which fish instances are labelled (e.g. ImageNet [55]), or are missing (e.g. VOC2012 [69]).

The following training pipeline is suggested to preserve the weak nature of the external mutli-domain datasets. A total of 4,000 images (known as FS10-VLQ) was expanded by 2,000 images from VOC2012, which were marked as containing no fish, as well as 1,000 images from each LCF-15 and QUT2014, which were marked as containing fish. The FS10 was thus expanded to include 4,000 images. Then, this newly augmented FD10-VLQ dataset containing 8,000 images was divided into 80%/20% training/validation folds. Since there are still many more images in all three domain-level data sets, all 4,000 additional external domain images from their corresponding datasets are

randomly redrawn at each training stage.

Xception CNN was trained on more than a million images (including some fish pictures). For modern high-performance CNNs such as Xception, the FD10 project collection of 4,000 training images still was very little. This study has consequently regularised XFish CNNs with negative general (i.e. no-fish) domain images, which have been used to achieve weakly negative supervision such that there are slightly more no-fish images than with-fish images, the 17,000 VOC 2012 [69] no-fish images used in this study. At the start of each video, when the camera was started manually before it was placed and secured to its undersea destination, all the original videos (useful as a basis of this project's training videos) contains above-water segments. The negative daily images of the VOC2012 type helped to reject these false-positives even more robustly.

3.3.5 The Heatmap

You can easily convert XFishHmMp for any localization task by eliminating your last max-pooling layer arriving at the XFishHm CNN. The grey input image is thus converted from 512×512 to 16×16 heatmap of $[0, 1]$-ranged values. The detection of fish usually involves *localising* within an image where a fish is likely to have been detected. The careful annotation of the fish-containing and missing-fish FD10 images from the same underwater sites achieved a weak fish-localization. Note that the direct fish level monitoring by means of a bounding box [70] to provide pixel level *semantic segmentation* or point-level [63] annotation, was considered outside of the scope of this project due to the increased time cost of labelling images.

3.4 Results and Discussion

3.4.1 Training Baseline

In Tensorflow [65], the XFishMp and XFishHmMp were created using the back-end in Keras [64], to establish a baseline in which the label-stratified split was 80% (using a fixed seed) for the training and validation of FD10 and FD10C datasets. Binary cross-entropy was used as the training loss function.

Applied to FD10-test dataset that had not been previously processed through CLAHE [61], all FD10 and FD10c models (Table 3.1) have been evaluated. The network processed 7-8 pictures per second (one image per batch) on GTX 1080 Ti, which was marginally acceptable. For instance, through further optimisation by running larger batches and/or only loading every second or third frame, increased processing of greater volumes of sub-sea videos during deployment is possible. Additional CLAHE pre-processing, however reduced the test rate to 0.5-1 pictures per second and was therefore not considered to be a viable deployment option.

Only the 10th frame was used for testing the accuracy of the model during training (or validation). It can be assumed that the remaining test frames were classified appropriately (zero false negative and zero false positive) from the holdout FD10-test dataset. The 0.5 threshold was used to accept the positive/fish activation output of the CNN and to qualify negative/no-fish if the output values were below this threshold. XFishMp (trained using FD10, see Table 3.1) achieved the lowest baseline false-positive ($FP/N = 0.73\%$), while XFishHmMp was based on heat-maps (trained using FD10) and had the lowest baseline False-Negative rate ($FN/P = 1.67\%$). The total number of positive (P) and negative (N) test pictures are displayed in Table 3.1, respectively.

Training using the FD10c picture *cleaner*, decreased the CNNs' capacity in general, with $FP/N = 0.73\%$ and 3.89% for the XFishMp+FD10c CNN, while the best baseline false-positive performance deteriorated. [40] reported conceptually similar results, in which training on the noisy LCF-15 dataset was more precise than training on cleaner LCF-14 and then testing on noisy LCF-15. Thus the image cleaning pre-processing is not necessary and may even be

damaging to the CNN's performance, despite being visually appealing.

3.4.2 General-Domain Image Datasets Supervision

In order to inspect the effect of additional weak domain level supervision, the XFishMp and XFishHmMp CNN's baseline-trained dataset were refined on FD10-VLQ and FD10c-VLQ (see Section 3.3.4). However, no external image was used for the CLAHE preprocessing. The training pipeline was virtually identical, whereby the corresponding initial learning rate was reduced by a factor of 10 and only one training phase was used. The training was therefore not been restarted once it was being aborted.

All baseline cases (Table 3.1) were improved to some extent by the use of general domain image datasets, with weak supervision. High false-positive rates ($FP/N = 0.50\%$) as well as false-negatives ($FN/P = 0.90\%$) were been achieved by heatmap-based XFishHmMp CNN (trained on raw FD10-VLQ). There were only two cases of no improvement: false-positives (FP) for XFishMp (trained on FD10-VLQ) and false-negatives (FN) for XFishMp (trained on FD10c-VLQ).

For both valid (positive) and empty (negative) images, there was a stronger separation of activation values achieved by the receiver operating characteristics' (ROC) area under the curve (AUC) [71] (see bottom right quarter of Table 3.1). The further weak positive supervision could not significantly improve the false negative rate (FN), where external datasets of fish were very different from the images in the project-domain dataset (in LCF-15 [49; 50] and QUT2014 [67; 68]). Furthermore, external negative, weak supervision may improve the false positive rate (FP), which decreased significantly from 752 to 492 in XFishMp+FD10C (Table 3.1).

3.4.3 Heatmap Localization

The lowest FP and FN errors were achieved by the XFishHmMp heatmap-based CNN (Table 3.1). XFishHmMp was converted to the XFishHm CNN (see Section 3.3.5) after removing the last layer of max-pooling. For further research into this localization task, annotated bounding boxes or segmentation

Table 3.1: Confusion Matrix for Model Testing Fold Results using the FD10-Test Dataset

Actual	Model	Predicted	
	Train dataset	Negative	Positive
Negative (no-fish) N=20,104	**XFishMp**	***TN***	***FP (FP/N%)***
	FD10c	19352	752 (3.89%)
	FD10c-VLQ	19612	492 (2.51%)
	FD10	19958	146 (0.73%)
	FD10-VLQ	19967	137 (0.69%)
	XFishHmMp		
	FD10c	17954	2150 (11.98%)
	FD10c-VLQ	17925	2179 (12.16%)
	FD10	18875	1229 (6.51%)
	FD10-VLQ	20,004	**100 (0.50%)**
Positive (fish) P=15,601	**XFishMp**	***FN (FN/P%)***	***TP (AUC%)***
	FD10c	1027 (7.05%)	14574 (92.95%)
	FD10c-VLQ	1154 (7.99%)	14447 (92.01%)
	FD10	195 (1.27%)	15406 (98.73%)
	FD10-VLQ	164 (1.06%)	15437 (98.94%)
	XFishHmMp		
	FD10c	2304 (17.33%)	13297 (82.67%)
	FD10c-VLQ	1758 (12.70%)	13843 (87.30%)
	FD10	257 (1.67%)	15344 (98.33%)
	FD10-VLQ	**139 (0.90%)**	15462 (99.10%)

masks for the FD10-Test images is required. However, XFishHm was used to verify the consistency of the heatmap fish locations for all FD10 images, and the results were visually inspected. In Fig. 3.5, typical heatmap segmenting examples are shown, where the original training size of the heatmaps was 512×512 (from the output XFishHm 16×16).

The XFishMp Architecture was not consistently higher than XFishHmMp and XFishMp (Table 3.1) could not be converted into heatmaps immediately. Therefore, XFishHmMp may serve as the starting architecture for future work. Note that in the XFishHmMp, the Xception CNN base could be easily and quickly substituted with any other contemporary CNN where the trainable grey-to-RGB convert layer takes care of any required image normalisation.

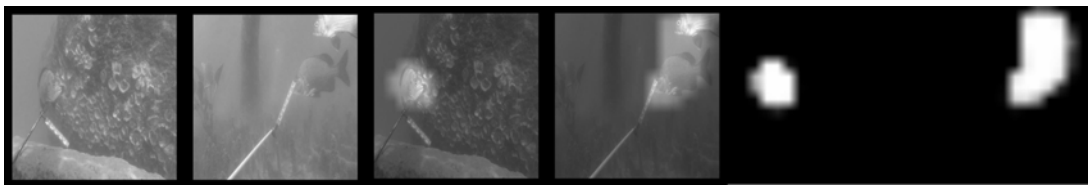

Figure 3.5: Typical examples of fish correctly detected and localized by XFishHm (trained as in XFishHmMp), where the left two subfigures are the padded and re-scaled original images in grayscale, the middle two subfigures are the same images overlapped with the prediction heatmaps, and right-most two subfigures being the generated prediction heatmaps.

3.5 Conclusion

This work developed a new method for the successful labelling of a CNN fish detector (the Xception CNN was used as a base) on a relatively small number (4,000) of underwater fish/no fish project-domain images from 20 different habitats along the Great Barrier Reef of Australia. Moreover, the general domain dataset (VOC2012) of 17,000 images with known negative (missing fish) and over-water photographs have been used. A further 27,000 above-water and underwater positive/fish pictures were supplied by two publicly available fish-domain datasets. With 0.50% false-positive and 0.90% false-negative images, a trained Xception binary classification (fish/no-fish) produced 20,104 negative and 15,501 positive images in the holdout test. The area of the ROC (AUC) curve was 99.10%.

The novel training procedure developed can be used more effectively for the training of a specific CNN fish detector, together with a significantly larger pool of multidomain images to classify project-domain images. The models were successfully tested using an efficient labelling technique involving a small number of human hours for annotation. The regularising impact of weak supervision on external large multi-domain image collections has been reviewed. The model generality and performance on the test set can be hindered somewhat by image cleaning pre-processing.

The next chapter is the second phase in the **first part of the thesis project**: Underwater Fish Detection (see Figure 4.8).

Chapter 4

Benchmark for Analyzing Fish Habitats

This chapter is the second phase in the **first part of the thesis project**: Underwater Fish Detection, specifically I present a benchmark called *DeepFish* that is based on a large image dataset of remote underwater video collected from remote coastal marine-environments of tropical Australia. The purpose of this benchmark is to motivate specialized algorithms that can automate the task of fish image analysis. The *DeepFish* dataset consists of approximately 40 thousand labelled images representing 20 fish habitats across Australia. As baselines, I also evaluate a variety of deep learning methods across four tasks: (1) classification, (2) counting, (3) localization, and (4) segmentation of fishes.

The structure of this chapter is as follows. Section 4.1 introduces the project. Section 4.2 discuss the methods used for this project. Section 4.3 evaluate the proposed methods on the DeepFish dataset across four tasks: classification, counting, localization, and segmentation. Section 4.4 conclude the chapter.

The content of this chapter is written by the candidate in this thesis, and also he wrote the code, annotated the dataset, performed the experiments.

4.1 Introduction

Assessing fish habitats is an important step for maintaining sustainable fisheries. These assessments provide information about which areas require protection or restoration to maintain healthy fish populations for both human consumption and environmental protection. However, such assessments require significant human effort. Reducing the labor cost involved in these assessments could have an enormous economic impact and improve our ability to maintain ecosystem health. In Australia, seafood exports alone provide 1.2$ billions a year in revenue [72; 73]. Therefore, it is important to develop tools that help in analyzing fish habitats requirements.

Many existing techniques used to understand fish habitats relationships suffer from the problem of fish flight response, especially for habitats with limited visibility [74]. For instance, a common surveying technique requires divers to conduct visual census [75]. Unfortunately, this causes disturbance to the fish leading to an inaccurate estimate of the fish dynamics. Further, divers cannot access areas with predators such as crocodiles. Other techniques include netting [76] and trawling [77] for catching then counting fishes. However, these methods are invasive and interfere with the behaviour of the fish which can lead to inaccurate estimates. Further, they are limited to estimating fish count only without analyzing their dynamics. To accurately assess fish-habitats in inaccessible and challenging environments, low-disturbance techniques are required to collect video samples [60]. Thus we focus on methods that rely on images collected by a fixed camera that cause almost no disturbance to the fish allowing for accurate long term monitoring.

Automatically analyzing fish footage using vision algorithms can significantly reduce the human effort required to understand fish habitats. However, automating this task is challenging due to the highly occlusive nature of the environments which includes illumination changes, overlapping and size differences of the fish. Existing methods that can automate computer vision tasks are based on deep learning [78; 79]. Such methods have been successfully applied to many fish datasets. For instance, they were used for classifying fish species on conveyor belts [80], for detecting and classifying fish in underwater

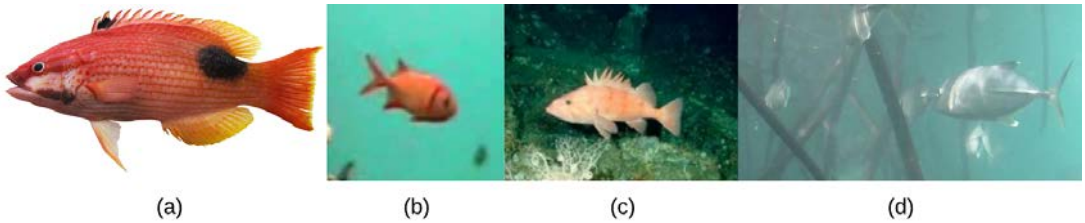

Figure 4.1: A comparison of fish datasets. (a) image from QUT [3], (b) image from Fish4Knowledge [4] and (c) image from Rockfish [5], (d) image from the proposed dataset DeepFish. Other datasets are acquired from constrained environments, whereas DeepFish has more realistic and challenging environments.

videos [81; 82; 83; 84] and for segmenting fish images [85; 19]. Therefore, deep learning is a suitable approach for automating the task of analyzing fish footage.

These methods need to be trained on large-scale realistic fish datasets. Unfortunately, existing datasets consist of images where fish are present in constrained environments [34; 36; 4; 3; 5]. For instance, some fish images from the QUT fish dataset [3] are taken in "controlled" environments where the background is white and the illumination is controlled (see Figure 4.1 (a)). Also, images collected for the Fish4Knowledge [4] and Rockfish [5] datasets are underwater but they are cropped to have single fish shown at the center (see Figure 4.1 (b,c)). However, Rockfish images have more occlusive background than Fish4Knowledge. Deep learning methods trained on such datasets will likely not perform well on unconstrained environments where overlapping, lack of visibility, and occlusions are present such as in Figure 4.3 and 4.1 (d). Thus, it is important to have a dataset that contains training images representing realistic challenging scenes as well as a diverse set of environments.

In contrast, DeepFish images represent realistic snapshots for a wide variety of fish habitats which often contain multiple fishes in challenging environments (Figure 4.1 (d) shows an example of a DeepFish image). The images come from 20 habitats from north-eastern Australia (Figure 4.3) that represent almost the entire breadth of coastal and nearshore benthic habitats commonly available to fish species in that area [7]. These diverse set of habitats cover areas of the mainland, Hinchinbrook Island and the Palm Islands (Figure 4.2).

It was shown that deep learning methods trained on a diverse set of images in the wild can perform well in a wide variety of novel environments [86; 87; 88].

Given a dataset of fish images, deep learning methods can be used to perform the following four tasks: classification, counting, localization and segmentation. Each of these tasks can help in analyzing fish habitats from different perspectives. Classifying images between those that contain and do not contain fish allows experts focus their efforts by analyzing only those images with fish. Further, having the fish count can help in monitoring fish population to avoid the risk of overfishing. Localizing the fish can be used for tracking fishes in order to analyze their dynamics. Moreover, by segmenting the fish, details about their sizes, shapes, and weights can be estimated [19; 20]. These are important statistics in applications such as commercial trawling [89]. Thus a deep learning method that can perform all four tasks has substantial advantages over many existing deep learning methods that only contain classification labels and so are limited to a single task of fish classification [80; 34; 36; 4]. Such a method provides the novel potential to allow comprehensive analysis of fish habitat utilisation. Further, I present deep learning methods trained on these labels in order to perform all these four tasks.

In this work, I present a benchmark called DeepFish based on a large-scale dataset of fish images consisting of classification, counting, localization, and segmentation labels. In addition I present a variety of deep learning methods as baselines for these 4 tasks. The dataset and the code will be made public, hopefully inspiring further research into more powerful and specialized deep learning models for analysing fish habitats.

4.2 Data and Methods

4.2.1 Dataset description

In the following sections I first summarize the data collection process, and explain the additional annotations acquired for the DeepFish dataset. Then, I describe the evaluation metrics used to evaluate the models trained on this

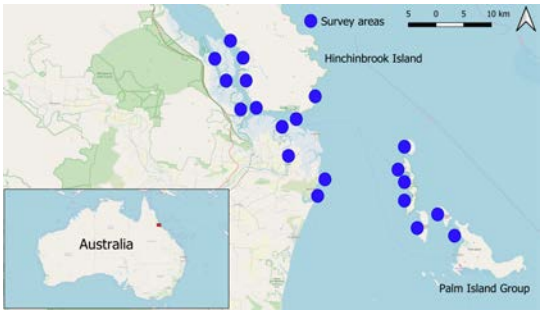

Figure 4.2: Locations where the DeepFish images were acquired. DeepFish has been acquired from the Hinchinbrook/Palm Islands region in North Eastern Australia.

dataset and how the dataset is split between training, validation, and testing.

4.2.1.1 Data collection

The images of DeepFish were collected for 20 habitats from remote coastal marine environments of tropical Australia (Figure 4.2). A brief description on the data collection process follows, but a detailed description is given in previously published works [60; 90].

Videos were acquired using cameras mounted on metal frames, deployed over the side of a vessel to acquire video footage underwater. The cameras were lowered to the seabed and left to record the natural fish community, while the vessel maintained a distance of 100m. The depth and the map coordinates of the cameras were collected using an acoustic depth sounder and a GPS, respectively. Video recording were carried out throughout daylight hours, and in relatively low turbidity periods. The video clips were captured in full HD resolution (*1920 x 1080 pixels*) from a digital camera. In total, the number of video frames taken is 39,766. Examples of these frames are shown in Figure 4.3.

Classification labels were acquired for each of these video frames, which indicate whether an image has fish or not. These labels are useful for comparing fish utilization between different habitats [84].

Figure 4.3: DeepFish image samples across 20 different habitats.

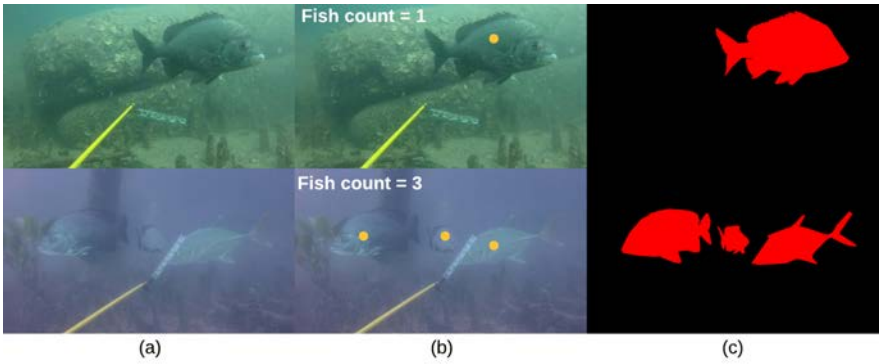

Figure 4.4: Additional annotations. (a) original images, (b) images with count and point-level annotations (c) the segmentation masks.

4.2.1.2 Additional Annotations

The original labels of the dataset are only suitable for the task of classification. Thus, I acquired extra labels in order to enable the tasks of counting, localization and segmentation, which I describe in more detail in the following sections.

4.2.1.2.1 Counting and localization annotations. I annotated 3200 images with point-level annotations. These annotations represent the x and y coordinates of each fish within the images and they are placed around the centroid of the corresponding fish (Figure 4.4(b)). These annotations were acquired using Labelme [91], which is an open-source software graphical image anno-

tation tool. These images have been also labeled with point-annotations for counting task.

4.2.1.2.2 Segmentation annotations. Since collecting segmentation labels is very time-consuming, only 620 images were labelled. To get these annotations, I labeled each pixel in the image to distinguish between pixels that belong to fish and those to the background (Fig.4.4(c)). This represents the size and shape of the fishes in the image. I used Lear [92] to extract these segmentation masks, which is a popular open-source image annotation tool commonly used for obtaining segmentation labels.

4.2.1.3 Evaluation Metrics

In order to evaluate how well models perform on this dataset, I use standard evaluation metrics for each task. For the **classification** task, I measure the accuracy of the model in predicting which images have fish in them. This is computed as,

$$ACC = (TP + TN)/N,$$

where TP and TN are the true positives (which represent the number of correctly predicted images) and true negatives, respectively, N is the total number of images.

For the **counting** task, I measure the model's ability in predicting the fish count by using the mean absolute error. It is defined as,

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{C}_i - C_i|,$$

where C_i is the true fish count for image i and $\hat{C}_i$ is the model's predicted fish count for image i.

For **localization**, I evaluate models on how well they locate fish in the images using grid average mean absolute error (GAME). It is computed as

$$GAME = \sum_{i=1}^{4} GAME(L), \qquad GAME(L) = \frac{1}{N}\sum_{i=1}^{N}(\sum_{l=1}^{4^L}|D_i^l - \hat{D}_i^l|),$$

where D_i^l is the number of point-level annotations in region l, and $\hat{D}_i^l$ is the model's predicted count for region l. $GAME(L)$ first divides the image into a grid of 4^L non-overlapping regions, and then computes the sum of the MAE scores across these regions. Note that $GAME(0)$ is equivalent to MAE.

To evaluate our models for **segmentation** I use mean intersection over union (mIoU), which is a popular metric for this task. This is computed as,

$$mIoU(P, T) = \frac{1}{N} \sum_{i=1}^{N} \frac{P_i \cap T_i}{P_i \cup T_i},$$

which is averaged over the two classes background and foreground, where P is the predicted mask and T is the ground truth mask.

4.2.1.4 Training, validation, and testing splits

DeepFish is divided into three sub-datasets: (1) FishClf for the classification task, (2) FishLoc for both the counting task and localization task, and (3) FishSeg for the segmentation task. I partitioned each sub-dataset into 3 sets. 50% of the dataset was reserved for training, 20% for validation and 30% for testing. Table 4.1 shows the statistics of these sub-datasets across the 20 habitats.

4.2.2 Deep Learning Methods

The second goal of this study is to present baselines that address four computer vision tasks for the DeepFish dataset. These tasks are classification, counting, localization, and segmentation. For each of these tasks, I showcase the efficacy of current state-of-the-art deep learning methods, which I explain in more detail in the following sections.

4.2.2.1 Classification

For classification, I used ResNet-50 [78], which is one of the most popular deep learning architectures for image classification. I used two versions of ResNet-50, CLF-1 which has weights initialized using Xavier's method [93], and CLF-2 which has weights initialized by training on ImageNet [55]. ImageNet consists

Table 4.1: DeepFish Dataset Statistics. Number of images annotated for each sub-dataset: FishClf for classification, FishLoc for counting/localization, and FishSeg for semantic segmentation

Habitats	FishClf	FishLoc	FishSeg
Low complexity reef	4977	357	77
Sandy mangrove prop roots	4162	322	42
Complex reef	4018	190	16
Seagrass bed	3255	328	16
Low algal bed	2795	282	17
Reef trench	2653	187	48
Boulders	2314	227	16
Mixed substratum mangrove	2139	177	28
Rocky Mangrove - prop roots	2119	211	27
Upper Mangrove	2101	129	21
Rock shelf	1848	186	19
Mangrove	1542	157	33
Sparse algal bed	1467	0	0
Muddy mangrove	1117	113	79
Large boulder and pneumatophores	900	91	37
Rocky mangrove - large boulder	560	57	28
Bare substratum	526	55	32
Upper mangrove	475	49	28
Large boulder	434	45	27
Muddy mangrove	364	37	29
Total	**39766**	**3200**	**620**

of over 14 million images categorized across 1000 classes. By training on such dataset the model can extract powerful features for unseen images that come from new datasets. I also replace ResNet-50's 1000-class output layer with a 2-class output layer for both CLF-1 and CLF-2. This is because DeepFish requires classifying images into either "fish" or "no fish", which is a binary classification problem (see first row of Figure 4.5).

I train CLF-1 and CLF-2 by minimizing the binary cross-entropy objective function [94]. I use Adam [95] as the optimizer with a learning rate of 1e-3 and a batch size of 16. Each image in the batch is resized to 224 x 224 which is

the expected resolution for ResNet-50. At test time, the model outputs a score for each of the two classes for a given unseen image. The predicted class for that image is the class with the higher score.

4.2.2.2 Counting

Similarly for the counting task, I used the same two versions of ResNet-50 but modified them for regression. I refer to them as REG-1 for the ResNet-50 randomly initialized with Xavier and REG-2 for the ResNet-50 pretrained on ImageNet. However, the output layer consists of one output node instead of two as the model directly outputs the predicted count (see second row of Figure 4.5). Further, I train the models by minimizing the squared error loss [94], which is a common objective function for the counting task. At test time, the predicted value for an image is the predicted object count. The rest of the hyper-parameters are kept the same as with CLF-1 and CLF-2.

4.2.2.3 Localization

For the localization task I chose a deep learning architecture that can output a heatmap representing where the objects are in the image. I use a state-of-the-art localization-based method called LCFCN [96]. Its architecture is based on FCN8 [97] which is a fully convolutional neural network that outputs a probability for an object being present at every spatial location in the image. Given a predicted probability map, the values are thresholded to become 1 if they are larger than 0.5 and 0 otherwise. This results in a binary mask, where each blob is a single connected component and they can be collectively obtained using the standard connected components algorithm. The number of connected components is the object count and each blob represents the location of an object instance.

LCFCN is trained using 4 objective functions: image-level loss, point-level loss, split-level loss, and false positive loss. The image-level loss encourages the model to predict all pixels as background for background images. The point-level loss encourages the model to predict the centroids of the fish. Unfortunately, these two loss terms alone do not prevent the model from predicting every pixel as fish for foreground images. Thus, LCFCN also mini-

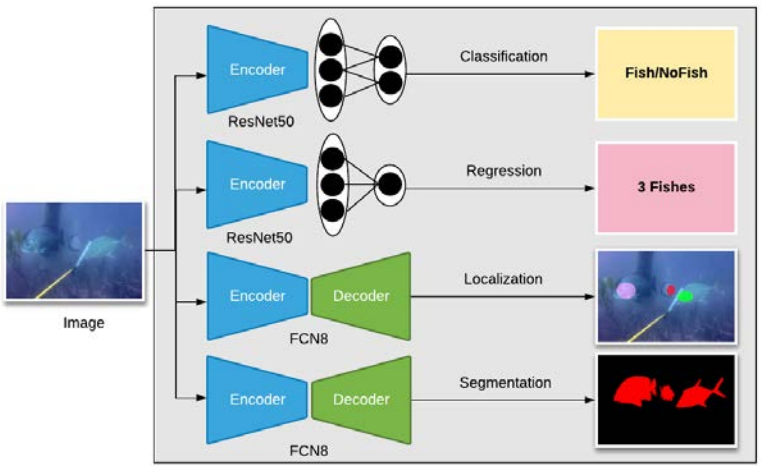

Figure 4.5: Deep learning methods. Four different deep learning models used for 4 different tasks. These tasks are from top classification, counting, localization, and segmentation.

mizes the split loss and false-positive loss. The split loss splits the predicted regions so that no region has more than one point annotation. This results in one blob per point annotation. The false-positive loss prevents the model from predicting blobs for regions where there are no point annotations. Note that training LCFCN only requires point-level annotations which are spatial locations of where the objects are in the image.

As shown in Figure 4.5 (third row), FCN8 is divided into a backbone and an upsampling path. The backbone extracts image features and can be chosen to be any of the major feature extraction architectures such as ResNet-50. The upsampling path uses the extracted features to obtain a per-pixel probability map of where the objects are in the image. I compared FCN8 with ResNet-50 initialized randomly with Xavier as with CLF-1, and FCN8 with ResNet-50 initialized after training on ImageNet as with CLF-2. I refer to them as LOC-1 and LOC-2 respectively.

These models are optimized using Adam with a learning rate of 1e-3 and weight decay of 0.0005, and have been run for 1000 epochs on the training set. In all cases the batch size is 1, which makes it applicable for machines with limited memory.

4.2.2.4 Segmentation

For the segmentation task, I have two methods: SEG-1 and SEG-2 which have the same initialization and architectures as LOC-1 and LOC-2, respectively. The difference between them lies in the objective function used to train the model. SEG-1 and SEG-2 are trained using the focal loss [98] which requires annotations of the full segmentation masks (Figure 4.4(c)). While most segmentation methods use per-pixel cross-entropy loss to train the network, it is not suitable for images where most of the pixels are background pixels. In other words, the imbalance between background pixels and pixels corresponding to fish makes cross-entropy not suitable for training. On the other hand, the focal loss can be effectively used to address such imbalance. SEG-1 and SEG-2 use the same optimization hyper-parameters as LOC-1 and LOC-2.

4.3 Results and Discussion

In this section, I evaluate the proposed methods on the DeepFish dataset across four tasks: classification, counting, localization, and segmentation. For each of these tasks I present a strong baseline based on a state-of-the-art deep learning method. These methods were evaluated on the test set by selecting the model that performed best on the validation set. The hope is that the DeepFish dataset will serve as a realistic benchmark to encourage more specialized methods for analyzing fish habitats.

4.3.1 Classification

For the classification task, the goal is to identify whether images contain fish or not.

In Table 4.3, I evaluate 4 different methods for the classification task. 'always-0' and 'always-1' are two baseline methods. 'always-0' labels every image as a background, whereas 'always-1' labels every image as a foreground. They fare poorly compared to CLF-1 and CLF-2, which are deep-learning methods. In fact, CLF-2 achieved near-perfect classification results. This suggests that deep learning has strong potential in helping practitioners analyze

Table 4.2: Counting and Localization results on FishLoc.

	always-mean	REG-1	REG-2	LOC-1	LOC-2
MAE	1.37	1.30	0.38	1.22.	0.21
GAME	-	-	-	1.30.	1.22

habitats without having to annotate all acquired images. On another note, CLF-2 performs better than CLF-1 because CLF-2 has weights initialized by training on ImageNet. Therefore, the model can extract powerful features for unseen images that come from new datasets.

Table 4.3: Classification results on FishClf.

	always-0	always-1	CLF-1	CLF-2
ACC	0.44	0.56	0.65	0.99

Table 4.4: Segmentation results on FishSeg.

	SEG-1	SEG-2
mIoU	0.49	0.93

4.3.2 Counting

The goal of the counting task is to predict the number of fish in an image. Acquiring labels for such a task is more challenging than for the classification task. This is because fish can be located in extremely occlusive areas (see Figure 4.3). Thus, only 3200 were acquired, with 50% used for training and 20% for validation sets, and the rest for the test set.

Regression-based (REG) and localization-based counting (LOC) are considered for this task. Table 4.2 shows the results of 5 methods on the counting task. The first method is 'always-mean', which computes the mean of the fish count on the training set and uses that value as the predicted count for the test images. It performs poorly compared to the other 4 methods, which are based on deep learning. This justifies the strength of using deep learning for this task. Out of these 4 methods, only REG-1 and REG-2 require count-level supervision. In contrast, LOC-1, and LOC-2 require point-level annotations, which are more costly to acquire. REG-2 and LOC-2 performed better than

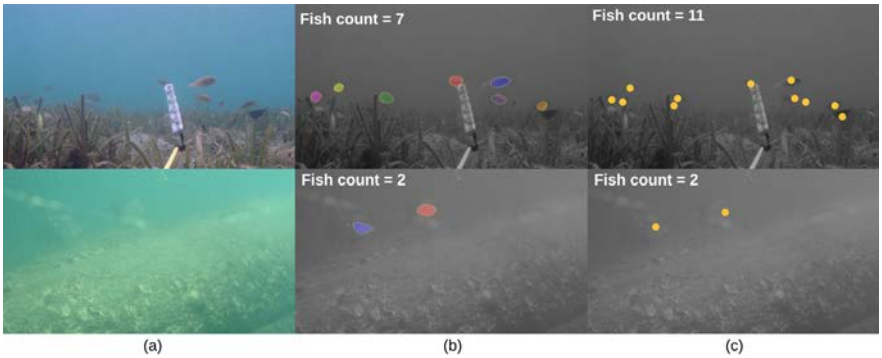

Figure 4.6: Qualitative results on counting and localization with Reg-2 and Loc-2. (a) Test images obtained from DeepFish, (b) Prediction results using localization-based loss function. (c) Annotations represent the x and y coordinates of each fish within the images and they are placed around the centroid of the corresponding fish.

REG-1 and LOC-1 because they have weights initialized by training on ImageNet. Therefore, the model can extract powerful features for previously unseen images that come from new datasets.

4.3.3 Localization

The localization task is about identifying the locations of the fish in the image. Thus it is a more difficult task than classification and counting. Fortunately, acquiring labels for localization is roughly the same cost as for counting. This is because counting fish in an image often requires pointing at each fish, giving us both localization and counting labels. As a result, the 3200 images collected for localization are the same used for counting.

In this task, I used two methods LOC-1, LOC-2. Both methods use a localization-based loss function that does not require defining the size and shape of the objects. In fact, the model learns to estimate those two properties during training (see Figure 4.6 for qualitative results). Another advantage of using a localization-based method is that it provides a probability output that can be thresholded to obtain the blobs. The threshold is based on the required tradeoff between precision and recall. In our case, I thresholded the probabilities by 0.5. Table 4.2 shows the results of LOC-1, and LOC-2. LOC-2 made a significant improvement over LOC-1. The reason behind this is that LOC-2 was pretrained on ImageNet.

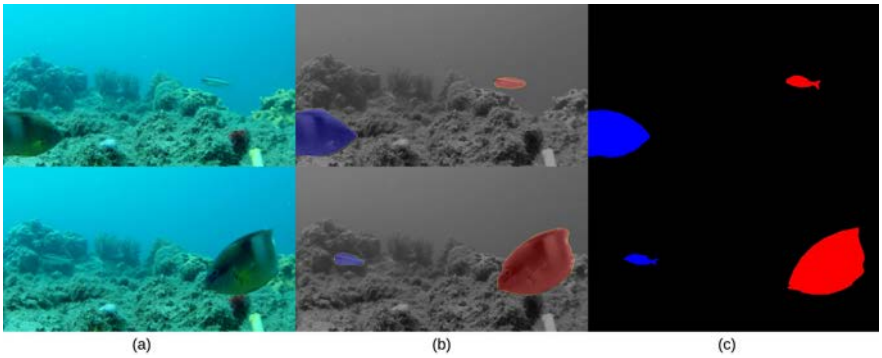

Figure 4.7: Qualitative results for segmentation with Seg-2. (a) Test images obtained from DeepFish, (b) Prediction results using the focal loss. (c) Annotations represent the full segmentation masks of the corresponding fish.

4.3.4 Segmentation

The task of segmentation is to label every pixel in the image as either fish or not fish (Figure 4.7). Combined with depth information, a segmented image allows us to measure the size and the weight of the fishes in a habitat. This vastly contributes to the accurate analysis of fish habitats. However, acquiring segmentation labels is very costly. As a result, only 620 segmented images were collected.

Similar to the localization task, I have two methods SEG-1 and SEG-2 which have the same initialization and architectures as LOC-1 and LOC-2 respectively. Table 4.4 shows the segmentation results for SEG-1, SEG-2. For the same reason as in the localization task (ResNet-50 backbone being initialized by training on ImageNet), SEG-2 made a significant improvement over SEG-1. This suggests that deep learning has promising applications in segmenting fish.

4.4 Conclusion

In this work, I introduce DeepFish which is a large public image dataset of remote underwater fish images collected entirely from coastal marine-environments of tropical Australia. DeepFish dataset consists of approximately 40 thousand labelled images across 20 fish habitats. I present strong deep learning methods that achieve good performance on the dataset across

four computer vision tasks: classification, counting, localization, and segmentation. For each of these tasks, I compared between a randomly initialized ResNet-50 against a pre-trained ResNet-50. I anticipate that the DeepFish dataset and our baseline results will inspire further research into further developments of specialized algorithms for these visual tasks for analyzing fish habitats. Future work in this area includes: developing specialized models for the DeepFish dataset, and field implementations of these models for generating critical data for the sustainable management of fisheries. The diversity of the dataset including the real-life complexity of the 20 fish habitats can help models achieve strong in-field performance when deployed in the wild.

The next two chapters will cover the **second part if the thesis project** - Estimating Fish Weight from Images. This part carried out in two phases (see Figure 4.8), the first phase is (Mass Estimation of Fish From Images). This phase developed a Segmentation Convolutional Neural Network trained on 200 images and was used to automatically segment fish-body from a background in all of this study's 1072 digital images of Asian seabass (barramundi, *Lates calcarifer*). The automatically extracted fish-body areas and the corresponding manually measured weights were fitted to yield highly accurate single- and two-factor mass-from- area estimation models. The second phase is to continue developing methods for the automatic estimation of harvested fish weight from images. I will discuss the details of the second phase in turn.

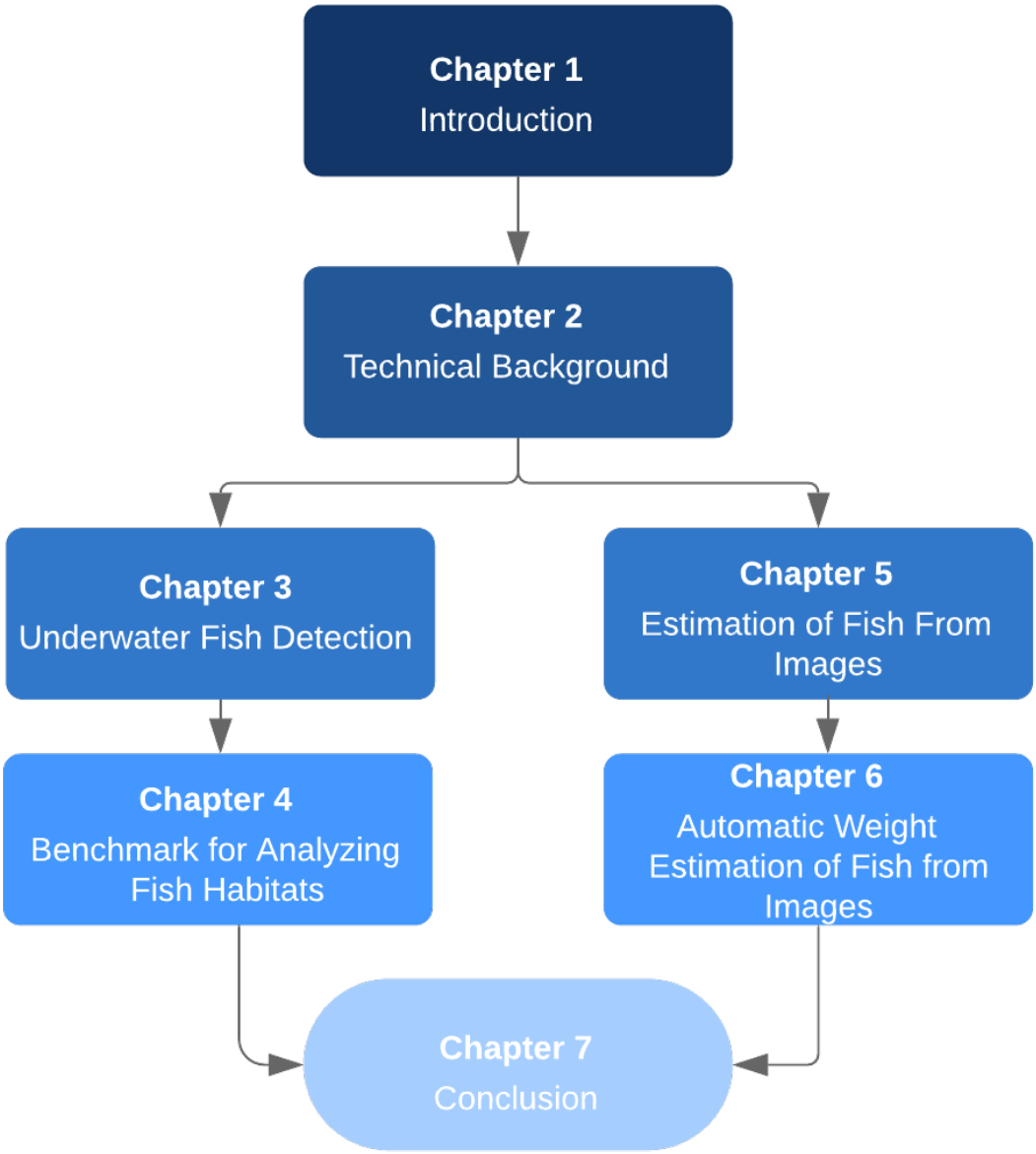

Figure 4.8: Thesis structure and interconnection of chapters.

Chapter 5

Mass Estimation of Fish From Images

This chapter and the following chapter is devoted to the **second part of the thesis project**: Estimating fish weight from images. In this chapter, a segmentation Convolutional Neural Network (CNN) was trained on 200 hand-segmented images from a total of 1,072 images of Asian seabass or barramundi (*Lates calcarifer*). Each fish in the dataset was digitally photographed and weighed. A subsample of 200 images (100 from two different locations in Queensland, Australia) were manually segmented to extract the fish-body area (S in cm^2), excluding all fins. After scaling the segmented images to 1 mm per pixel, the fish mass values (M, in grams) were fitted by a single-factor model achieving the coefficient of determination R^2 = 0.9819 and the Mean Absolute Relative Error $MARE$ = 5.1%

The content of this chapter is written by the candidate. He also wrote the code, annotated the dataset, and performed the experiments.

5.1 Introduction and Related work

Fish producers need to know the average weight of each fish over time so that they can monitor the average fish weight and accordingly change the diet routine of fish to the right intake. Individual fish weights can assist in monitoring the consistency and quality of the fish harvest. A good fish production system with careful monitoring of fish weight will increase fish farm productivity and profit for fish producers. An application that can estimate the fish weight directly from images is highly valuable for fish producers, because it will save the time and the cost of the manual weighing of the fish. Weighing each fish manually is often economically not viable and the average fish weight can thus not be obtained in order to monitor fish production. Additionally, the economic value of an aquaculture species is mainly determined by weight (M). While it is costly to manually weight fish, measuring the fish length from digital images is much more feasible by measuring from the tip of the fish's nose to the middle of the tail. Mathematical models for determining fish mass from length (L) have been established. The length-mass power model, for example,

$$M = aL^b \tag{5.1}$$

is usually used when the species-dependent parameters **a** and b are empirically fit using least squares optimization [99; 100].

Digital cameras are one application that can be used to estimate fish weight, because it is possible to automatically collect and use not only the length but also other fish shape characteristics for mass estimation. The area of the fish

Figure 5.1: (left) Drawing of how fish total length (L) is measured; (right) drawing of fish surface area (S) not including fins or the tailfin.

(S) in the image (figure 5.1) can be used to predict fish mass, M, by using a linear model. For example, for grey mullet (*Mugil cephalus*), St. Peter's fish (*Sarotherodon galilaeus*) and common carp (*Cyprinus carpio*) [101].

$$M = a + bS \tag{5.2}$$

It was verified that the same linear area-mass model (Equation 5.2) was more accurate than the length-mass power model (Equation 5.1) used on jade perch (*Scortum barcoo*) [102]. Each model was used to obtain a coefficient of determination, $R^2 = 0.99$, and the mean absolute relative error, $MARE = 6\%$. Since the linear model (Equation 5.2) appeared to do better than Equation 5.1 [101; 102], the spectrum of larger fish is limited to every non-zero fitted parameter a . The area-mass power model, on the other hand,

$$M = aS^b \tag{5.3}$$

does not show the same limitations as Equation 5.2, and has achieved $R^2 = 0.99$ for Alaskan Pollock (*Theragra chalcogramma*) [103]. The model fitted $b \approx 1.5$ [103], and this was also consistent with proportional relations between the fish length $(L \propto \sqrt{S})$, height $(H \propto \sqrt{S})$, width $(W \propto \sqrt{S})$ and between the fish volume $(V \propto LWH)$ and fish mass (M), thus the equation:

$$M = aS^{1.5} \tag{5.4}$$

from $M \propto LWH \propto S^{1.5}$. For Atlantic salmon (*Salmo salar*), a similar area-mass power model was fitted as $S \propto M^{0.61}$ (or $M \propto aS^{1.64}$) with $R^2 = 0.97$ by [104], and $S \propto M^{0.629}$ (or $M \propto S^{1.59}$) with $R^2 = 0.998$ [105].

Two objectives for this project have been set from the above arguments: 1) Establish an industrial area-mass power model to harvest Asian seabass (*Lates calcarifer*) or barramundi in Queensland, Australia. I address this aim by fitting Equations 5.3 and 5.4 below. 2) Develop a convenient image processing method for extracting the fish body area, with the exception of fins, for enhanced precision in the estimation of fish mass, and also discuss potential applications in modern selective breeding programs [106; 107]. I address this

using a neural segmentation network in Section5.2.2.

5.2 Materials and Methods

5.2.1 Harvested Barramundi Datasets

In this project, two datasets have been used: 1) Barra-Ruler-445 (BR445) is available to the public through [108], derived from an analysis of [107]. 2) Barra-Area-600 (BA600) is available to the public through [109], upon the publication of [110]. Both datasets digitally captured and recorded each harvested fish by recording the weight for each image file. All pictures had a millimetre ruler next to the fish, see Figure 5.2 for examples. BR445 weighed fish between 0.2 kg to 1 kg and BA600 weighted fish between 1 kg to 2.5 kg. The picture scales (in millimetres/pixel) have been measured manually, by measuring the pixel count between the end-points of each 300 mm ruler in each picture. An automatic ruler-scaling (RS2) algorithm [110] confirmed that the BR445 image scales were true. The BA600 images were taken from the same distance, and thus they had the same scale.

5.2.2 Segmentation of Fish Surface Area

Fins often add to the area of the fish (see illustrative examples in the figure 5.2) despite fins having negligible fish mass. Thus, in theory, fish-body area alone without fins should be used for determining fish mass. For example, when the mass of Jade perch *Scortum barcoo* [102] was predicted, the use of the fish area without taking into account the area of the tailfin was considered more precise. In addition, the fins are highly flexible and more prone to changing their shape or be lost due to damage or erosion during industrial production.

In the present study, 200 photos (100 from both datasets) have been used, which were segmented manually into fish body (excluding fins and tailfins) from the background using open-source GIMP software (see Figure 5.3). The findings were calibrated independently for the same size of the fish-body binary masks of 1 mm per pixel. All programs created to calculate fish-body pixel area were written in the Python programming language. Equation 5.4 was applied to the fish areas collected to obtain the predicted weights of fish, with the results of the two equations shown in figure 5.4. The fit was very

Figure 5.2: Examples of images from the BR445 (left images) and BA600 (right) datasets.

strong $R^2 = 0.9804$ and $MARE = 5.1\%$, which was similar to the results for other fish species [102; 103; 104; 105]. Figure 5.4 clearly demonstrates how the weight of harvested Asian seabass *Lates calcarifer* can be predicted with high accuracy using fish body surface area. However, a robust automated body-area extraction algorithm would still be required for large-scale use, which is the focus of the remainder of this section, before such an assessment method can be applied in aquaculture.

The second objective of this project is to design a practical computer vision algorithm that can extract the fish body from pictures. As stated in section 5.1, CNN semantic segmentation [111] is very successful at addressing challenges such as classifying image pixels into pixel classes [110; 111; 112]. Deep learning neural networks [32] have revolutionised modern machine learning upon the introduction of many segmentation CNN models. However, comparing among many different and more common CNN segmentation models

Figure 5.3: Examples of the 200 fish images that were manually-segmented into fish-body and background.

was beyond the scope of this research. Instead, the FCN-8s [111] used in this study were the most accurate fully convolutional network. They have a relatively high number of citation relative to other CNNs (more than 4,000 Google scholar citations at the time of writing); thus, some would consider FCN-8 CNNs as common CNN baseline segmentation models.

The FCN-8 model was coded in Python, using the high-level API known as Keras [64] using the machine-learning Python package TensorFlow [65]. The FCN-8s model is a common features-to-segmentation *decoder* CNN, that uses an image-to-features CNN *encoder*. The original VGG16 [53] convolutional layers of the FCN-8 [111] were constructed as encoders. In Keras, the model VGG 16 was trained on 1,000 different objects in ImageNet [113] and was generally known as ImageNet-trained. When CNN models have more training by ImageNet in recognising new object classes, they are often more accurate than random CNN models [114]. Thus, my version of the FCN-8 model, referred to in this study as the Fish Area Segmentation (FAS) model, was made using the convolutional layers in the ImageNet-trained VGG16 model.

In order to train the FAS, the 200 images were used along with the corresponding hand-segmented body masks. The 200 image-mask pairs were randomly divided into 80%-20% training/validating sets. In order to facilitate

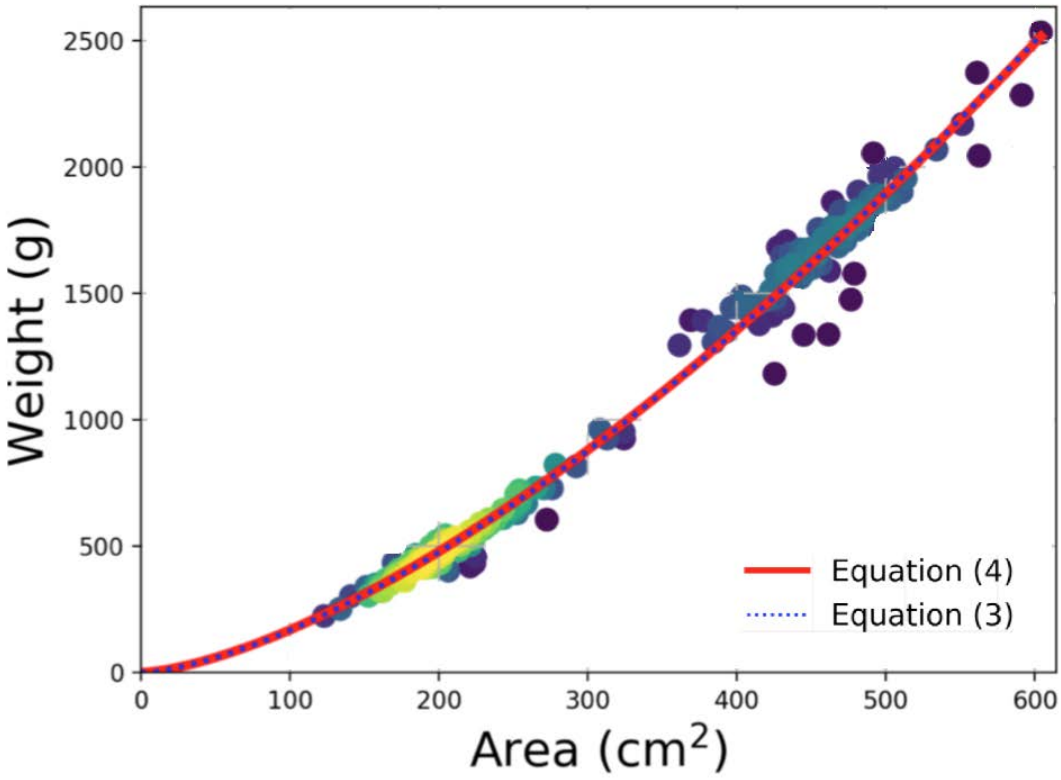

Figure 5.4: Relation between the measured fish weight (M in g) and the manually-segmented fish body image area (S in cm^2) fitted by: Equation 5.4 as $M = 0.1695 \times S^{1.5}$, $R^2 = 0.9819$, $MARE = 5.13\%$; and Equation 5.3 as $M = 0.1622 \times S^{1.5073}$, $R^2 = 0.9819$, $MARE = 5.06\%$. Higher densities of data points are denoted by lighter coluors.

knowledge transfer [114], the FAS model was loaded with the corresponding VGG16 weights, in which the remaining non-convolutional FCN-8 layers had been initialised using a uniform distribution [115]. In comparison to the original FCN-8 with 4,096 neurons in [111], the number of new neurons in the first two FCN-8 decoder layers had been reduced to 512. The need to recognise and segment only a single class of objects, *i.e.*, fish body, explained this radical decrease in neurons. In the last layer, the *sigmoid* activation function was used. The VGG16 encoding layers in FAS are fixed and omitted from training, as this training set has only a limited number of photos. A weight decay of 1×10^{-4} was regularised for the remaining weights (with the exception of biases). The photos and masks for training and testing were reduced to 1 mm per pixel. For each training pass, the image-mask pairs were extensively aug-

mented. In particular, python-opencv was used for augmentations (see Figure 5.5). For each fish image, I randomly shifted each colour channel in the ± 12.5 range, randomly cropped, randomly rotated, and randomly flipped images both horizontally and vertically.

In order to improve segmentation, the following loss function was adopted,

$$loss\left(Y_{gt}, Y_{pred}\right) = 1 - dice\left(Y_{gt}, Y_{pred}\right) + bc\left(Y_{gt}, Y_{pred}\right) \tag{5.5}$$

where: Y_{pred} and Y_{gt} were the predicted and ground truth (*i.e* segmented- by-hand) 480×480 masks; $bc\left(Y_{gt}, Y_{pred}\right)$ was the standard binary cross-entropy loss function; and $dice\left(Y_{gt}, Y_{pred}\right)$ was the Dice coefficient [116] ranging between 0 and 1 (the latter for identical Y_{pred} and Y_{gt}). The last layer used the *sigmoid* activation function; thus Y_{pred} predictions for each pixel ranged from 0 to 1. The ground-truthed data Y_{gt} were encoded as 0s for the background pixels and 1s for the pixels within the fish body area. The losses for training and validation sets were averaged across all pixels and all corresponding images when obtaining the total loss for each epoch of training and validation.

Figure 5.5: Example of image augmentation: (left) the augmented image, (centre) the corresponding augmented binary mask of the fish body (without fins), (right) both the image and the superimposed mask.

As a training optimizer, Adam [66] was used. The Adam learning rate (lr) was set to $lr = 0.001$, whereby the rate was halved after every sixteen epochs when the total validation loss did not decrease. Training in batches of eight images was stopped if the validation loss did not decline after 32 epochs, where the loss of validation was calculated using a series of images

and masks that they did not use to train the FAS model. During training, the FAS model has been continuously saved with the least running validation loss. Also, if the training was aborted, the initial learning rates of $lr = 0.5 \times 10^{-3}$ and $lr = 0.25 \times 10^{-3}$ respectively were restarted twice (from the previously saved FAS model). Notice that the previous augmentation steps were also used to augment validation pictures and to avoid indirectly overfitting to the validation set.

5.3 Results and Discussion

On a Nvidia GTX 1080Ti GPU, it took two to three hours to train the FAS model. Once the FAS model had been trained, it was able to process images of size 640×640 at a rate of 30 images per second, using the same GPU. The FAS model was used for multiple trainings with very similar results and had negligible over-fitting, as shown in the comparable final values of 0.063 ± 0.001 and 0.072 ± 0.003 for validation and training sets (means of equation 5.5). The accuracy of training and validation sets was 0.9945 ± 0.0005 and 0.9935 ± 0.0005, respectively. The trained FAS model was used to include all available images (for detailed steps, see Figure 5.6). FAS can be used for images of any size. However, the use of zero-value pads to fill the fixed shape of 640×640 pixels prior to FAS prediction sped up the algorithm in practise. The output *heat-map* of [0, 1] range pixel values was further processed for each image, where values above 0.50 were set to 1 (*i.e.* the body-pixels) and the remaining pixels to 0 (*i.e.* the background pixels). After this segmentation process, the largest non-zero connected area within each image and its associated area were calculated as pixel^2 (*i.e.*, mm^2). Due to the limited scope of this work, I did not include or examine images of multiple fish or overlapping fish.

In Figure5.7, all projected areas have been compared to calculated weights. In order to minimise the mean squared error between predicted and measured weights, the results were matched with Equations 5.3 and 5.4. Some points may be considered outliers, *e.g.* because of human weight errors or because of fish with an odd shape due to poor nutrition, disease or deformity (Figure 5.7). Only about 1% of images from the BR445 set contained human errors, and these were corrected using an automated picture scaling method [110]. A comparable human error rate of 1% could thus be assumed to be present in the weight values as well, as these cannot, unfortunately, be monitored or corrected based on the available data. A significant practical recommendation for the future would be for researchers measuring fish weights to, where possible, photograph both the measurement ruler and digital weight display together in the same picture for each fish.

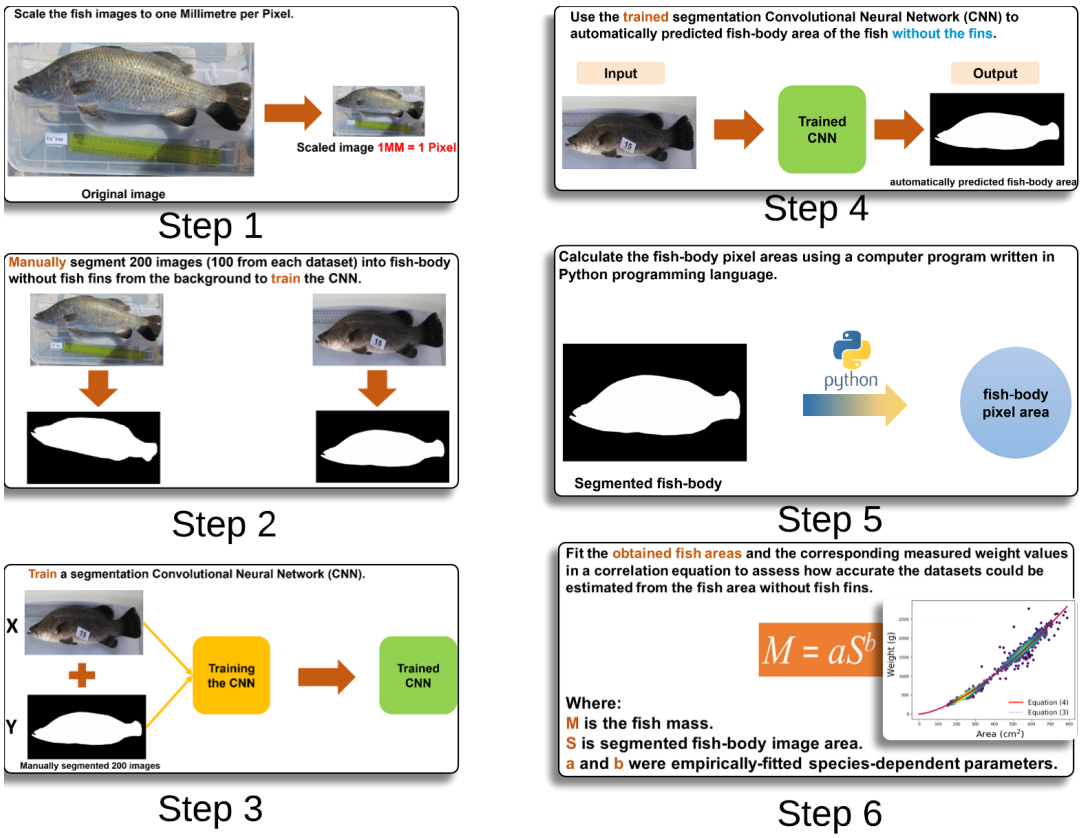

Figure 5.6: Diagram summarizing the full process from training the FAS model to fitting of model results.

The disparities between the fitted outcomes for Equations 5.3 and 5.4 are open to discussion. A better fit certainly does not provide greater predictive accuracy for future unseen samples; see [117] for a detailed discussion. The 5.4 equation was thus arguably more robust to errors because only one parameter was used. The consistency of equation 5.4 has also been verified by applying it in a hand-segmented image training set (Figure 5.4), with the results being exactly the same, $M = 0.1695 \times S^{1.5}$ and $M = 0.170 \times S^{1.5}$, respectively, in over 1,000 automatically segmented photos.

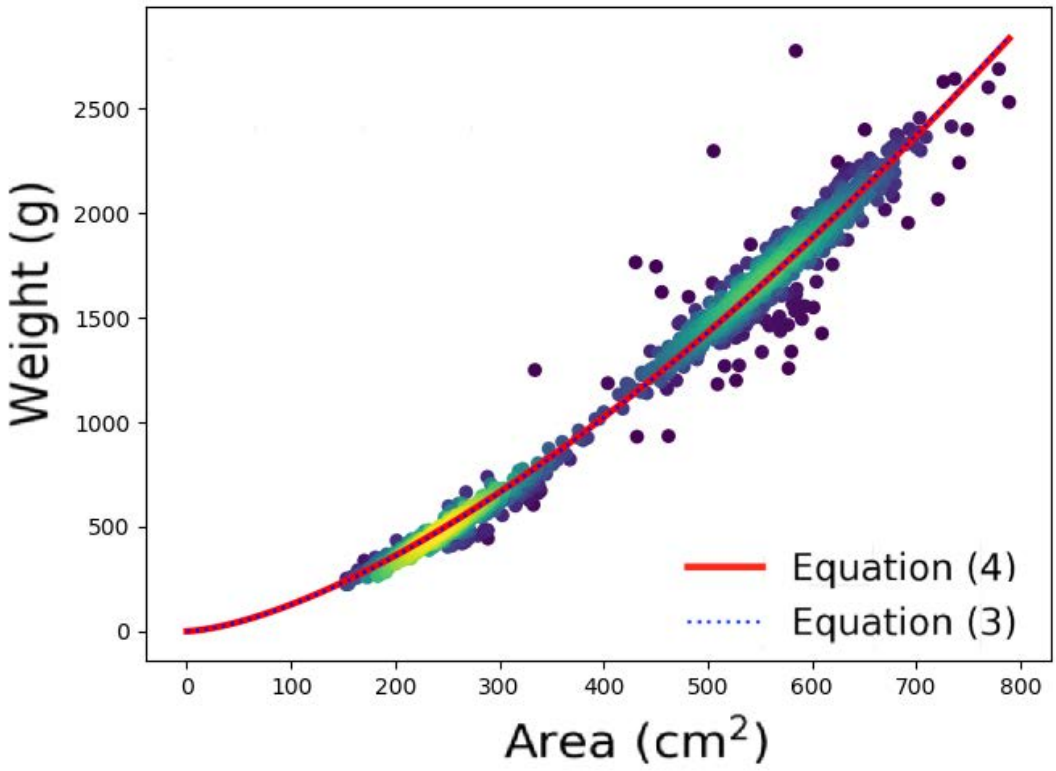

Figure 5.7: Relation between the measured fish weight (M in g) and the automatically-segmented fish-body image area (S in cm^2) as the red line using Equation 5.4: $M = 0.17 \times S^{1.5}$, $R^2 = 0.9804$, $MARE = 5.128\%$; the dotted line is the fitted Equation 5.3, $M = 0.12 \times S^{1.5}$, $R^2 = 0.9808$, $MARE = 4.84\%$. Higher densities of data points are denoted by lighter colours.

5.4 Conclusion

Using the 1,072 digital images of Asian seabass (barramundi, *Lates calcarifer*) for this project, the segmentation CNN was trained on only 200 images in order to automatically segment fish from the background. Figure 5.7 highlights the highly accurate one- and two-factor mass estimation models by plotting the automatically-extracted areas of fish-body against the corresponding manually-measured weights. Given the previously-documented automatic scaling of fish images [110], the introduced automated segmentation approach could considerably reduce fish mass estimation costs and the time required to process them on the industrial scale.

Chapter 6

Automatic Weight Estimation of Fish from Images

This chapter is the second phase of the **second part of the thesis project**: Estimating Fish Weight from Images. This chapter continues the development of methods for the automatic estimation of harvested fish weight from images. In three separate places in Queensland, Australia, about 2,500 weights and associated pictures of fish have been collected for harvested *Lates calcarifer* (Asian seabass or barramundi) . The segmentation Convolutional Neural Network (CNN) of the LinkNet-34 has been trained on these datasets in two instances. The first instance was trained on 200 manually-segmented fish masks with the fins and tails omitted. The second instance was trained on 100 whole-fish masks. The two trained CNNs were then used on the rest of the images, creating segmented masks automatically. Around 1,072 area-weight pairs were fitted from both the first and the second places using one-factor and two-factor simple mathematical weight-from-area models, where the values for fish area were obtained from the automatically segmented masks. The one-factor CNN that included fish masks with fins had the best mean absolute percentage error (MAPE) MAPE = 5.84% when applied to 1,400 test pictures from the third location. CNNs were also trained using direct weight-from-picture regression, where the CNN trained for fish masks with no fins was deemed the most accurate, with an associated MAPE = 4.77% for the test image set.

The content of this chapter is written by the candidate in this thesis, and he also wrote the code, annotated the dataset, performed the experiments.

6.1 Introduction and Related work

The quality assurance of marine products is of high importance for both fish farmers and their customers. The value, price and best before date is usually determined by the quality, freshness and authenticity of fishery products [118; 119; 120]. Thus, fish farmers are interested in all possible ways of maximising their product's quality and increasing their farm's profitability [121]. Monitoring of fish and fish products at various stages may result in increased productivity and profitability for farmers, and closer monitoring of farmed fish can result in earlier detection of disease and stress in fish that often manifest through the serious deformation or malformation of fish vertebrae [122]. In other words, by selecting an appropriate method for assessing the status of fish and fish products, more effective management can be achieved at different stages of fish growth and thus the quality of fish products in aquaculture can potentially be improved. Normal testing and assessment methods for fish health are usually time-intensive, intrusive, costly and permanent. Therefore, it is important to use quick and cost-effective methods that avoid stress from reared fish during various phases of cultivation.

Aquatic products may be categorized into two groups, namely external physical attributes (e.g., morphology, size) and internal attributes (e.g., chemical structure, taste, smell) [123; 124]. Physical attributes such as size, shape, weight, colour and texture are very important and often easier to observe than internal attributes, and shape the first visual impression of consumers that the fish product is acceptable and satisfactory. Therefore, external attributes need to be evaluated and measured closely by fish farmers. For effective aquaculture farm management, monitoring of fish mass is essential. Fishermen can measure the regular feed ratio and fish inventory density by measuring details regarding fish mass. The harvest and classification of fish depend also on the mass of fish and the mass distribution of fish within the population.

Fish weight can also be calculated by determining fish length. On a linear scale, fish length explains about 99% of the variation in fish weight ([125]). Additionally, surface area obtained using SVM (support vector machine) can also be used to predict fish weight [126]. For example, the weight of rainbow

trout (*Oncorhynchus mycosis*) is predicted well by surface area [127]. Finally, weight has also been estimated in *Clupea harengus* using Stereovision systems, which use multiple 2D and 3D features within video and images in order to accurately estimate weight and at high speed [128].

There has been a long interest in the interaction between fish body shape and mass [129; 99; 130]. The weights (W) of fish based on their total length (L) and a constant (q) that is associated with total fish volume and their specific gravity were estimated by the equation $W = qL^3$. However, due to allometric growth (i.e. non-linear increases in fish weight relative to other portions of the fish), [99] the Cubic function is replaced with a variable exponent (n): $W = qL^n$. The conventional lateral profile dimensions of salmon were measured manually by [131], using 52 parameters multi-factor model to accurately predict fish weight. Additionally, the growing use of aquaculture image processing in the mid-1980s [132] provided the opportunity for the automatic evaluation of the weight of fish without the necessity to remove fish from the water. Up to now, various techniques for automated mass assessments of fish outside and within their aquatic environment have been used. Other methods include [133], who used the projected trunk area for turbots, *Scophthalmus maximus*, and a logarithmic relationship between weight and fish area. [134] extracted computer vision measurements of the fish length from the binary image of the fish moving on a conveyor. In comparison with manual measurements, the error in the estimation of fish length using this application of computer vision was ±3%. In [135], the fish mass from pictures was also easily estimated. The correlation coefficients between weight and shape for grey mullet, carp and st. Peter's fish were 0.954, 0.986 and 0.986, respectively. [136] used a SVM to describe to a bias of only ±3% for fish weight estimated using fish shape/form parameters. The pictures were taken both from the top and side perspectives, and the vector support system was trained using 13 form parameters. Both of [103; 126] used image processing to predict the weight of different salmon species (Alaskan, Pink, Red, Silver, Chum) with coefficients of determination ranging from 0.93 to 0.99 between weight and shape. Therefore, computer vision is a powerful way of accurately and in real-time, measuring the mass of fish. There is, however, no general method

for estimating the mass of each species and the optimum relationship for each individual species thus needs to be estimated.

Using weight and fish body surface area measurements from 120 jade perch (*Scortum barcoo*), three mathematical relationships of mass were developed and tested in [137]:

$$\text{Polynomial: } M = a + bS + cL + dH, \tag{6.1}$$

$$\text{Power curve: } M = aL^b, \tag{6.2}$$

$$\text{Linear: } M = a + bS, \tag{6.3}$$

Where S is the surface area of the fish body (with or without fins), H is the height of the fish, and all lowercase symbols (a, b, c, d) are fit based on least sum of squares methods according to the true measurements. The polynomial model (eq. 6.1) performed the best on a testing set of 64 images, and reached MAPE $= 5\%$ for the contours of fish with and without fins. The only other comparable model that was found was the Power Curve (Eq. 6.2) that achieved MAPE $= 10\%$ for without fins contours and MAPE $= 12\%$ for fine-tuned contours. Similarly, [138]'s third-ground polynomial model (MAPE $= 11.2\%$) was consistent with and comparable to the power-curve model results of [137] (Eq. 6.2, MAPE $= 10 - 12\%$).

Using only the surface area S (Eq. 6.3), Viazzia *et al.* [137] reported MAPE $= 5 - 6\%$, while Konovalov *et al.* [19] fitted the following two mathematical models for harvested Asian seabass (*Lates calcarifer*, also known in Australia as barramundi):

$$M = cS^{3/2}, \quad c = 0.170, \tag{6.4}$$

$$M = aS^b, \quad a = 0.124, \quad b = 1.55, \tag{6.5}$$

Where the mass (M) was measured in grams and the surface area (S) of the fish body was in cm^2 for 1 mm-per-pixel image. For 1,072 different fish photos of two distinct barramundi farms (Queensland, Australia)[107], the MAPEs were 5.1% and 4.5%, both for the single-factor (Eq.6.4) as well as for the two-factor (Eq. 6.5) versions. Generally, fitting parameters of a, b, and c appears to be always species-dependent [139; 140].

This project aims to expand methods for automatically estimating the weight of harvested fish. The following two practical and technical issues were dealt with, in particular:

1) Are models trained on datasets with fish masks excluding the fins and tailfins better than models trained using whole-fish masks (e.g. in [19])? Entire surface areas of the fish are easier to extract, rather than excluding fins that are not easily defined in some videos and images [103; 137]. Therefore, fish segmentation based on fish masks without fins and tailfins is only warranted if the extra complexity in training the CNN provides a great improvement to the mass estimation model [19].

2) The way in which Eqs. 6.4 and 6.5 are consistent in fish mass prediction when the different barramundi photos are added is examined. They both used semantic segmentation FCN-8s CNN [111; 97; 141], which are substituted here by more modern LinkNet-34 CNNs [142; 22] and are compared to FCN-8s networks in [19] to check stability and precision of the automated segmentation of fin and no-fin fish masks.

A pipeline for weight estimation is presented so that video frames could be processed quickly as single pictures in real-time for frame sizes up to 480×480 pixels. Similar to [134], a standard conveyor can be fitted with a video camera for weight estimate analysis to use within the aquaculture industry. Conveyor videos can now be processed off-site, making the estimation method financially viable and technological advancements have increased the accuracy of these tasks by handling the frames in higher resolution, thus making this method increasingly feasible. By placing or putting a measuring ruler (or a known scale object) on or next to the conveyor, appropriate scaling of fish is easily achieved.

Keeping in mind the importance of monitoring fish and the simplicity of this measuring framework, this project is of great importance for potential industry-level deployment purposes.

6.2 Materials And Methods

6.2.1 Harvested Barramundi Datasets

Three data sets were first introduced in [107] and used in this chapter. The first dataset was Barra-Ruler-445 (BR445), which included 445 images with manual weights measured between $1 - 2.5$kg. BR445 has previously been used in [19; 110; 143]. The second data set was BW1400, and contains 1,400 harvested images of barramundi with weight values between $0.15 - 1.0$kg. The third dataset was Barra-Area-600 (BA600), containing over 600 pairs of images and weights (used in [19])

In Figs 6.1 and 6.1, a couple of examples from each dataset are presented, both originals and in greyscale.

The BR445 and BA600 images were taken outdoors in natural sunlight, while the BW1400 images were taken indoors under artificial illumination. Note that BR445 and BW1400 images have the same white holding plate (Figs. 6.1a-d), with BR445 images having a slight blue tint (Figs. 6.1a-b). All photographs were converted to grayscale prior to the computer vision tasks, to reduce the importance of transient colours for training and testing (Fig. 6.2), and normalized in [0,1] numerical values.

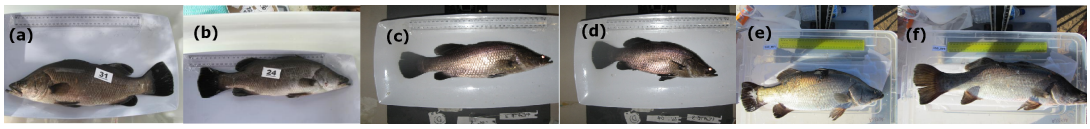

Figure 6.1: Samples of original images from the used datasets: BR445 (a) and (b), BW1400 (c) and (d), BA600 (e) and (f).

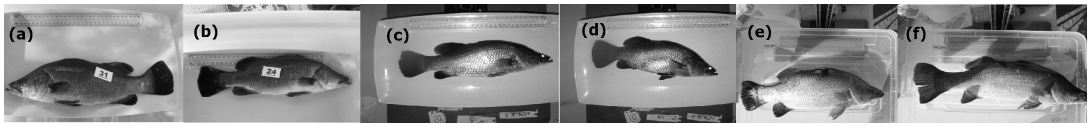

Figure 6.2: The same samples as in Fig. 6.1 converted to grayscale and enhanced by Enhance Local Contrast (CLAHE) [6].

6.2.2 Segmentation of Fish Surface Area

In this project, together with the corresponding fish images, 100 mask pairs from BR445 and 100 from BA600 have been scaled to 1 mm-per-pixel; see an examples of the same 200 no-fins masks from [19] in Fig. 6.3. To examine the effect of fins/no-fins in the fish masks, 100 additional fin masks (50 from each BR445 and BA600) were manually segmented; see an example in Fig. 6.3(h). The lower number of whole-fish masks (with fins) was justified by the anticipation of the whole-fish segmentation being a much simpler learning problem.

The Fully Convolutional Network from [111], FCN-8s, has been trained and deployed on 200 with fin masks previously in [19]. Although FCN-8s were a significant theoretical advance in the field ([111; 97]), FCN-8s are still less accurate than the most recent CNN segmentation models using U-Net ([141]). In addition, only 200 no-fin masks of 1,072 images were manually-segmented ([19]), thus, the exact accuracy of the FCN-8s segmentation process for other non-segmented pictures was not evaluated. Therefore, in this chapter, I tried to evaluate the accuracy of the original findings recorded using the FCN-8s by relying on the U-Net family of CNNs, which are, at least in theory, more accurate as a segmentation model.

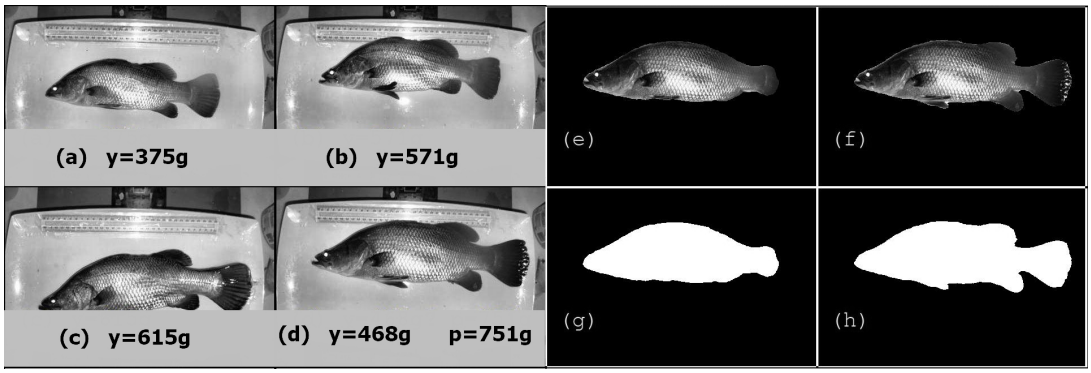

Figure 6.3: An example of weight measuring error in the BW1400 dataset: (a-c) the correctly measured reference images with y weight values; (d) the identified recording/measuring error (predicted $p = 751$g); (e) the mask without fins (including tailfin) for the fish in (d); (f) the whole fish mask for the fish in (d); (g) the mask without fins and tail; (h) the whole fish mask.

A variant of U-Net [141], LinkNet-34 [142], was chosen for this project, which includes the feature encoder ResNet-34 [144] and the [22] implementation of PyTorch. The reasoning for selecting LinkNet-34 over other models was twofold. First, CNN results are a challenge in many cases in terms of reproducibility. This was alleviated by the use of the ResNet-34 CNN (which can be found in the PyTorch distribution) together with the LinkNet-34-style decoder [22]. The second critical factor was that, in the Endoscopic Vision Sub-Challenge: Robotic Instrument Segmentation (MICCAI 2017), LinkNet-34 offered a high balance between speed (also tested within this project) and very high accuracy [22].

6.2.3 Training Pipeline

A training pipeline [19] was maintained, to ensure the reproducibility of the method.

- The first 200 no-fins masks and 100 with-fins masks were divided into 80% and 20% training and validation folds/sets
- ResNet-34 layers were loaded with ImageNet [11] weights they were trained on, in order to speed up the training process through knowledge transfer [114]. In the final output layer, the *sigmoid* activation function was used
- 1×10^{-4} weight decay was applied to all trainable weights
- All images and masks were scaled up to 1 mm-per-pixels
- To minimise overfit for both training and validation, all image-mask pairs were randomly: rotated, cropped, flipped horizontally, and/or flipped vertically.
- Training was done in batches of 8 images
- Adam [29] was used as a training optimizer

Compared to [19], the following training steps were improved. As per [22], the loss function (Eq. 6.6) was replaced by (Eq. 6.7):

$$loss(y, \hat{y}) = bc(y, \hat{y}) + (1 - dice(y, \hat{y})) \tag{6.6}$$

$$loss(y, \hat{y}) = bc(y, \hat{y}) - \ln(dice(y, \hat{y})) \tag{6.7}$$

where y was a target mask, $\hat{y}$ was the corresponding LinkNet34 output, $bc(y, \hat{y})$ was the binary cross entropy, and $dice(y, \hat{y})$ was the Dice coefficient [116]. Additional grey-to-color trainable conversion layer on the front of the LinkNet-34 [145] has been added to reuse the ImageNet training ResNet-34 encoder. Image blurring (kernel sizes 3 or 5 pixels) or Enhance Local Contrast (CLAHE) [6] were applied with 0.5 probability each, in addition to the original augmentations [19]. LinkNet-34 has eliminated the need for freezing the ImageNet-trained encoders' weights [19] for advanced segmentation CNNs (as compared with FCN-8 CNNs) with grey scale images (and other augmentations).

6.3 Results and Discussion

In this chapter, I developed an important benchmark practises to automatically estimate the weight of harvested fish from images. This objective was addressed using *weight-from-area* and *weight-from-image* models.

6.3.1 Estimating weight-from-area by mathematical models

The first step was to see if the mathematical models used to estimate fish mass (M) from fish surface area (S) in the image were correct and reliable for industry use (See Eqs. 6.4 and 6.5). Two mathematical models were examined, see Eqs. 6.4 and 6.5.

6.3.1.1 Which is better, with or without fins?

I also examined whether model accuracy was improved by using only the fish body (herein referred to as "*no-fins*", see associated rows of Table 6.1 for results), rather than using a whole fish mask that included all fins and the tailfin (herein referred to as "*whole*", see associated rows of Table 6.1). Results for the one-factor model in rows 1 and 2 (cells highlighted in green) showed that for no-fins models using Eq. 6.4, both the coefficient of determination (R^2) was higher (i.e. greater model fit), and the mean absolute percentage error (MAPE) was lower (i.e. greater predictive accuracy) than other combinations. Note that R^2 increases based on the number of parameters included in the model, and thus, will always be higher for models using more parameters, and thus is best to compare only between whole and no-fins mask types based on the same model equation. Thus, for models using masks with no-fins, the two-factor model (Eq. 6.5) was most accurate, according to MAPE values.

6.3.1.2 The logarithmic scale approach

The original fit was not performed on a logarithmic basis in the previous chapter (row 3 of Table 6.1), which implies that large weights contributed more to model fit (compare top and bottom rows in Fig. 6.4) by reducing

Table 6.1: Mass estimation models

Mask type	**Model** Fitted or trained on BR445 and BA600	**Fit** R^2	**Fit** *MAPE* [%]	**BW1400** *MAPE* [%]
1. whole	$c = 0.12$	**0.963**	**6.15**	5.84
2. no-fins	$c = 0.17$	**0.965**	**6.04**	7.63
	Eq. 6.4, log-MSE fit			
3. no-fins	$c = 0.17$	0.9804	6.128	9.21
	Eq. 6.4, MSE fit [19]			
4. whole	$a = 0.09, b = 1.5$	**0.964**	**5.42**	7.47
5. no-fins	$a = 0.1, b = 1.5$	**0.971**	**5.31**	11.34
	Eq. 6.5, log-RANSAC fit			
6. no-fins	$a = 0.12, b = 1.5$	0.9808	6.84	12.17
	Eq. 6.5, MSE fit [19]			
7. whole	LinkNet-34R		4.68	12.06
8. no-fins	LinkNet-34R		4.57	**4.77**

the mean squared error (MSE). Rows 1 and 2 were fitted to the logarithmic-scale [137] (top row of Fig. 6.4) to decrease the MAPE from 6.128% to 6.04% (rows 3 and 2, respectively, of Table 6.1). Thus, the MAPE improvement was less noticeable between when MSE was done directly on the area and weight values rather than their logarithms, for no-fin masks. Fig. 6.4 demonstrates how the no-fins (top sub-figures) and whole (bottom sub-figures) distributions were qualitatively comparable, where lighter (more yellow) colour indicates a higher density. The results in Fig. 6.4 may help to explain why some previous studies have not found differences when using no-fin masks [103].

6.3.1.3 Robust fit and Outliers

A number of outliers have also been found in Fig. 6.4. One solution to possible outliers was the use of a robust [146] linear regression, which had been done in the study by changing the two-factor model (Eq. 6.5) to the logarithmic scale through the RANSAC algorithm [147] (top row of Fig. 6.4, see rows 4 and 5 of Table 6.1). The fitting of two factor coefficients of b differs by 1% between the models with-fins ($b = 1.56$) vs. no-fins ($b = 1.57$), which indirectly confirms that the fit of the RANSAC is more robust to outliers. The fit of automatically segmented fish shapes with no fins and tails also produced the best MAPE = 5.31% of all models examined.

6.3.1.4 Image acquiring method

The no-fins MAPE = 5.31%, (see Table 6.1 cells filled with yellow), showed an increase in MAPE values of around 0.11% relative to the one-factor model. The image scales were precise to about 1-2% difference in MAPE value, where the scales were taken from the rulers shown in each frame. The visual distortion in the ruler (per ruler long) often produced up to 1% different pixels between the top and bottom markings. If feasible, creating image masks that do not include fins or tailfins may thus be better for the purposes of building more accurate mass estimation models.

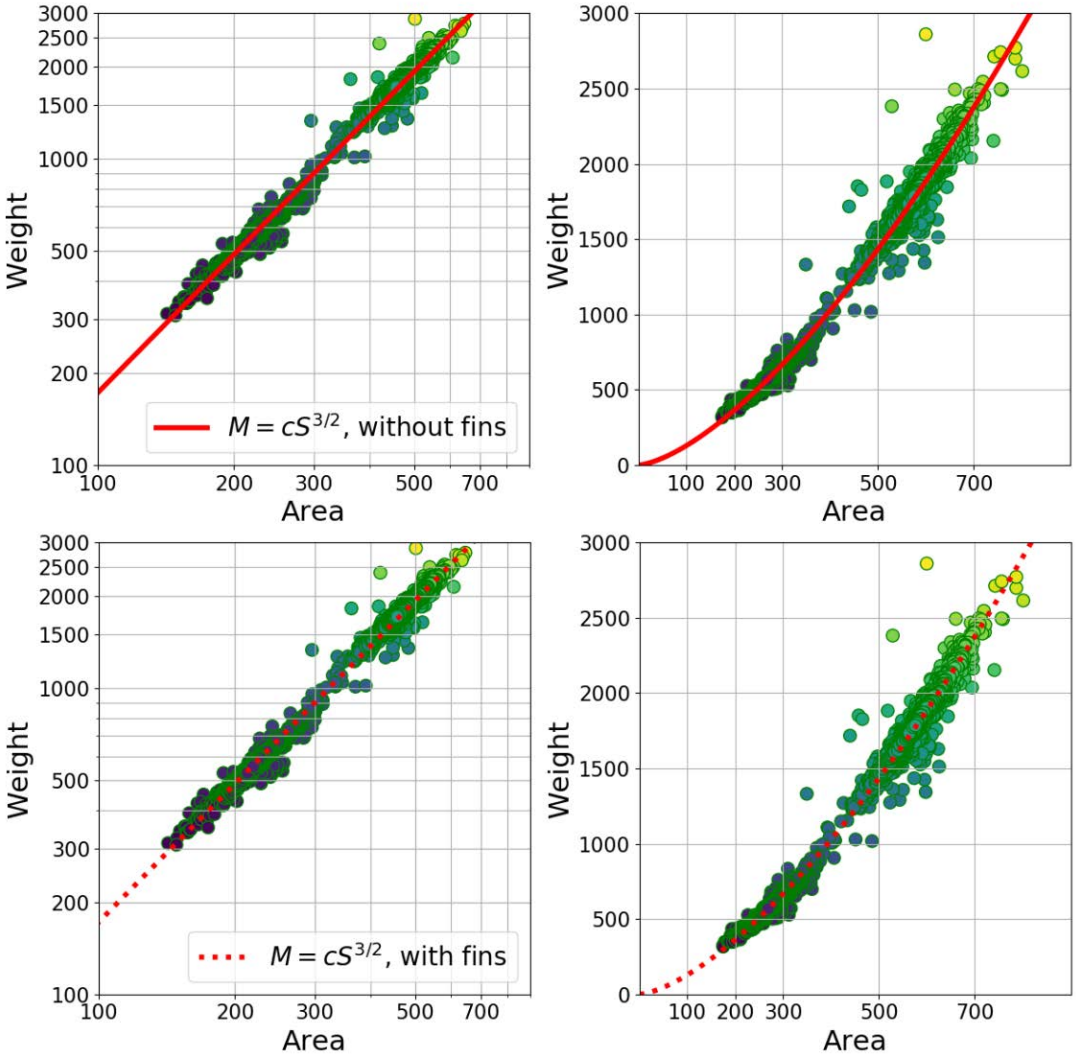

Figure 6.4: Relation between the measured fish weight and the automatically segmented fish area for the combined BR445 and BA600 datasets: no-fins (fish masks without fins or tailfins) on top and whole fish (fish masks including fins and tailfins) on the bottom.

6.3.2 Direct weight-from-image estimation

In previous sections, I showed how a fish image can be segmented using zero-values to indicate background pixels and one-values to indicate pixels containing a fish body, by using CNN segmentation LinkNet-34 with or without fins. The LinkNet-34 sigmoid output as a value of one (foreground pixels) was not fine-tuned and the threshold was left at its default value of 0.5. Then, the total number of non-zero pixels was used to obtain S for a fish area that was fitted by Eqs.6.4 or 6.5 to a corresponding fish weight M. Each foreground pixel of a fish was expected to contribute to the total fish mass equally. In 2017, Standley *et al.* [148] reported one of the first CNN image-to-mass conversion applications to achieve MAPE $< 1\%$ on more than 1,300 test images of generic everyday objects and home objects, where there were about 150,000 images in the training set. The direct converting of the segmented mask into weights via the regression version of LinkNet-34, known as LinkNet-34R, was therefore interesting and worth comparing to the previously-used methods. LinkNet-34R obtains weights by adding all the sigmoid outputs (y_s) from LinkNet-34 to the logarithmic scale without a threshold:

$$y_r = log(y_s + 1) \tag{6.8}$$

Thus, when y_s=1, the images without detected foreground fish masks are added to assign a zero mass value. The fish images (not just masks) are also automatically segmented, see examples in Fig. 6.3(e) and 6.3(f). To ensure that weight values predicted from the CNN are associated with the fish and not with something else in the image, segmented images are used as LinkNet-34R inputs (with or without fins). The corresponding log-scaled fish weights then replace y_s by weights of M with the same Eq. 6.8. The LinkNet-34R training pipeline stayed the same as in LinkNet-34, with the only difference being that the images were not resized randomly. Specifically, the random scaling of 80%-120% for LinkNet-34 was used, but not for LinkNet-34R. As the LinkNet-34 had already been trained to correctly detect fish, the LinkNet-34R version was fit with parameters of LinkNet-34 and trained from learning rates reduced by 10 when fine-tuning the model.

Model errors were examined using numerical experiments and some image and/or recording/measurement errors were identified in around 1–2% of the image weight pairs. For example, for analysis of images with identical scaling (1 mm-per-pixel) in Fig. 6.3(a)-(d), the weight estimated for the case of image (d) would exceed 615g and predicted to be 751g while it was reported to be 468g, likely due to a measurement error. These errors have been specifically excluded from the data set BW1400, but not from the data sets BR445 and BA600 to allow a clear comparison of the project's [19] performance. As per the *image2mass* study [148] and since some BR445 and BA600 data sets have remained outliers (Fig. 6.4), the metric MAE was used as the loss function to the LinkNet-34R regression training model. If MSE had been used, the outliers would have affected the end fit more [146]. For training and validation subsets, all the 1,072 available BR445 and BA600 segmented image-weight pairs have been randomly divided into 80% and 20% folds, respectively, and the training subset used for training LinkNet-34R models. In rows 7 and 8 of Table 6.1, the MAPE values during validation using the testing fold are reported (4.68 and 4.57%, respectively).

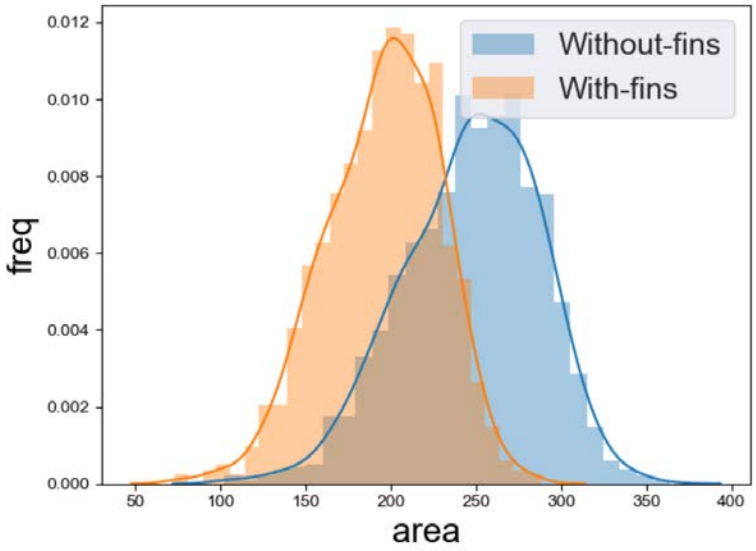

Figure 6.5: Normalized distributions of automatically-segmented mask areas in the BW1400 images

6.3.3 Models predictive performance

The use of mathematical models and neural networks for estimating fish weight has little practical value unless new fish weights from different locations can be predicted by the models. The predictive accuracy of the BR445 and BA600 models is reported in the last column of Table 6.1, and it is also used with the previously-unseen BW1400 dataset. Metrics such as R^2 are less useful in practical industrial applications, so only MAPE has been reported hereafter. The following is an interpretation of the somewhat conflicting results of these MAPE values.

6.3.3.1 Better MAPE values for Whole-fish mathematical model

For the unseen BW1400 images, as opposed to the corresponding MAPE values for no-fins (7.63% and 11.34%), both one- and two-factor models using *whole-fish* achieved substantially better MAPE values (5.84% and 7.47%). the no-fins models (see rows 3 and 6 in Table 6.1). The results from [19] were also consistent with this result, when developed models were applied to an entirely new and previously unseen dataset.

6.3.3.2 Errors in no-fins masks had a larger effect

When attempting to understand why whole-fish models are better predicted, it is important to consider that the lower (pelvic) fins had often overlapped with the body area of the fish, and thus, the no-fins CNN subtracted this area from the fish body (see Figs. 6.2(c), 6.2(d) and 6.3(b)). Fish mask with no-fins were, on average, less than 20% of the corresponding entire-fish areas (see Fig. 6.5). Consequently, incorrect reductions of no-fin masks were more significant than variations in the whole-fish masks (e.g. due to overlapping pelvic fins).

6.3.3.3 One-factor Vs Two-factor models

An analysis was carried out on how one-factor models were much better than the two-factor models (one-factor models: 5.84% and 7.63% MAPE, rows 1 and 2 in Table 6.1; two-factor models: 7.47% and 11.34% MAPE, row 4 and

5). The best fits for the two-factor models (5.42% and 5.31%, rows 4 and 5) were possibly due to the overfitting of the training data sets, which were most compatible with the one-factor model when refit according to all available data in BR445 and BA600 samples in [19].

6.3.3.4 Direct weight-from-image CNN regression

Eqs. 6.4 and 6.5 were based on the hypothesis that each fish pixel was equally allocated towards the total weight of the fish. The previous results showed that these models could be very rough approximations. However, the LinkNet-34R CNN models performed using a highly non-linear weight conversion of the segmented fish images, but at the cost of an ease-of-interpretability, in contrast to Eqs. 6.4 and 6.5. Nearly identical validation MAPE $=$ 4.57% and test MAPE $=$ 4.77% were obtained using the no-fins version (row 8 of Table 6.1). The whole-fish version, showed some indications of overfitting for the two-factor model: MAPE $=$ 4.68%, while predictions for the testing set were much less accurate MAPE $=$ 12.06% (row 7 in Table 6.1).

In the future, a comprehensive analysis may be needed to determine how th LinkNet-34R CNN models determined weight predictions. In the no-fins fish images, for example in Fig. 6.3(e), the contour was smooth, so the LinkNet-34R was required to calculate the weight by using features from the fish in the image. However, the contours using the whole-fish masks were more complex, and were thus more likely to be specific to individual training images, indicating overfitting of the training set due to the more than 21 million parameters set by LinkNet-34R (see Fig. 6.3(f)).

6.4 Conclusion

The mass estimation of objects from images is an evolving computer vision task with represents an important contribution to industrial use [148]. I showed how typical CNN segmentation ("*off-the-shelf*") such as LinkNet-34 [22], can be trained effectively using: (i) image-mask pairs for as little as 100-200 images in the training set; (ii) a schedule of linear learning rates; and (iii) reduced ImageNet training encoder learning rates (ResNet-34). Fish masks were automatically segmented and fitted with simple mathematical models to achieve MAPE values between 5-11%. These MAPE values are comparable to other studies, e.g. [138; 137]) on 1,400 test images not used in the fitting procedure and from different geographical locations.

The main question of this research was to evaluate whether a fish mask used in automatic segmentation by CNNs should include fin and tailfin of fish or not. In particular, when models were used on unknown test images from a different geographical area, the two basic mathematical models based on the whole-fish silhouette, in general, performed best (lower MAPEs). The question of model complexity was also addressed, as new test images were best predicted by the simplest one-factor (one-parameter) mathematical model relative to the two-factor model. In addition, models were relatively stable and low MAPE $= 5.84\%$ for the experiments, compared to MAPE $= 6.15\%$ for the training images.

Chapter 7

Conclusion

In this thesis, practical computer vision applications capable of detecting fishes in various fish habitats have been developed, and a large image dataset is presented. The *DeepFish* consists of approximately 40,000 labelled images representing 20 fish habitats collected from remote coastal marine-environments across tropical Australia. Moreover, to evaluate a variety of deep learning methods, I applied the *DeepFish* dataset across four different tasks: classification, counting, localization, and segmentation of fishes. The second method estimated fish weight from an image of harvested fish. I hope that the results from this thesis may contribute to large-scale applications of artificial intelligence that reduces both the cost and time required to determine fish mass within the aquaculture industry. The methods outlined in this thesis are a step towards the development of valuable practical computer vision applications capable of classifying various fish species habitats, determining the presence or absence of fishes, and measuring their weights and sizes. Thus, they represent a valuable contribution to fisheries management, ecosystem management and fish stock conservation programs. This chapter summarizes the thesis's approaches and contributions to the field, and discusses future work (see figure 7.1).

7.1 Summary and contributions

The field of computer vision has witnessed rapid advances over the last few years. For image classification (a core visual recognition problem), ImageNet's Large-Scale Visual Recognition Challenge serves as a highly-cited benchmark

exemplifying this progress. The Challenge saw a decrease in the top-5 error rates from 27% in 2010 to 2.25% in 2017. The top-5 error is the percentage of test examples for which the correct class was not in the top 5 predicted classes, and is a difficult task to achieve considering ImageNet has 1,000 classes. This performance matches human recognition abilities and even exceeds it in some more specialized categories, such as the recognition of different dog breeds [11].

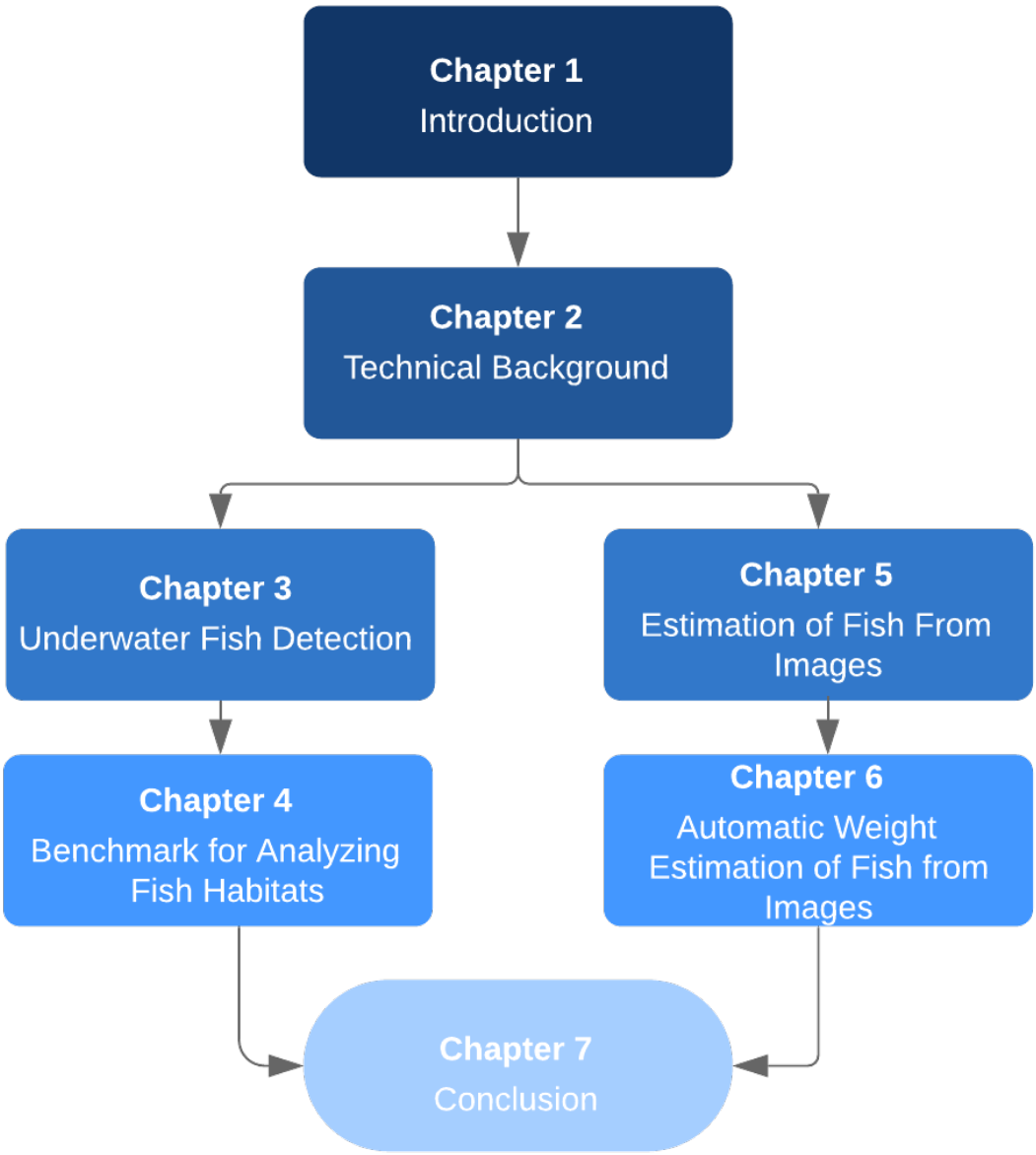

Figure 7.1: Thesis structure and interconnection of chapters

In Chapter 3, a CNN-based fish-detector was developed to train to a project-specific dataset of underwater fish/non-fish images from 20 different habitats with relatively small numbers of images (4,000). Above-water photos (VOC2012 [149]) were also used, along with 17,000 negative (i.e. missing fish) images. An additional 27,000 overwater and underwater positive (i.e. having a fish in the picture) were used from two publicly available fish-domain datasets. The resulting CNN binary classifier (fish/not-fish) trained on a portion of these images produced 0.17% false positives and 0.61% false negatives in the 20,000 negative and 16,000 positive holdout test image set of the project.

In Chapter 4, a benchmark called *DeepFish* was developed based on a large image dataset of remote underwater video collected from remote coastal marine-environments of tropical Australia. The purpose of this benchmark was to motivate specialized algorithms that can automate the task of fish image analysis. The *DeepFish* dataset consisted of approximately 40,000 labelled images representing 20 different fish habitats across Australia. As baselines, I also evaluated a variety of deep learning methods across four tasks: (1) classification, (2) counting, (3) localization, and (4) segmentation of fishes. The goal of the classification task was to identify which images contain fish. For the counting tasks, the goal was to determine the number of fish in each image. The goal of the localization task was to identify the locations of the fish in the images. For the segmentation task, the goal was to extract the shape and size of the fish that are present in the image. These tasks are important for researchers assessing fish habitats. For instance, localization can help experts find fish that are in highly occluded or turbid areas, whereas segmentation can be used to estimate the size of fish passing into the camera's range. The experimental results show that the developed deep learning methods achieve compelling results. This suggests that a tool built on deep learning may be possible, which would significantly reduce the amount of human effort needed to analyze fish habitats. *The dataset and the code will be made public, which will hopefully inspire further research into this area, through developing more powerful and flexible algorithms*

In Chapter 5, I developed a method using a Segmentation Convolutional Neural Network trained on 200 images, which were used to automatically

segment fish body area from the background in all of this study's 1,072 digital images of Asian seabass (barramundi, *Lates calcarifer*). The fish-body areas were automatically extracted and used to predict weights of fish using one- and two-factor mass-of-area estimation models achieving high accuracy, such as model $R^2 = 0.9828$, $MARE = 5.58\%$, and $R^2 = 0.9834$, $MARE = 4.53\%$, respectively.

Finally, the focus of Chapter 6 was to continue developing methods for the automatic estimation of harvested fish weight from images. I showed how CNN segmentation through typical "*off-the-shelf*" models such as LinkNet-34 [22] can be trained effectively using: (i) image-mask pairs of only 100-200 images; (ii) a schedule of linear learning rates; and (iii) by reducing the ImageNet training encoder's learning rate (ResNet-34). Fish masks were automatically segmented and fit with simple mathematical models to achieving MAPE values between 5-11% (these values are comparable with other studies, e.g. [138; 137]) on 1,400 test images not used in the fitting procedure and from an entirely different geographical location. The main question of this research was to evaluate whether a fish silhouette automatically segmented by the CNNs should include the fin and tailfin or not. In particular, when used on unknown test images from a different geographical area, the two basic mathematical models based on the whole-fish silhouette generalized better (lower MAPEs) than fish masks using no fins/tailfins. Additionally, the simplest one-factor (one-parameter) mathematical model performed better than the two-factor model on the new test images. Finally, the results were very stable, with low MAPE $= 5.84\%$ for the experiments, compared to the MAPE $= 6.15\%$ for the training images.

7.2 Future Work

The research presented in this thesis demonstrates methods and techniques that are developed to produce real-world applications for marine habitats using computer vision. There are several extensions to the methods that were beyond the scope of this thesis that can be considered. This section discusses two of these extensions.

When analyzing fish habitats, both a dataset and a method are required. **The dataset** is a collection of videos of underwater fishes with different annotations for various computer vision tasks (e.g., classification, counting, localization and segmentation). The collection of an underwater fish data can be used to help researchers in Australian fisheries and the rest of the fisheries community in training and testing their computer vision methods on a public dataset. This dataset will be unique if it covers the full diversity of different aquatic habitats in the coastal/marine environment. These kind of habitats and environments are common across the tropics; therefore, this could be done anywhere in the world. It would be valuable if scientists and environmental managers had this kind of data from a broad diversity of habitats. The dataset utilizes remote underwater video (RUV) recordings and is thus a promising tool for fisheries, ecosystem management and conservation programs [30; 31] in coastal marine environments of tropical Australia.

The method involves the development of a computer vision system capable of classifying various fish species habitats, determining if contain fishes, analyze where they are, how many, and measure their sizes. The output of this application is to provide a knowledge of habitats that need to be protected, and enhance our understanding of the impacts of environmental change on fish populations. My proposed method would provide valuable information (i.e. fish numbers, location and sizes) to fishery managers, ecosystem managers and conservation officers to aid in maintaining the integrity of fish stocks.

In the second method applied to Asian seabass (barramundi, *Lates calcarifer*), phenotyping of fish via predictive modelling of features measured after slaughter can be assessed using images of fish before being harvested, including 3-dimensional images. Deep learning and computer vision are promising fields for the creation of such models. The resulting models could then be used to design a high-performance phenotyping technique.

In future work, video data may be automatically extracted regarding live fish including morphometrics and weight measurements, as well as fillet yield predictions. Existing fish spawning data collection schemes rely on manual labour and are expensive, and are also susceptible to human error. Thus,

computer vision can be used to develop new solutions for high-throughput phenotyping in commercial environments. Researchers can then analyze characteristics of harvested fish such as their body weight, viscera weight and fillet weight. Most of these measurements can only be taken after slaughter, and so these features can not be determined on live fishes prior to the establishment of predictive models. In addition to fish weights, three-dimensional photographs of fish prior to harvest can be taken and used to build a predictive model. This will create a new phenotyping technique that aims to enhance animal selection by accelerating the collection of data on fish and eliminates the need for slaughtering fish unnecessarily in order to quantify internal features.

References

[1] "Cs231n convolutional neural networks for visual recognition." 2019. [Online]. Available: http://cs231n.github.io/neural-networks-1/. (cited on pages xv and 14)

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. (cited on pages xv, 3, 13, 15, and 17)

[3] K. Anantharajah, Z. Ge, C. McCool, S. Denman, C. Fookes, P. I. Corke, D. Tjondronegoro, and S. Sridharan, "Local inter-session variability modelling for object classification," *IEEE Winter Conference on Applications of Computer Vision*, pp. 309–316, 2014. (cited on pages xvi and 43)

[4] R. B. Fisher *et al.* [Online]. Available: https://bit.ly/2Ex7dnZ (cited on pages xvi, 24, 43, and 44)

[5] C. G. Stierhoff, K., "Rockfish training and validation image dataset," *NOAA Southwest Fisheries Science Center remotely operated vehicle (ROV) digital still images.*, 2013. (cited on pages xvi and 43)

[6] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*. Academic Press Professional, Inc., 1994, pp. 474–485. [Online]. Available: http://dx. (cited on pages xviii, 78, and 81)

[7] M. Sheaves, "Consequences of ecological connectivity: the coastal ecosystem mosaic," *Marine Ecology Progress Series*, vol. 391, pp. 107–115, 2009. (cited on pages 1 and 43)

[8] M. Bradley, R. Baker, I. Nagelkerken, and M. Sheaves, "Context is more important than habitat type in determining use by juvenile fish," *Landscape Ecology*, vol. 34, no. 2, pp. 427–442, 2019. (cited on pages 3 and 4)

[9] M. K. Alsmadi, K. B. Omar, S. A. Noah, and I. Almarashdeh, "Fish recognition based on robust features extraction from size and shape measurements using neural network," *Journal of Computer Science*, vol. 6, no. 10, p. 1088, 2010. (cited on page 3)

[10] G. Wei, Z. Wei, L. Huang, J. Nie, and H. Chang, "Robust underwater fish classification based on data augmentation by adding noises in random local regions," in *Pacific Rim Conference on Multimedia*. Springer, 2018, pp. 509–518. (cited on pages 3 and 4)

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y (cited on pages 3, 80, and 92)

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99. (cited on page 3)

[13] F. Yu, "Multi-Scale Context Aggregation by Dilated Convolutions." 2016. [Online]. Available: http://mdpi.com/abs/1511.07122 (cited on page 3)

[14] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford *et al.*, "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018. (cited on page 4)

[15] D. Rathi, S. Jain, and S. Indu, "Underwater fish species classification using convolutional neural network and deep learning," *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, Dec 2017. [Online]. Available: http://dx.doi.org/10.1109/icapr.2017.8593044 (cited on page 4)

[16] D. Dagneaux, F. Boschetti, R. Babcock, and M. Vanderklift, "Detecting general patterns in fish movement from the analysis of fish tagging data," in *18th World IMACS/ MODSIM Congress, Cairns, Australia*, 2009, pp. 13–17. (cited on page 4)

[17] S. Bleisch, M. Duckham, P. Laube, and J. Lyon, "Identifying candidate causal relationships in fish movement patterns," *Proc. 21st GIS Research UK (GISRUK)*, 2013. (cited on page 4)

[18] D. A. Konovalov, A. Saleh, M. Bradley, M. Sankupellay, S. Marini, and M. Sheaves, "Underwater Fish Detection with Weak Multi-Domain Supervision," vol. 2019-July, pp. 1–8, 7 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8851907/ (cited on page 4)

[19] D. A. Konovalov, A. Saleh, J. A. Domingos, R. D. White, and D. R. Jerry, "Estimating mass of harvested asian seabass lates calcarifer from images," *World Journal of Engineering and Technology*, vol. 6, no. 03, p. 15, 2018. [Online]. Available: http://dx.doi.org/10.4236/wjet.2018.63B0 (cited on pages 4, 43, 44, 76, 77, 78, 79, 80, 81, 83, 87, 88, and 89)

[20] D. A. Konovalov, A. Saleh, D. B. Efremova, J. A. Domingos, and D. R. Jerry, "Automatic Weight Estimation of Harvested Fish from Images," 12 2019. (cited on pages 4 and 44)

[21] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2017)*, 2017. (cited on pages 6 and 32)

[22] A. A. Shvets, A. Rakhlin, A. A. Kalinin, and V. I. Iglovikov, "Automatic instrument segmentation in robot-assisted surgery using deep learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 624–628. [Online]. Available: http://dx.doi.org/10.1109/ICMLA.2018.00100 (cited on pages 7, 77, 80, 81, 90, and 94)

[23] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A Survey of Optimization Methods

From a Machine Learning Perspective," *IEEE Transactions on Cybernetics*, pp. 1–14, 11 2019. (cited on page 12)

[24] A. Ipsen and T. M. Ebbels, "Prospects for a statistical theory of lc/tofms data," *Journal of the American Society for Mass Spectrometry*, vol. 23, no. 5, pp. 779–791, 2012. (cited on page 12)

[25] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, p. 85–117, Jan 2015. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.09.003 (cited on page 13)

[26] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition." in *Interspeech*, vol. 11, 2013, pp. 73–5. (cited on page 14)

[27] K. Korekado, T. Morie, O. Nomura, H. Ando, T. Nakano, M. Matsugu, and A. Iwata, "A convolutional neural network vlsi for image recognition using merged/mixed analog-digital architecture," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2003, pp. 169–176. (cited on page 14)

[28] A. Karpathy, "Connecting images and natural language," Ph.D. dissertation, Ph. D. thesis, Stanford University, 2016. (cited on pages 17 and 18)

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://www.researchgate.net/publication/269935079_Adam_A_Method_for_Stochastic_Optimization (cited on pages 18 and 80)

[30] F. Shafait, A. Mian, M. Shortis, B. Ghanem, P. F. Culverhouse, D. Edgington, D. Cline, M. Ravanbakhsh, J. Seager, and E. S. Harvey, "Fish identification from videos captured in uncontrolled underwater environments," *ICES Journal of Marine Science*, vol. 73, pp. 2737–2746, 2016. (cited on pages 22, 25, 27, and 95)

[31] S. A. Siddiqui, A. Salman, M. I. Malik, F. Shafait, A. Mian, M. R. Shortis, and E. S. Harvey, "Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data," *ICES Journal of Marine Science*, vol. 75, pp. 374–389, 2018. (cited on pages 22, 26, 27, 28, and 95)

[32] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015. (cited on pages 23 and 64)

[33] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with fast r-cnn," in *OCEANS 2015 - MTS/IEEE Washington*, 2015, pp. 1–5. (cited on pages 23, 25, and 27)

[34] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planque, A. Rauber, R. Fisher, and H. Müller, "Lifeclef 2014: Multimedia life species identification challenges," in *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, ser. Lecture Notes in Computer Science, E. Kanoulas, M. Lupu, P. Clough, M. Sanderson, M. Hall, A. Hanbury, and E. Toms, Eds., vol. 8685. Cham: Springer International Publishing, 2014, pp. 229–249. (cited on pages 24, 43, and 44)

[35] R. B. Fisher *et al.* [Online]. Available: www.fish4knowledge.eu (cited on page 24)

[36] B. J. Boom, J. He, S. Palazzo, P. X. Huang, C. Beyan, H.-M. Chou, F.-P. Lin, C. Spampinato, and R. B. Fisher, "A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage," *Ecological Informatics*, vol. 23, pp. 83 – 97, 2014, special Issue on Multimedia in Ecology and Environment. (cited on pages 24, 25, 43, and 44)

[37] C. Spampinato, S. Palazzo, P. H. Joalland, S. Paris, H. Glotin, K. Blanc, D. Lingrand, and F. Precioso, "Fine-grained object recognition in underwater visual data," *Multimedia Tools and Applications*, vol. 75, pp. 1701–1720, 2016. (cited on pages 24, 25, 26, and 27)

[38] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008. [Online]. Available: http://www.vlfeat.org (cited on page 24)

[39] ——, "VLfeat: An open and portable library of computer vision algorithms," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1469–1472. (cited on page 24)

[40] A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, and E. Harvey, "Fish species classification in unconstrained underwater environments based on deep learning," *Limnology and Oceanography: Methods*, vol. 14, pp. 570–585, 2016. (cited on pages 24, 26, 27, and 36)

[41] O. Barnich and M. V. Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011. (cited on page 24)

[42] K. Blanc, D. Lingrand, and F. Precioso, "Fish species recognition from video using svm classifier," in *Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data*, ser. MAED '14. New York, NY, USA: ACM, 2014, pp. 1–6. (cited on pages 24 and 27)

[43] R. Girshick, "Fast r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. (cited on page 24)

[44] Y. Hu, A. S. Mian, and R. Owens, "Face recognition using sparse approximated nearest points between image sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1992–2004, 2012. (cited on page 25)

[45] Y.-H. Hsiao, C.-C. Chen, S.-I. Lin, and F.-P. Lin, "Real-world underwater fish recognition and identification, using sparse representation," *Ecological Informatics*, vol. 23, pp. 13 – 21, 2014, special Issue on Multimedia in Ecology and Environment. (cited on page 25)

[46] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher, "Detecting, tracking and counting fish in low quality unconstrained underwater videos," in *Proceedings of 3rd International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 2, 2008, pp. 514–519. (cited on pages 25 and 26)

[47] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, pp. 773 – 780, 2006. (cited on pages 26 and 28)

[48] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2017. (cited on pages 26 and 32)

[49] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planqué, A. Rauber, S. Palazzo, B. Fisher, and H. Müller, "Lifeclef 2015: Multimedia life species identification challenges," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ser. Lecture Notes in Computer Science, J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. Jones, E. San Juan, L. Capellato, and N. Ferro, Eds., vol. 9283. Cham: Springer International Publishing, 2015, pp. 462–483. (cited on pages 26, 27, 34, and 37)

[50] "Fish species recognition." [Online]. Available: https://bit.ly/2LomRTp (cited on pages 26, 27, 34, and 37)

[51] E. S. Harvey, M. Cappo, G. A. Kendrick, and D. L. McLean, "Coastal fish assemblages reflect geological and oceanographic gradients within an australian zootone," *PLOS ONE*, vol. 8, 2013. (cited on page 27)

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105. (cited on page 27)

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. (cited on pages 27 and 65)

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. (cited on page 27)

[55] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. (cited on pages 27, 34, and 48)

[56] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 512–519. (cited on page 27)

[57] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 1717–1724. (cited on page 27)

[58] S. Marini, E. Fanelli, V. Sbragaglia, E. Azzurro, J. Del Rio Fernandez, and J. Aguzzi, "Tracking fish abundance by underwater image recognition," *Scientific Reports*, vol. 8, p. 13748, 2018. (cited on pages 27 and 28)

[59] "The western mediterranean expandable seafloor observatory (OBSEA)." [Online]. Available: http://www.obsea.es (cited on page 27)

[60] M. Bradley, R. Baker, I. Nagelkerken, and M. Sheaves, "Context is more important than habitat type in determining use by juvenile fish," *Landscape Ecology*, 2019, in press. (cited on pages 30, 42, and 45)

[61] K. Zuiderveld, *"Contrast Limited Adaptive Histogram Equalization"*. San Diego: "Academic Press Professional", 1994, pp. 474–485. (cited on pages 32 and 36)

[62] International Color Consortium, *Specification ICC.1:2004-10 (Profile version 4.2.0.0) Image technology colour management - Architecture, profile format, and data structure*, 2004. [Online]. Available: http://www.color.org/icc1v42.pdf (cited on page 32)

[63] O. Russakovsky, A. L. Bearman, V. Ferrari, and F. Li, "What's the point: Semantic segmentation with point supervision," *CoRR*, vol. abs/1506.02106, 2015. (cited on pages 32 and 35)

[64] F. Chollet *et al.*, "Keras: The python deep learning library," 2015. [Online]. Available: https://keras.io/ (cited on pages 32, 36, and 65)

[65] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: http://tensorflow.org/ (cited on pages 34, 36, and 65)

[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. (cited on pages 34 and 67)

[67] K. Anantharajah, Z. Ge, C. McCool, S. Denman, C. Fookes, P. Corke, D. Tjondronegoro, and S. Sridharan, "Local inter-session variability modelling for object classification," in *IEEE Winter Conference on Applications of Computer Vision*, 2014, pp. 309–316. (cited on pages 34 and 37)

[68] "QUT fish dataset." [Online]. Available: https://bit.ly/2APDvGB (cited on pages 34 and 37)

[69] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, 2015. (cited on pages 34 and 35)

[70] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 560–576. (cited on page 35)

[71] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2005.10.010 (cited on page 37)

[72] M. A. Hussain, T. Saputra, E. A. Szabo, and B. Nelan, "An overview of seafood supply, food safety and regulation in new south wales, australia," in *Foods*, 2017. (cited on page 42)

[73] M. Skirtun, M. Stephan, and K. Mazur, "Australian fisheries economic indicators report," *null*, 2013. (cited on page 42)

[74] M. Sheaves, R. Johnston, and R. Baker, "Use of mangroves by fish: new insights from in-forest videos," *Marine Ecology Progress Series*, vol. 549, pp. 167–182, 2016. (cited on page 42)

[75] L. M. Barnes, D. R. Bellwood, M. Sheaves, and J. K. Tanner, "The use of clear-water non-estuarine mangroves by reef fishes on the great barrier reef," *Marine Biology*, vol. 159, pp. 211–220, 2012. (cited on page 42)

[76] M. Sheaves, R. Johnston, R. M. Connolly, and R. Baker, "Importance of estuarine mangroves to juvenile banana prawns," in *Estuarine, Coastal and Shelf Science*, 2012. (cited on page 42)

[77] L. P. Rozas and T. J. Minello, "Estimating densities of small fishes and decapod crustaceans in shallow estuarine habitats: A review of sampling design with focus on gear selection," *Estuaries*, vol. 20, pp. 199–213, 1997. (cited on page 42)

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (cited on pages 42 and 48)

[79] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. (cited on page 42)

[80] G. French, M. Fisher, M. Mackiewicz, and C. Needle, "Convolutional neural networks for counting fish in fisheries surveillance video," in

Machine Vision of Animals and their Behaviour (MVAB), 2015. (cited on pages 42 and 44)

[81] A. Salman, S. A. Siddiqui, F. Shafait, A. Mian, M. R. Shortis, K. Khurshid, A. Ulges, and U. Schwanecke, "Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system," *ICES Journal of Marine Science*, 2019. (cited on page 43)

[82] W. Xu and S. Matzner, "Underwater Fish Detection Using Deep Learning for Water Power Applications," pp. 313–318, 12 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8947884/ (cited on page 43)

[83] D. Rathi, S. Jain, and S. Indu, "Underwater fish species classification using convolutional neural network and deep learning," *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, pp. 1–6, 2017. (cited on page 43)

[84] D. A. Konovalov, A. Saleh, M. Bradley, M. Sankupellay, S. Marini, and M. Sheaves, "Underwater fish detection with weak multi-domain supervision," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8. (cited on pages 43 and 45)

[85] M.-C. Chuang, J.-N. Hwang, and C. S. Rose, "Aggregated segmentation of fish from conveyor belt videos," *International Conference on Acoustics, Speech and Signal Processing*, pp. 1807–1811, 2013. (cited on page 43)

[86] R. Girshick, "Fast r-cnn," in *international conference on computer vision (ICCV)*, 2015. (cited on page 44)

[87] K. He, G. Gkioxari, P. Dollar, and R. B. Girshick, "Mask r-cnn." *IEEE transactions on pattern analysis and machine intelligence*, 2018. (cited on page 44)

[88] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision (ECCV)*, 2014. (cited on page 44)

[89] R. Garcia, R. Prados, J. Quintana, A. Tempelaar, N. Gracias, S. Rosen, H. Vagstol, and K. Lovall, "Automatic segmentation of fish using deep learning with application to fish size measurement," *ICES Journal of Marine Science*, 2019. (cited on page 44)

[90] M. Bradley, R. Baker, and M. Sheaves, "Hidden components in tropical seascapes: deep-estuary habitats support unique fish assemblages," *Estuaries and Coasts*, vol. 40, no. 4, pp. 1195–1206, 2017. (cited on page 45)

[91] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, pp. 157–173, 2005. (cited on page 46)

[92] A. Klaser, "Image annotation tool with image masks," https://lear.inrialpes.fr/people/klaeser/software_image_annotation. (cited on page 47)

[93] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010. (cited on page 48)

[94] K. P. Murphy, "Machine learning - a probabilistic perspective," in *Adaptive computation and machine learning series*, 2012. (cited on pages 49 and 50)

[95] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. (cited on page 49)

[96] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 560–576. (cited on page 50)

[97] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," vol. 39, no. 4. IEEE Computer Society, 4 2017, pp. 640–651. (cited on pages 50, 77, and 79)

[98] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988. (cited on page 52)

[99] J. S. Huxley, "Constant differential growth-ratios and their significance," *Nature*, vol. 114, no. 2877, p. 895, 1924. (cited on pages 60 and 75)

[100] B. Zion, "The use of computer vision technologies in aquaculture – a review," *Computers and Electronics in Agriculture*, vol. 88, pp. 125–132, 2012. (cited on page 60)

[101] B. Zion, A. Shklyar, and I. Karplus, "Sorting fish by computer vision," *Aquacultural Engineering*, vol. 22, no. 3, pp. 165–179, 2000. (cited on page 61)

[102] S. Viazzi, S. Van Hoestenberghe, B. M. Goddeeris, and D. Berckmans, "Automatic mass estimation of jade perch scortum barcoo by computer vision," *Aquacultural Engineering*, vol. 64, pp. 42–48, 2015. (cited on pages 61, 63, and 64)

[103] M. O. Balaban, M. Chombeau, D. Cırban, and B. Gümüş, "Prediction of the weight of alaskan pollock using image analysis," *Journal of food science*, vol. 75, no. 8, pp. E552–E556, 2010. [Online]. Available: http://dx.doi.org/10.1111/j.1750-3841.2010.01813.x (cited on pages 61, 64, 75, 77, and 84)

[104] C. Frederick, D. C. Brady, and I. Bricknell, "Landing strips: Model development for estimating body surface area of farmed atlantic salmon (salmo salar)," *Aquaculture*, vol. 473, pp. 299 – 302, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0044848616306810 (cited on pages 61 and 64)

[105] A. JAWORSKI and J. HOLM, "Distribution and structure of the population of sea lice, lepeophtheirus salmonis krøyer, on atlantic salmon, salmo salar l., under typical rearing conditions,"

Aquaculture Research, vol. 23, no. 5, pp. 577–589, 1992. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2109.1992.tb00802.x (cited on pages 61 and 64)

[106] K. Zenger, M. Khatkar, D. Jerry, and H. Raadsma, "The next wave in selective breeding: implementing genomic selection in aquaculture," in *Proc. Assoc. Advmt. Anim. Breed. Genet*, vol. 22, 2017, pp. 105–112. [Online]. Available: https://bit.ly/2I5gV00 (cited on page 61)

[107] J. A. Domingos, C. Smith-Keune, and D. R. Jerry, "Fate of genetic diversity within and between generations and implications for dna parentage analysis in selective breeding of mass spawners: a case study of commercially farmed barramundi, lates calcarifer," *Aquaculture*, vol. 424, pp. 174–182, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.aquaculture.2014.01.004 (cited on pages 61, 63, 76, and 78)

[108] D. J. Konovalov, D.A. and D. Jerry, "Barra-ruler-445 %(br445) dataset," 2017. [Online]. Available: https://github.com/dmitryako/BarraRulerDataset445 (cited on page 63)

[109] ——, "Barra-area-600 %(ba600) dataset," 2018. [Online]. Available: https://github.com/dmitryako/BarraAreaDataset600 (cited on page 63)

[110] D. Konovalov, J. Domingos, R. White, and D. Jerry, "Automatic scaling of fish images," in *Proceedings of the 2nd International Conference on Advances in Image Processing*. ACM, 2018, pp. 48–53. [Online]. Available: http://dx.doi.org/10.1145/3239576.3239595 (cited on pages 63, 64, 69, 71, and 78)

[111] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, p. 640–651, Apr 2017. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2016.2572683 (cited on pages 64, 65, 66, 77, and 79)

[112] D. Konovalov, S. Hillcoat, G. Williams, A. Birtles, N. Gardi-Ner, and M. I Curnock, "Individual minke whale recognition using deep learning convolutional neural networks," 02 2018. (cited on page 64)

[113] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015. (cited on page 65)

[114] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.222 (cited on pages 65, 66, and 80)

[115] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 249–256. (cited on page 66)

[116] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945. [Online]. Available: http://dx.doi.org/10.2307/1932409 (cited on pages 67 and 81)

[117] D. A. Konovalov, L. E. Llewellyn, Y. Vander Heyden, and D. Coomans, "Robust cross-validation of linear regression qsar models," *Journal of Chemical Information and Modeling*, vol. 48, no. 10, pp. 2081–2094, 2008. (cited on page 70)

[118] D.-W. Sun, *Hyperspectral imaging for food quality analysis and control*. Elsevier, 2010. (cited on page 74)

[119] C. Costa, F. Antonucci, F. Pallottino, J. Aguzzi, D.-W. Sun, and P. Menesatti, "Shape analysis of agricultural products: a review of recent re-

search advances and potential application to computer vision," *Food and Bioprocess Technology*, vol. 4, no. 5, pp. 673–692, 2011. (cited on page 74)

[120] D.-W. Sun, *Computer vision technology for food quality evaluation*. Academic Press, 2016. (cited on page 74)

[121] J. R. Mathiassen, E. Misimi, M. Bondø, E. Veliyulin, and S. O. Østvik, "Trends in application of imaging technologies to inspection of fish and fish products," *Trends in Food Science & Technology*, vol. 22, no. 6, pp. 257–275, 2011. (cited on page 74)

[122] M. A. Pavlidis and C. C. Mylonas, *Sparidae: Biology and aquaculture of gilthead sea bream and other species*. John Wiley & Sons, 2011. (cited on page 74)

[123] R. L. Shewfelt and B. Bruckner, *Fruit and vegetable quality: an integrated view*. CRC Press, 2000. (cited on page 74)

[124] C. Alasalvar, K. Miyashita, F. Shahidi, and U. Wanasundara, *Handbook of seafood quality, safety and health applications*. John Wiley & Sons, 2011. (cited on page 74)

[125] C. A. Martınez-Palacios, E. B. Tovar, J. F. Taylor, G. R. Durán, and L. G. Ross, "Effect of temperature on growth and survival of chirostoma estor estor, jordan 1879, monitored using a simple video technique for remote measurement of length and mass of larval and juvenile fishes," *Aquaculture*, vol. 209, no. 1-4, pp. 369–377, 2002. (cited on page 74)

[126] M. O. Balaban, G. F. Ünal Şengör, M. G. Soriano, and E. G. Ruiz, "Using image analysis to predict the weight of alaskan salmon of different species," *Journal of food science*, vol. 75, no. 3, pp. E157–E162, 2010. (cited on pages 74 and 75)

[127] B. Gümüş and M. O. Balaban, "Prediction of the weight of aquacultured rainbow trout (oncorhynchus mykiss) by image analysis," *Journal of Aquatic Food Product Technology*, vol. 19, no. 3-4, pp. 227–237, 2010. (cited on page 75)

[128] J. R. Mathiassen, E. Misimi, B. Toldnes, M. Bondø, and S. O. Østvik, "High-speed weight estimation of whole herring (clupea harengus) using 3d machine vision," *Journal of food science*, vol. 76, no. 6, pp. E458–E464, 2011. (cited on page 75)

[129] H. Spencer, *The principles of biology*. D. Appleton, 1896, vol. 1. (cited on page 75)

[130] E. Le Cren, "The length-weight relationship and seasonal cycle in gonad weight and condition in the perch (perca fluviatilis)," *The Journal of Animal Ecology*, pp. 201–219, 1951. (cited on page 75)

[131] T. A. Beddow, L. G. Ross, and J. A. Marchant, "Predicting salmon biomass remotely using a digital stereo-imaging technique," *Aquaculture*, vol. 146, no. 3-4, pp. 189–203, 1996. (cited on page 75)

[132] B. Zion, "The use of computer vision technologies in aquaculture–a review," *Computers and electronics in agriculture*, vol. 88, pp. 125–132, 2012. (cited on page 75)

[133] M. Poxton and G. Goldsworthy, "The remote estimation of weight and growth in turbot using image analysis," *IFAC Proceedings Volumes*, vol. 20, no. 7, pp. 163–170, 1987. (cited on page 75)

[134] N. Strachan, "Length measurement of fish by computer vision," *Computers and electronics in agriculture*, vol. 8, no. 2, pp. 93–104, 1993. (cited on pages 75 and 77)

[135] B. Zion, A. Shklyar, and I. Karplus, "Sorting fish by computer vision," *Computers and electronics in agriculture*, vol. 23, no. 3, pp. 175–187, 1999. (cited on page 75)

[136] F. Odone, E. Trucco, and A. Verri, "A trainable system for grading fish from images," *Applied Artificial Intelligence*, vol. 15, no. 8, pp. 735–745, 2001. (cited on page 75)

[137] S. Viazzi, S. Van Hoestenberghe, B. Goddeeris, and D. Berckmans, "Automatic mass estimation of jade perch scortum barcoo by computer

vision," *Aquacultural engineering*, vol. 64, pp. 42–48, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.aquaeng.2014.11.003 (cited on pages 76, 77, 84, 90, and 94)

[138] G. Sanchez-Torres, A. Ceballos-Arroyo, and S. Robles-Serrano, "Automatic measurement of fish weight and size by processing underwater hatchery images," *Engineering Letters*, vol. 26, no. 4, 2018. [Online]. Available: https://bit.ly/2Z8jw0h (cited on pages 76, 90, and 94)

[139] J. S. Huxley, "Constant differential growth-ratios and their significance," *Nature*, vol. 114, pp. 895–896, 1924. (cited on page 76)

[140] B. Zion, "The use of computer vision technologies in aquaculture – a review," *Computers and Electronics in Agriculture*, vol. 88, pp. 125 – 132, 2012. (cited on page 76)

[141] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24574-4_28 (cited on pages 77, 79, and 80)

[142] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4. [Online]. Available: http://dx.doi.org/10.1109/VCIP.2017.8305148 (cited on pages 77 and 80)

[143] D. Konovalov, J. Domingos, C. Bajema, R. White, and D. Jerry, "Ruler detection for automatic scaling of fish images," in *Proceedings of the International Conference on Advances in Image Processing*. ACM, 2017, pp. 90–95. [Online]. Available: http://dx.doi.org/10.1145/3133264.3133271 (cited on page 78)

[144] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer*

vision and pattern recognition, 2016, pp. 770–778. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.90 (cited on page 80)

[145] D. A. Konovalov, S. Jahangard, and L. Schwarzkopf, "In situ cane toad recognition," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2018, pp. 1–7. [Online]. Available: http://dx.doi.org/10.1109/DICTA.2018.8615780 (cited on page 81)

[146] D. A. Konovalov, L. E. Llewellyn, Y. Vander Heyden, and D. Coomans, "Robust cross-validation of linear regression QSAR models," *Journal of Chemical Information and Modeling*, vol. 48, pp. 2081–2094, 2008. (cited on pages 84 and 87)

[147] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981. (cited on page 84)

[148] T. Standley, O. Sener, D. Chen, and S. Savarese, "image2mass: Estimating the mass of an object from its image," in *Conference on Robot Learning*, 2017, pp. 324–333. (cited on pages 86, 87, and 90)

[149] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010. (cited on page 93)