Machine Learning for Tarakihi Fish Length Estimation in Aotearoa

by

Michael Stanley

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Master of Science
in Artificial Intelligence.

Victoria University of Wellington
2023

Abstract

Current practices for monitoring the catch of deep sea fishing vessels is labour intensive requiring a person on vessel measuring individual fish lengths manually. Capturing videos of fish on-vessel instead allows the use of machine learning algorithms for tackling a computer vision based problem to automate the collection of morphological data of the observed fish.

In this thesis we investigate essential methods required and develop a system that uses machine learning algorithms and computer vision techniques to calculate centimetre accurate lengths of singulated fish from video footage. The first stage in this process is the data acquisition, where we explore the use of both a fixed camera and a free camera (one that is held in hand) for gathering the video data from which we extract millimeter lengths. Lengths were gathered on-site to compare the lengths found from images at different orientations and translations. An analysis of the different camera positions and rotations found that a camera positioned above the object of interest and calibration pattern was able to achieve the most accurate lengths. Rotation was found to have an increasingly detrimental effect on predicted lengths as rotation, measured in radians, increased.

Secondly, we use binary masks that are created both manually, and by using an automated approach, based on edge detection, for training a segmentation model to identify fish in images. We leverage shape features and interpretable ML classifier to analyse the features of contours from both inferred masks and those derived from an edge detection. In our analysis of these shape features, we identify a range of values for the

feature ”circle deviation” which may be used to identify potential fish contours that did not pass the classification, and flag such contours for further training. We use contours, derived from the edge detection approach, that do pass the classification for creating a dataset of cropped images, to train a GAN from which a synthetic imagery set is created.

Thirdly, we develop a method from extracting the lengths of fish from images by using a checkerboard pattern as a point of reference, to relate pixel lengths to millimeters. Only inferred contours with shape features within the range identified by our analysis are used in calculating lengths. This approach reduced the number of partially visible fish or false positive inferences from affecting our recorded lengths.

Finally, the performance of models trained on real images, synthetic images, and a combination of the two are compared. A model that was trained on both real and synthetic images achieved an average for the absolute differences between true and predicted lengths of below one centimetre over 128 samples. Our results suggest that the use of synthetic data to assist in the creation of a robust training dataset is viable. However, this synthetic data works best when there is also real data available in the training set.

Acknowledgments

I would first like to thank both my supervisors Andrew Lensen and Mengjie Zhang for their guidance, support and thorough critique throughout the course of this thesis. A huge thank you goes to both my supervisors and Lynker Analytics for helping me arrange the financial support to make this research possible, and to both Lynker Analytics and Callaghan Innovation for providing the support.

I would like to extend this thanks to the tutors who provided me with the knowledge on computer vision and machine learning to carry out this research. As well as, those who went above and beyond to help me get on this path, namely Peter Andreae and Karsten Lundqvist.

A big thank you also goes to all of the team from Lynker Analytics. Especially, Matt Lythe, Gordon Morris and Sam Brown for all their advice and hard work in making this research possible. A special mention goes out to David Knox, for his professional guidance, mentorship and patience during the many whiteboard sessions over the years.

To my family, I thank you for the example you set, the support and opportunities your hard work has provided me, and the motivation that has given me the strength to come this far. To my siblings, Alan, Mary-Anne and Leam, I thank you for your humour and emotional support.

This work was financially supported by Callaghan innovation and Lynker Analytics and was conducted as a collaboration between Lynker Analytics and Fisheries Inshore New Zealand.

Contents

Chapter 1

Introduction

The environmental impacts of overfishing have been clear for decades [27]. With 70% of assessed fish populations being fully used, overused or in crisis [51], governments have been pushing for more oversight and regulation of the fishing industry[27]. Despite this, a study in 2016 [28] found that reconstructed catches between 1950 and 2010 were 53% higher than the reported data.

In New Zealand, monitoring of commercial fisheries consists of reporting on the numbers and kinds of fish caught. Only in important fisheries is the size, age and the quantity of fish that are being caught also measured [4]. Just over a quarter of fish caught in deepwater fishing activities is caught by vessels which have a fisheries observer present on the commercial fishing boat [16], while the rest is left to commercial fishers. The Ministry for Primary Industries in New Zealand is beginning to develop digital monitoring systems to be able to verify what is being reported [17]. Artificial intelligence (AI) provides the opportunity for the development of automated systems to quantify fishing activities, allowing for better monitoring and catch estimates that are less susceptible to manipulation.

The field of artificial intelligence is at the precipice of a significant change in focus, and the shift from model-centric approaches to a data-centric approach more accurately represents real-world issues when implementing data science solutions [59].

Much of the recent innovation in data science has focused on improving or re-imagining model architectures. Over the past few years models have been created that are eight times smaller than their predecessors [80] while also improving their performance. However, data remains the most significant factor when developing a machine learning (ML) solution and is the greatest contributor to the performance of supervised models[33]. With the digitalization of monitoring methods on fishing vessels, the data required to implement AI fish surveying systems are now present.

1.1 Motivation

New Zealand's marine fisheries waters measure 4,200,000 km^2 making it the fourth-largest exclusive economic zone[1] and was ranked fifth in 2016 out of 28 nations, for healthy fish stocks. Despite this, mandatory measures were called for in 2020 to stop the overfishing of tarakihi, which have been fished down to 15% of their original levels[57]. By leveraging recent advances in AI, more specifically deep learning, we hope to help mitigate some of the damage caused by fisheries, by providing a method to accurately measure the length of caught fish.

Science-based management of commercial fisheries suffers from a lack of availability of data [15]. For informed decisions to be made, governments need greater insight into the state of sea life. Neural Networks applied in computer vision offer the opportunity to gain information on current practices without introducing biases or information loss at the catch stage [46].

Advances in neural networks and the increase in speed of the GPUs required to run them over the last decade have allowed the practical implementation of this technology in a wide range of industries. However, the adoption of smart monitoring systems in the fishing industry in New Zealand is yet to be implemented. Availability of data surrounding the fishing industry and their onboard activities is not widely available; visual data of the deck is often locked behind privacy concerns where employees of the vessels may be identified, and all imagery is reliant on having the approval of either the captain or the organisation to be distributed. As such, the analysis of the potential of these techniques in the industry has been hampered.

Non-invasive approaches to monitoring are crucial when applying automated systems on-vessel; the privacy of those working on-vessel is highly important. In New Zealand, data collection for monitoring the types and quantity of fish caught is mainly reliant on human reviewers [16]. With over 4 million square kilometres of ocean, efforts to monitor the multi-billion dollar New Zealand fishing industry are difficult and expensive, resulting in only around 25% of deepwater catch occurring on a vessel with an observer [16].

Deep learning generally requires large datasets to automate the process of feature engineering [73], this data must be gathered, cleaned and in the case of supervised learning, labelled. According to a survey in 2020 by Anaconda [13], getting data ready for training (data loading and data cleansing), on average, takes up 45% of data scientists' time, while model selection takes 11% and model training and scoring take 12%. The bottleneck presented from data preparation is clear.

A data-driven approach to machine learning presents the opportunity to achieve reliable results without the burden of having to gather hundreds of thousands of images. Recent innovations, such as the generation of synthetic images using adaptive discriminator augmentation [42] enable the creation of high-quality synthetic imagery with far fewer input data than previously required and its implementation into data-driven machine learning has been advocated for by industry leaders [78].

The generation of robust datasets that are clean enough to be used to tackle machine learning problems is often the stage that presents the greatest difficulty. Development of data science solutions on problems for which no cleaned and publicly available dataset exists may face many issues, such as time or cost constraints related to gathering the data, as well as privacy concerns when the data pertains to the general public. The development of synthetic data generation aims to overcome these issues [86]. With synthetic data we hope to develop a reliable model that may be used to predict the location of nemadactylus macropterus (tarakihi), inferring a segmentation from which length may be estimated.

1.2 Goals

The overall goal of this thesis is to train an ML algorithm to reliably detect tarakihi fish and to estimate their length from this detection. We will also explore the use of synthetic data to help create a varied dataset large enough to train a NN, Figure 1.1 illustrates the core flow of data, from raw images to a length prediction per fish. Specifically, this research will focus on the following objectives

1. Gathering a diverse dataset suitable to achieve accurate length estimation. This dataset will be used to train both a network for semantic segmentation and another for the creation of synthetic images.

   Data collection will be performed in iterations to allow for lessons learnt to be applied to subsequent data collection methods.

   (a) Data collection will first be performed with a handheld camera to allow for a large variety of observation angles and distances to be captured. A checkerboard calibration pattern will be visible in the imagery and provide a point of reference from which lengths may be inferred.

   (b) The second data collection approach will use a fixed camera with a bird-eye perspective looking down at fish placed on a table below. Fish will be measured on-site providing us with a ground truth set of lengths to which predicted lengths may be compared.

(c) The factors that may impact the accuracy of length measurements, such as the orientation and distance of the camera will be explored and discussed.

2. Investigating the use of generative adversarial networks to create synthetic data. This is expected to provide us with a larger and more varied dataset, improving the final model's performance.

   (a) A generative adversarial network will be trained on the imagery gathered as part of this research and used to synthesize additional images.

   (b) The effect of synthetic imagery when included in the training data for a segmentation model will be evaluated by comparing the performance of three different models, one trained on real images, another trained on synthetic images and a third model trained on a combination of the two.

3. Exploring the use of bounding boxes, as well as object segmentation, to estimate the pixel length of fish. Pixel to length ratios found from the calibration pattern, included when gathering data, will be used to relate pixel lengths to millimeter lengths.

   (a) Measuring fish length from bounding boxes may prove difficult particularly if the fish is not parallel to the bottom of the image. Measuring length as the distance from each corner of the bounding box may be explored as well as using image processing tools to rotate the image until the fish length can be attained by measuring the pixel distance across a single axis.

   (b) Object segmentation offers a method for deriving pixel lengths directly from the inferred segmentation of a fish. This approach would not be impaired by the rotation of the fish as the inference is per-pixel allowing the maximum length of the segmented region to be used as the total pixel length for the fish.

1.3 Major Contributions

This thesis makes the following major contributions.

1. This thesis has provided a method for extracting centimetre accurate length estimations from the contours of a binary mask created from the inference of a semantic segmentation model. The impact of camera location and orientation on lengths estimated from a static object and reference pattern is analysed providing justification for the optimal camera position.

2. This Thesis presents a method for evaluating the quality of contours derived from segmentation masks by using shape features in conjunction with interpretable machine learning methods to generate shape functions that explain the contribution of features towards determining a high or low-quality contour.

3. This thesis explores the use of synthetic imagery for assisting in data creation where little is available. Synthetic data is manually annotated and used for training segmentation models. Our results suggest that synthetic data may be used in addition to real imagery to improve the training data for segmentation models. Other similar research particularly in medical imaging where privacy concerns are an issue has also explored the use of synthetic imagery for training deep neural networks and found similar results.

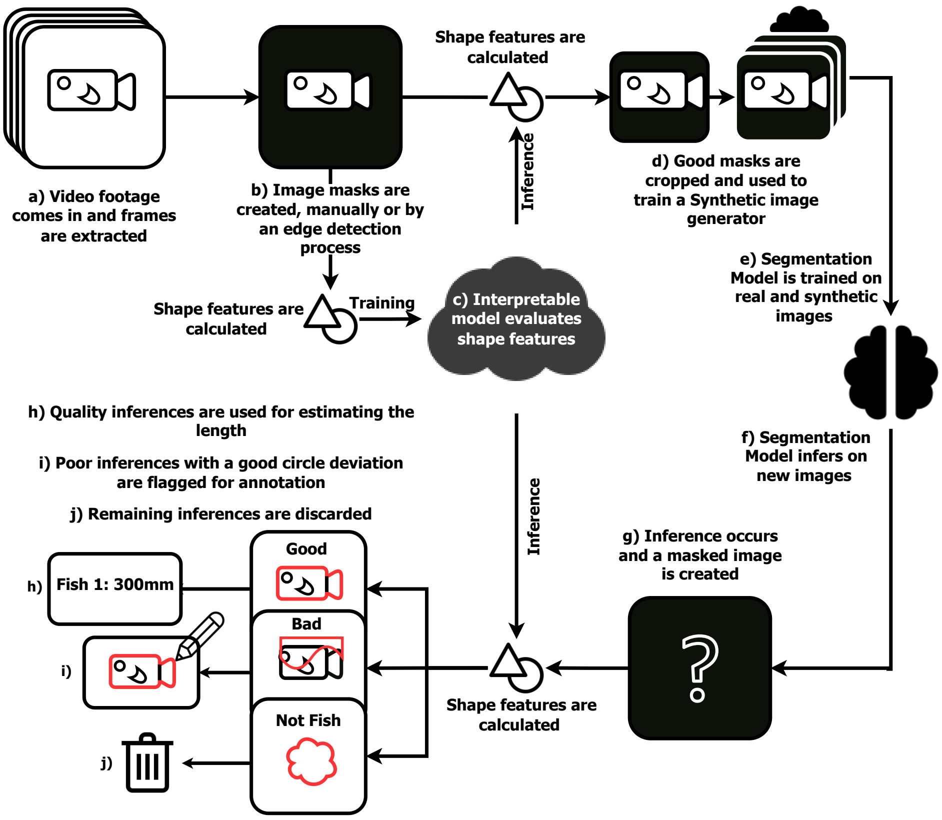

Shape features are calculated
a) Video footage comes in and frames are extracted
b) Image masks are created, manually or by an edge detection process
Inference
d) Good masks are cropped and used to train a Synthetic image generator
e) Segmentation Model is trained on real and synthetic images
Shape features are calculated
Training
c) Interpretable model evaluates shape features
h) Quality inferences are used for estimating the length
i) Poor inferences with a good circle deviation are flagged for annotation
j) Remaining inferences are discarded
f) Segmentation Model infers on new images
Inference
g) Inference occurs and a masked image is created
Good
h)
Fish 1: 300mm
Bad
i)
Not Fish
j)
Shape features are calculated
?

Figure 1.1: Flowchart illustrating the entire process surrounding fish segmentation and the use of contours. Image frames are extracted from a video (a) and masks are created where fish are visible (b). The contours of fish masks are used to train an interpretable model, which then determines the quality of future masks (c). Quality masks are used to crop the original image, and the cropped images are used to train a GAN to synthesize new images (d). Real and synthetic images are then used in training a segmentation model, new mask images are made from the inference of this model on unseen fish images (g). Finally, the contours of these inferred segmentations are evaluated by the interpretable model. Good contours are used to estimate length (h), and contours with a circle deviation in a certain range are flagged as potential fish and used for further training if a fish is present (i). Poor inferences are discarded if no fish is present (j).

1.4 Outline

The remainder of this thesis is structured as follows:

Chapter 2 discusses the background for this thesis. First, detailing the current practices in New Zealand fisheries and introducing the concepts relevant to this research. We then survey literature with a similar scope to this thesis.

Chapter 3 describes the data collection process for each of the three factory visits that were conducted to gather the data for this work. We also provide a summary of the datasets created from this raw data and the lessons learnt from each visit.

Chapter 4 presents the augmentation that was used on image data and investigates the importance of camera positioning across the different datasets. We introduce interpretable ML techniques for analysing camera location and orientation in this investigation.

Chapter 5 describes our two approaches for creating segmentation masks and our method for generating synthetic images. We also present a method for evaluating the shapes of contours from inference masks to classify the quality of the inference.

Chapter 6 provides a detailed description of our approach for estimating millimeter lengths from the contours of inferred fish. We then investigate the impact of synthetic data when training a segmentation model on both the quality of inferences and the accuracy of predicted lengths. Our approach for adjusting predicted lengths to correct for unknown error is also described.

Chapter 7 Summarises the conclusions made for each of the major areas of this work and discusses the topics we believe future work would benefit from.

Chapter 2

Background

2.1 NZ Fishing Industry

Long-term profitability of the fishing industry requires sustainable practices, fish stocks must be exploited to maximise the profitability of the industry, but not over-exploited to the point where stocks collapse and future supply is reduced. Trawling was recently found to have killed around 80% more sea life than previously reported [84] (2018), at a total of 25 million tonnes up from the previously estimated 14 million, the difference between the two estimates was almost equally contributed to by unreported landings and discards.

Declining fish stocks and damage caused to sea habitats from large-scale fishing practices has led to the development of strict regulation as to where fishing may be conducted, the quantity of fish that may be caught and the size of fish that may be kept. Size limits were imposed to help protect fish stocks, by allowing fish to grow to a certain size they are given a chance to breed at least once [5].

The dangers brought forth by unsustainable fishing practices not only present an immediate issue by damaging seafloor habitats but also bring us ever closer to a dwindling supply of food [64]. Current methods are far from optimal, focusing on the size of the catch irrespective of the percentage of bycatch or damage to aquatic habitats [29]. Greater oversight is needed to ensure that more efficient fishing practices may be implemented.

2.2 Computer Vision

Computer vision involves techniques for the analysis and interpretation of visual data from digital images, typically using machine learning algorithms to extract useful information. The key computer vision techniques explored in this research are canny edge detection, object detection and semantic segmentation.

Edge detection is the use of mathematical methods for extracting edges from an image by detecting sharp changes in the brightness of an image. A convolutional kernel is a common method for calculating the image gradient to identify the locations in the image where a sharp change in brightness exists.

Semantic segmentation provides a pixel-level classification for the objects present in an image. This information is often stored in the form of an image mask of equal width and height to the original image, but where pixel values instead represent the class of the object at those coordinates in the original image.

2.3 Machine Learning

Machine Learning is a branch of Artificial Intelligence (AI). AI is the imitation of human intelligence through computer science and mathematical algorithms. The core goals of artificial intelligence include learning, reasoning and problem solving at or above a human level, concepts first covered in [83] Alan Turing's 'Computing Machinery and Intelligence' (1950).

Machine Learning algorithms focus on the automation of learning with the goal of the trained algorithm being able to achieve some task or prediction. During Supervised and unsupervised training, data is provided to the algorithm in order for it to identify trends and develop a foundation for its future predictions, though data has to be provided in this case, the process of identifying patterns in the data is automated. Transfer learning allows an algorithm to use the information that has already been learned in recognizing patterns for one problem to be then applied to a different but similar challenge. An example of this may be a model that is trained to detect dogs, then being trained to detect cats. The use of transfer learning means that this model does not need to be trained from scratch for the second challenge.

Supervised learning is performed when training data has a target class or label. The supervised model learns to recognize patterns in the data which distinguish the target classes, it may then apply the learned patterns to new data to predict the target class.

Unsupervised training is performed when the data does not have a target class. Clustering is a popular branch of unsupervised training, instead of learning trends in the data that help in distinguishing a targeted class, the model learns to cluster the data based on similarities, with probabilistic machine learning each data point may be assigned to multiple clusters, with a per cluster probability, we assume that each data point belongs to the cluster with the highest probability.

Other methods of machine learning, such as reinforcement learning, do not require data to be provided but instead may reward the machine for performing correct actions. The field of AI in robotics and simulation contains many examples of this [45].

OpenAI's [22] use of reinforcement learning for multi-agent hide and seek is a prime example of how reinforcement learning can lead to intelligent solutions to complex problems. In this research there were two teams of agents, the first group (hiders) was rewarded +1 if all team members were hidden and penalized otherwise. The second team (seekers) were penalized -1 if all hiders were hidden and rewarded +1 otherwise.

Through these seemingly simple goals, the hiders learnt to manipulate objects and construct shelters while seekers learnt to use ramps to traverse shelter walls.

2.4 Neural Networks

Neural Networks, inspired by neuron connections in the brain, are one of the most powerful tools for supervised learning [77]. Neural networks consist of an input layer, an output layer and a layer in-between (hidden layer). Deep neural networks consist of a greater number of layers, with each hidden layer typically having multiple nodes, and are more suited for complex tasks due to a larger number of learnable parameters.

These nodes are used to identify patterns in the data, with the nodes in the final layer representing some target or class. Nodes in each layer have an activation function, determining their output, and a connection to the nodes in the layers before and after to pass a signal from the input layer to the output. Weighted connections are adjusted during the training phase so that the network may learn which patterns are associated with the correct output.

Convolutional Neural Networks (CNNs) are commonly used to tackle computer vision problems, they include one or more convolutional layers, consisting of convolutional filters (kernel), this kernel has a size of $n \times n$.

In the convolutional layer, each value of the kernel matrix is multiplied by the corresponding value in the input image. The total sum of each multiplication is then the result for a single value in a new output image, at corresponding pixel coordinates located at the centre of the kernel's applied area. This process is iterated over the entire image, with a configured step size, to decide how many rows or columns the kernel is moved each time it is applied. Padding is used to maintain the size of the original image and to allow edge pixels to be processed [58].

By using kernels spatial information from neighbouring pixels is captured. Changing the values of the kernel allows for various features of the image to be captured, such as edges, colour gradient or texture, CNNs learn to use these features in predicting the class of the input image. Pooling layers are commonly used in CNNs, this allows for a reduction in the spatial size of the image, reducing the computational complexity.

2.5 EfficientNet

Modelling for fish length estimation will comprise of training two EfficientNet-based models on the images of fish. The first, named EfficientDet [81] is an object detector. The second is a segmentation model with an EfficientNet backbone [39]. Both are based on Google-autoML's EfficientNet [79] which used neural architecture search to create a baseline network that achieved state-of-the-art performance on 5 commonly used transfer learning datasets and often used far fewer parameters. This more efficient model allows less demanding hardware to be used to implement the results of this work.

Research into developing EfficientNet looked into re-thinking the way that neural networks are scaled. Previously width, depth and image size had been arbitrarily up-scaled, a time-consuming approach that, according to the research by Mingxing Tan & Quoc V.Le [81], was tedious and yielded sub-optimal results. The new approach presented in the EfficientNet research paper instead looks at finding a principled method for upscaling convolutional networks that balances all the dimensions of the network, by scaling each with a constant ratio.

This ratio consists of constant coefficients, α, β, γ, found from a small grid search of the small base model. To increase computational resources by 2^N, depth is increased by α^N, width by β^N and image size by γ^N.

EfficientDet is based on an EfficientNet backbone and uses the same principles in its approach to scaling. A bi-directional feature pyramid network

(BiFPN) is used for multi-scale feature fusion. This approach was previously used by Shu Liu et al in their work on path aggregation [47]. EfficientDet research optimises this approach by removing nodes with one input edge, the bi-directional node block is repeated, allowing for further high-level feature fusion. The features found from the EfficientNet backbone are passed to this BiFPN and through to a class and box network for the final prediction.

2.6 Generative Adversarial Networks

Generative adversarial networks (GANs) use deep learning techniques to generate synthetic data. This is done by using two neural networks competing against one another, one network to generate the synthetic data (generator) and another to try and distinguish this from real data (discriminator). This is an iterative process that is concluded once the discriminator can no longer separate the synthetic data from the real, this stage is known as the Nash Equilibrium[38].

Generators generally begin with an input consisting of random noise and attempt to replicate real data from this, if the generator fails to convince the discriminator that the generated data is real then the loss is calculated from this outcome, backpropagation occurs through both networks but is only used to update the weights of the generator.

GANs have been used in a wide variety of problems from the creation of artwork via style transfer to the creation of completely new imagery. This research will focus on the latter.

There are many challenges with taking such an approach to creating a data set with sufficient fidelity to justify its use in the field.

One such challenge is that the creation of synthetic data with GANs may result in a model that has over-fit the generated data. This may lead to reduced performance when the model is used to infer on real data. Several approaches will be explored to overcome this issue and facilitate the creation of an optimum data set. By training a final model on both synthetic and real imagery, as well as applying data augmentation, we will reduce overfitting to this imagery.

As the generator network will be attempting to convince the discriminator that the generated images are real, a particularly convincing image may cause mode collapse. Mode collapse occurs when the generator consistently produces the same or a small group of outputs that succeed in getting past the discriminator. Should this occur there may be a lack of diversity in the synthetic training data. Proven methods of overcoming this issue consist of implementing a Wasserstein loss function [20] or incorporating a generator loss function [53]. Multiple generators may also be utilized to increase the diversity of synthetic data. The captured imagery will consist of a large variety of different poses and rotations to help reduce the likelihood that the networks will favour a single one.

2.7 Stylegans

StyleGANs overcame some of the issues presented by early implementations of generative adversarial networks (GANs). By borrowing from style transfer literature they proposed a new architecture that automatically learned to separate high-level attributes from stochastic variations [43]. This was done by separating features through adjustments in image resolution, pose and identity (coarse resolution) may be separated from stochastic variations such as freckles, eye and hair colour (fine resolution). The model architecture developed in the stylegans research deviated significantly from traditional GANs at the time, by removing the first layer of the feed-forward network (mapping network) through which the la-

tent code would be passed through. Instead, the input is passed through 8 fully connected layers mapping it to an intermediate latent space, this latent space is used to control the generator (synthesis network) by applying instance normalization at each convolutional layer, Gaussian noise with per-channel learned scaling factors applied and is added per channel after each convolution [43].
By using the intermediate latent space to assign instance normalization at each convolution, the learned affine transformation from the mapping network adaptively normalizes the noise-added output of the convolution with the relevant style twice at each resolution of the synthesis network. applying a targeted resolution-specific style when upscaling the learned constant

Fréchet inception distance (FID) was the metric chosen to evaluate the performance at various stages of the development of Stylegans. The prior standard for evaluating the quality of synthetic imagery was to use the Inception score, which used a pre-trained Inception V3 model to classify images from the 1,000 classes it has been trained on. Using the confidence of the predictions on the synthetic images and the integral of the marginal probability of the predicted classes the inception score is able to provide a quantifiable performance metric without the need to manually verify each of the generated images. However, IS does not measure how well these images compare to the real images [24].

“We introduce the Fréchet Inception Distance which captures the similarity of generated images to real ones better than the Inception Score.” - Heusel et al.[37]

FID provides a method of evaluating the performance of the generator in a GAN by comparing the similarities between the generated and real images [41]. Similarly to the inception score, FID uses an inception model to evaluate the quality of synthetic images. However, FID compares the GANs-created images to real images by using the last pooling layer of the model to compare computer-vision-specific features of the images [24]. A Gaussian is the maximum entropy distribution for the first two moments, the mean and covariance. The measured difference between the two Gaussians, one for real data and another for synthetic data, is the Fréchet Distance [37].

StyleGAN2 is a development from the code produced in an NVIDIA Research paper[43], Analyzing and Improving the Image Quality of StyleGAN, which sought to improve the generated image quality of the original StyleGAN network by redesigning the generator network and proposing changes to the training methods.

Synthetic data was generated using a style-based GAN architecture, StyleGAN2-ADA-Pytorch. Adaptive discriminator augmentation (ADA) is leveraged to dynamically adjust the augmentation probability, the purpose of this technique is to increase the probability of augmenting the images when too much overfitting occurs and to decrease the probability when there is too little overfitting [42].

2.8 Interpretable Machine Learning

Interpretable ML techniques are rapidly gaining traction as they provide algorithms that may leverage the power of machine learning for pattern recognition, while also being able to present the contributing factors of an inference. ML models that are able to provide comprehensible justification for their decisions intrinsically promote trust in the developed model. The use of ML models for determining the result of loan applications has been under the spotlight in recent years [35], and mistrust of such systems has grown. Interpretable techniques now offer the potential for institutions to implement powerful automated systems that are also accountable.

Interpret ML [62], is an open-source project that enables further insight into interpretable ′glass-box′ models such as decision trees, logistic regression and linear regression both on a global scale; to better understand the importance of input features, and on a local scale; for understanding what particular features led towards a decision. Black box models trained outside the project may also be explained, but only on the local scale using interpretability techniques such as SHAP [50] and LIME [70].

2.9 Related Work

Some of the earliest research in fish length estimation using computer vision techniques was done in the early 1990s, Nielson et al. (1991) [60] discussed the potential uses of an automated system for quality assurance using vision techniques, though they summarised that commercially available equipment to perform this with the right precision, was at the time unavailable. A paper in 1993 Arnarson et al. described a prototype machine in which fish were passed under a video camera over a conveyor belt and were then sorted into bins, achieving 99% accuracy for flatfish and round fish [21].

Work by Qiu et al. [68] investigated the use of transfer learning for image classification on a popular fish dataset in 2018. Their use of pre-trained bilinear convolutional neural networks showed some success, by improving the accuracy of popular networks, though the computational load was significantly increased. Prior research into the use of machine learning-based computer vision techniques in the fishing industry has shown some success. Monkman et al [2019] [55] looked at performing object detection on side-face images of European sea bass and was able to predict the location of fish in images with a mean intersection over union (IOU) of 93%. In this research fiducial markers of varying sizes were used for calibrating images and provided a point of reference when measuring the length of the fish. Raw imagery in this dataset consisted of the side-profile of fish, imagery was horizontally aligned with the fish. Such an approach meant that using the difference between the minimum and maximum x values for bounding boxes could be converted to the total length of the fish by calculating the pixel to millimetre ratio from the known calibration object in each image. Such an approach provides insight into how length estimating may be tackled. However, though fruitful in its accuracy when assessing estimated length, this method will not be directly translatable to many real-world use cases. Imagery datasets rarely consist of such favourable examples where a single axis may be used to estimate length.

Stereo optical systems are relatively popular in the use of monitoring fish in water, the National Oceanic and Atmospheric Administration have used baited remote underwater video stations (BRUVS) equipped with a stereo video system for at least 10 years [61]. The use of stereo video systems opens up the opportunity for distance to be measured. Rodriguez et al. [72] paired this technology with computer vision, namely background subtraction, to generate a segmentation for the fish by separating each pixel into a foreground or background category. The output of this process was a binary representation of the candidate foreground from which size, area, and length to height ratios are used to determine whether the object should be considered a fish or not. If determined to be a Fish its size was then measured.

Álvarez-Ellacuría et al. [65] sought to use a different approach for the estimation of the length of the European hake. Instance segmentation was used to identify individual fish in every image by using polygons. This provides an advantage over bounding boxes as polygons are not bound to just two axes, meaning regardless of rotation a length may be estimated. As the data used for this research consisted of hake in boxes, stacked over one another, many of the fish were not fully visible. To combat this, researchers segmented the heads of fish and used known head to total length relationships to extrapolate the total length per fish detected. This approach was able to identify 87% of heads in the imagery. However, the individual level precision for newly measured fish was $\pm 10.0cm$ for fish measuring 20-27.5cm, with the unsigned median difference between measured and estimated fish lengths being 1.1cm.

More recently, Palmer et al. [65] also looked into the use of instance segmentation to measure the number of obscured fish in buckets. Focusing on the common dolphinfish, an important part of the commercial fishing industry in the Mediterranean. Their use of high-resolution imagery and a larger dataset provided much insight into the benefit of segmentation approaches and was able to achieve an accuracy of 86.10% on a dataset consisting of 4117 fish from 276 images. A different approach was used to estimate the length of fish here. As the weight of each bucket was also measured, researchers were able to use the bucket weight and the predicted number of fish in each bucket to estimate the mean fish length per bucket. Deviations between observations and estimates ranged between -7.4 and 4.8 centimetres.

The use of GANs for the creation of synthetic training data has been explored prior to this work, particularly in the field of medical imaging, where data is not widely available or protected due to privacy. [32] Eilertsen et al (2021) provided empirical evidence supporting the use of ensemble GANs in the creation of synthetic data. The approach in this research used entirely synthetic imagery for training data to overcome the lack of available training data for tumour classification. An ensemble of GANs was used here to reduce trained models overfitting to synthetic data. Though these models did not perform as well as a model trained entirely on the real data used in training the GANs, their performance was still notable and their techniques to avoid overfitting may prove useful for this research.

Data gathering for modern deep neural networks often requires large training datasets to infer reliably. Researchers at Nvidia have recently sought to address this by automating the process of segmenting synthetic images. DatasetGAN [86], using styleGAN as a generative backbone, seeks to reduce the amount of manual annotating required when training segmentation models by using decoded latent GAN code to automate semantic segmentation of images. Their proposed method only requires a few of the synthetic images to be manually segmented to train the discriminator.

Álvarez-Ellacuría et al's work [18] on Image-based, unsupervised estimating fish size from commercial landings using deep learning set out to achieve a similar goal to that of this research, to automate reviewing of fish length to reduce the exploitation of stocks. This work looked at estimating the length of hake based on the size of the fish's head, as the data in this research consisted of images of boxed hake the entire length of the fish was not always in view, as such, the head was masked using mask-RCNN and the total length of the fish was inferred using known and measured head to total length ratios for hake. Though the mask-RCNN approach worked well, identifying 87% of hake heads, the individual level precision for newly measured fish length, for fish in the 20-27.5cm range, was ± 10.0cm. Their research suggests that insufficient training data for fish at various postures may have led to this result, as well as variance relating to the sex of the fish that were unaccounted for.

2.10 Summary

Several challenges with using the generated imagery for length estimation are present. There is no true value for the generated fish's length nor will a ruler be present in this imagery. Using the pixel length of the fish should suffice as generated images will also be of the same perspective as those used in training.

Semantic segmentation models can also require large amounts of data and be difficult to train, we suggest using a pre-trained model, with frozen lower layers of the discriminator trained on ImageNet or COCO and fine-tuning the model for this problem. This was successfully performed in both 'Freeze the Discriminator' [54] and 'Transforming the output of GANs by fine-tuning them with features from different datasets' [23].

This research aims to leverage recent advances in object detection and generative adversarial networks to tackle a prominent issue in the fishing industry. GANs will be used to generate synthetic imagery, based on data gathered on the vessel. This will be used to create a diverse dataset that is able to identify, using an EfficientDet object detector, and estimate the length of tarakihi fish to automate the process of discarding undersized fish. This will facilitate the fisheries industry meeting compliance while providing a non-invasive method to measure the tarakihi population and size.

Chapter 3

Data Collection and Pre-processing

In this section, we discuss the different approaches taken to gather the data used in the later stages of this research. This data is the source for all models and processes developed in later chapters. Each iteration of the data collection process is described and key findings that altered the methods for gathering future data are explained.

Some of the methods used for deriving fish length from images and the ground control points from a checkerboard pattern are mentioned, at a high level, here to better explain the iterative data collection process. These methods are later discussed in greater detail.

3.1 Introduction

Data collection conducted in this stage of the research was an iterative process. The data collection process was not done just by collecting the information to be used in training computer vision learning algorithms but also to provide evidence on the methods required to most accurately infer the length of a fish from a single camera.

Three factory visits were conducted to gather the data to be used in this research. Each visit uses the findings from the last to improve our data collection process to allow us to gather a varied dataset that is similar to the environment found on fishing vessels. In New Zealand, tarakihi management strategies enforced by the government [12] require sub-minimum (25 centimeters) size catches to be reported. Therefore a below one centimeter average distance of inferred predictions to ground truth lengths was set as the target for this research.

Much of the research in AI systems over the past decade has focused on developing new methods and architectures, as result learning networks have improved significantly [78]. Many practical applications of these networks now have more to gain by moving away from developing the architecture and instead improving the data. As such machine learning research is beginning to move from a focus on improving model architectures toward the data-centric approach of systematically engineering the data that may be used to successfully develop these models [78].

Modern learning networks are often trained on large public datasets such as LSUN [85] or ImageNet, which contains 1.2 million images for object localisation [75]. Datasets of this size are not readily available in many industries, including the fishing industry, and their development may face significant challenges, particularly when concerning the privacy of individuals. The performance of architectures developed using these large datasets suffers from problems such as overfitting when data is scarce.

> ”Synthetic data is an important tool in the tool chest of data-centric AI” - Andrew NG [78]

To explore the potential for synthetic data to be used in tackling the data scarcity problem in the fishing industry the data gathered in this stage was also to be used to train a stylegans model, allowing the creation of a larger dataset through the generation of synthetic imagery.

From this dataset, we go on to train a segmentation model and develop systems to predict lengths from inferred segmentation masks. Once these two milestones have been achieved we can then evaluate the performance of the system when synthetic data is used to train the segmentation model.

Where this research differed from past literature is that fish were removed from the buckets they are auctioned from, meaning that where other research had to approximate length from the head size [19] or average length per box [65], this research focused on directly measuring the length of individual fish from imagery.

The ultimate goal of this research is to develop a system that would allow catch data for undersized fish to be gathered directly from footage gained on the vessel. Despite this, our research also had data collection limited to an auction centre. As such, efforts were made to try and replicate what the imagery would look like if it were gathered on a vessel. During the first and third visits, a metal surface was used to provide a similar backdrop to that of the metal surfaces found in the fish handling areas of commercial fishing vessels [10]. The second and third visits were conducted under shade in natural light, as the main source of light during the day on these vessels will be natural.

3.2 First Factory Visit

For the first factory visit data collection consisted of a handheld camera, facing top-down over a metal table. Two buckets were placed on the metal table for ease of access, this allowed us to quickly take a fish from the bucket containing unseen fish and display it below the camera. After placing an individual fish on the table, it would be rotated and flipped to maximise the data gained from each individual. As all fish handled as part of this research were provided by the fishery and were ultimately to be sold on, special care was taken to avoid keeping the frozen fish outside of the buckets of ice for a prolonged period.

The camera used in this research was a custom-built camera used specifically for fishing vessels, though when on-vessel this camera is fixed, it was established that holding the camera in hand would allow us to gather a large amount of data at various angles and distances. The camera used for gathering the videos for this research had a resolution of 960x540 and recorded imagery at 15 frames per second.

A checkerboard pattern with a known tile size was used for calibrating the camera. It also provided the information necessary, in the form of a ground control point, to track the position of the camera in 3D space relative to the table. Duraprint paper was used for the material of the calibration pattern as this provided a waterproof surface that is tear-resistant and less reflective than a laminated piece of paper, all factors that may distort measurements.

The benefit of this approach was to reduce overfitting to a particular setup. Footage from a free camera is less likely to lead to a neural network model only learning features that are apparent from a certain distance or angle. Thus reducing a loss in accuracy across the varying distances and angles of footage from different boats.

As the purpose of the dataset gathered in this visit was to provide us with a diverse imagery dataset, fish observed in this factory visit were not measured on-site. As there exists no ground truth for the lengths of these fish this dataset was not used for evaluating the accuracy of predicted lengths.

A challenge that arose from this approach was simulating fish in motion. Though one of the targets of future work is to be able to estimate length from live fish, data gathered in the auction room was not using live samples, as such fish had to be placed in positions that, though not ideal for modelling efforts, were more similar to imagery that may occur with fish moving down a chute or being passed at high speeds during sorting on a vessel. As shown in Figure 3.1 fish in motion present a significant amount of motion blur, making a precise length estimation more difficult. Holding the camera in hand also presented some issues. The large variability in the position of the camera and position of the fish on the table

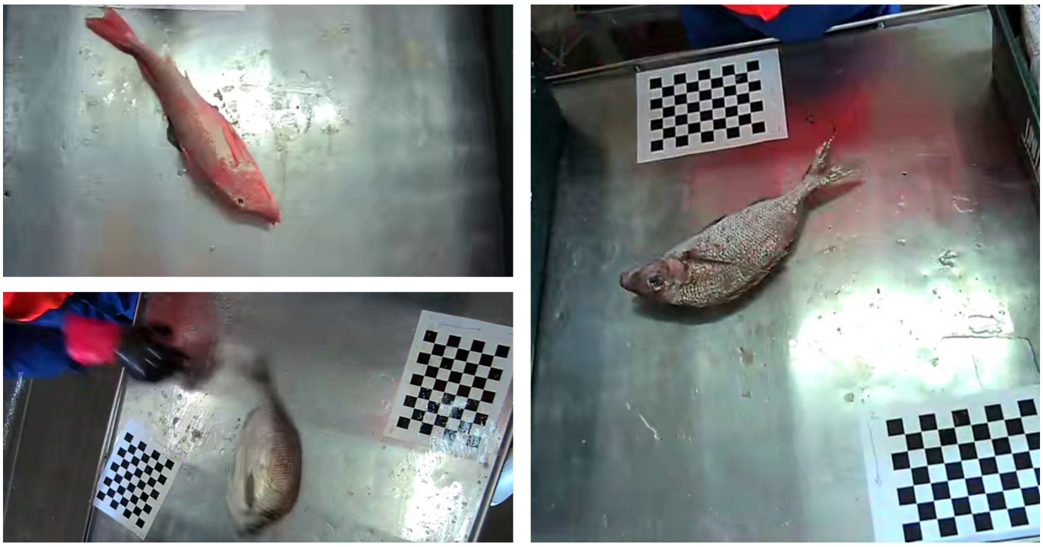


Figure 3.1: Examples of imagery taken. The top left is a gurnard. The bottom left is a tarakihi in motion. The right is a clear example of tarakihi.

resulted in some fish not being fully visible in the frame or the view of these fish being obstructed by the buckets that held the fish. The importance of a calibration pattern being present in the imagery is made more significant when the distance and angle of the camera are highly variable. However, by moving the camera to capture many different perspectives, to allow for a more varied dataset, the calibration pattern was not always fully visible, or visible at all in some of the images. The fish used in this research had been frozen whilst stacked in the boxes, this factor helped with simulating the curl that may be observed with live fish. from the 15 buckets of fish provided, with 12 of tarakihi and 3 of fish common in a tarakihi haul, these other fish were largely comprised of red Gurnard. A total of 13 videos were gathered, these videos ranged in lengths of up to 15 minutes. In total 100,664 frames were extracted from these videos to create the first dataset for this research, though many did not contain images of tarakihi. This data had to be sorted and unusable data was removed.

3.3 Second Factory Visit

A second factory visit was conducted to gather additional imagery data. The lessons learned from the first visit were used to adapt the data acquisition process and gather a second dataset to support the development and evaluation of our length estimation process.

Though the approach of using a free camera was able to provide us with a dataset that contained a large variety of different observation angles for the measured fish there were also significant drawbacks. One such drawback was the lengths derived from such an approach suffered from large variations in the scale factor. The scale factor was a measured value for the known difference between pixel and millimeter distance, found by measuring the checkerboard calibration pattern. Depending on the position of the pattern relative to the fish the derived scale factor would lead to an over or under prediction of the true scale factor at the position of the fish. Using a free camera also meant that the scale factor had to be recalculated for every image, as the camera angle and distance were always moving.

A second issue with the free camera approach was the presence of unwanted objects in the image. one instance of which was the buckets of fish that were kept on the table for ease of access. The buckets containing both seen and unseen fish provided both an issue when training and inferring. The number of polygons required to create a training mask for a single image significantly increases when either bucket is present if it contains fish, as the number of fish present in the image increases. Annotating these images would increase the time necessary to develop our dataset. However, not annotating these objects would result in negative training for the segmentation model and lead to a model with worse performance.

The free camera approach was still useful for providing valuable data for training the segmentation model. A fixed camera proved to be necessary for accurate length predictions. To identify whether sub-centimetre inferred lengths were possible the camera was fixed above the table on the second visit. A metal frame was used to hold the camera stationary above the table, allowing us to have a more consistent scale factor, a visualisation of this set-up is shown in Figure 3.2. The buckets of fish were also kept below the table to ensure that they did not obstruct the view of either the fish or the calibration pattern.

The imagery from this visit was split into two distinct groups. The first consisted of images with a ruler present in which the lengths of the fish were measured on-site. The second category consisted of fish being placed on the table once on each side and at varying rotations. The ruler was not present in the imagery of this second group, as the lengths were not being measured at this stage, the calibration pattern was kept to ensure a point of reference for lengths.

Three buckets of tarakihi were used as part of this data collection process, totalling 15 videos which when extracted, produced 100,232 image frames. The fish from each bucket were placed on the table beneath the camera and their length was measured against a ruler also present on the table, after each individual was measured it was placed into a second bucket reserved for measured fish. After all the fish from a bucket were measured, they were taken from the secondary bucket that held the measured fish and placed on the table once more at varying rotations with both sides of each fish being shown to the camera. This was done to gather a separate set of data which may be used for training. Measuring the lengths of fish in person allowed us to gather some ground truth measurements to compare both the human-measured length from images and the prediction lengths from an automated process using computer vision. For each of the fish measured on-site a singular frame was extracted from the image dataset. This image was generally the image in which the fish was most clearly visible, if there were many frames in which the fish was clearly visible the frame closest to when the true length was recorded was selected. Lengths inferred from these images were then used to evaluate the performance of our lengths estimation process from the inferred masks

3.3.1 Key Findings from The Second Visit

The second factory visit provided us with insights on how to improve the accuracy of our length estimation that the approach used for gathering data in the first visit was unable to provide. A fixed camera provided us with the opportunity to manipulate the imagery to reduce distortion, which may affect our measurements of the length of fish, on a large scale as the camera and angle were fixed. By identifying the pixel coordinates of an object of known size the perspective of images could be warped reducing variations in the size of objects based on their distance from the camera. The edges of the table were used for this, as a rectangular table was used the width of the table was consistent on both the bottom and top sides. Though this was not apparent in the imagery as the camera was not perfectly centred and was closer to the left side of the table, making the right side appear smaller. By warping the perspective of the image using each corner of the table we were able to reduce the distortion present and conduct a bird's eye view transform.

In a sample set of 69 images from the second factory visit fish lengths were on average predicted to be 18.7mm greater than the measured length prior to the warping of the image dataset. An absolute average difference between measured and predicted lengths of 22.5mm was measured for these images, the median of the absolute differences was 17.2mm and an R^2 of 0.76 was measured.

By warping the perspective of the image through the use of a bird's eye transform inferred lengths of fish were improved. The absolute average difference for inferred lengths was reduced to 13mm for the 69 test samples, with a median absolute difference of 9.0 and an R^2 of 0.85. The greater difference between true and measured lengths when the image was not transformed is attributed to the calibration pattern being closer

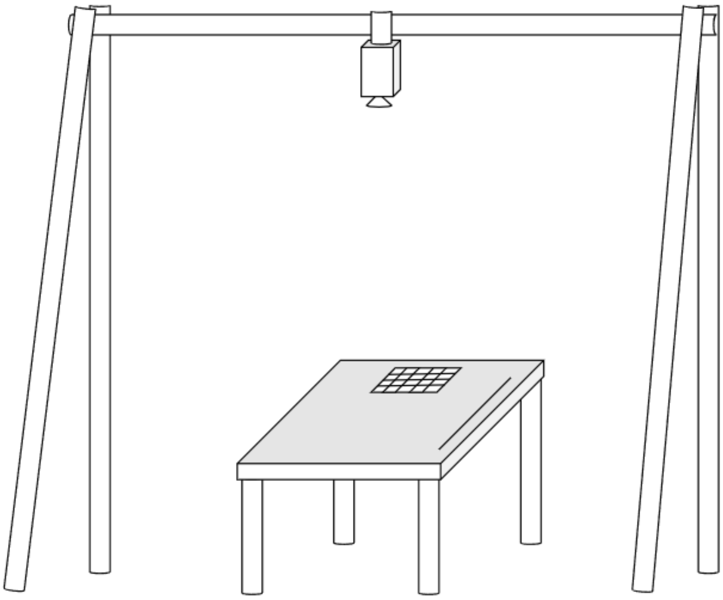


Figure 3.2: visualisation of the supporting structure and table placement used in the second factory visit. Poles are made up of steel and aluminium poles, steel poles were 1 metre in length and aluminium poles were 1.2 metres in length. The surface of the table was 610mm millimetres by 1220 millimetres.

to the camera than the fish. As a result, our scale factor at the point of the calibration pattern was greater than at the point of the fish resulting in inflated values for the estimated fish lengths. More information on this process can be found in subsection 4.2.1.

3.4 Third Factory Visit

The process of the third factory visit closely followed that of the second. The main differences were that this time a metal sheet was placed on top of the table used in the second visit. The intention of which was to make our surface more similar to the metal surface found on the areas of fishing vessels where fish are handled. This approach also removed the sharp line visible down the centre of the trestle table used in the second visit. Aluminium was chosen for the material as this provides a more matte surface and reduced the glare of various light sources around the factory. Though our camera had an anti-glare filter we wanted to minimise the adverse effects glare may have on training.

The main purpose of this visit was to gather more ground truth lengths of tarakihi. As such, every fish that was handled had its length recorded. For this visit researchers were able the acquire a measurement device used specifically for fish, which may provide more reliable ground truth measurements for the full length, fork lengths for 155 tarakihis were measured during this visit. 15 Videos were gathered as part of this visit, these produced 192,204 image frames when extracted. After measuring an individual fish, it was then placed below the ruler flat on the metal sheet at various orientations and both sides of the fish were often shown to the camera. Fish from the final bucket, bucket 6, were used a second time. After removing the ruler from the table, each of the fish was placed on the table again, in multiple positions. This would allow us to use the automated masking process developed on the data from the second visit to generate more masks, with a reduced likelihood of nearby objects affecting the contours of the fish.

A final set of 5 fish were measured at the end of the visit, with a free-held camera to ascertain how variations in the angle and distance of the camera affect length predictions. The length of each fish was recorded and it was then placed on the table, the camera was then moved to various positions while keeping the fish in frame. The fish was not placed in different positions in this stage as the focus was on how the angle of observation affected lengths rather than maximising the information gained from the individual fish.

Data from this visit was sorted differently from the data from the two prior visits. All frames in which an individual fish was above or on the table, regardless of whether it was clearly visible or obstructed by hands, were placed in a folder named with an ID, the species (tarakihi) and the measured length for that fish. This was done differently to the method for the second visit in which a single frame was selected for each measured fish, this frame being the image in which the fish was most clearly visible.

The imagery was stored in this manner as it allowed predicted lengths across multiple frames to be easily compared to the actual measured length for said individual.

3.5 Summary

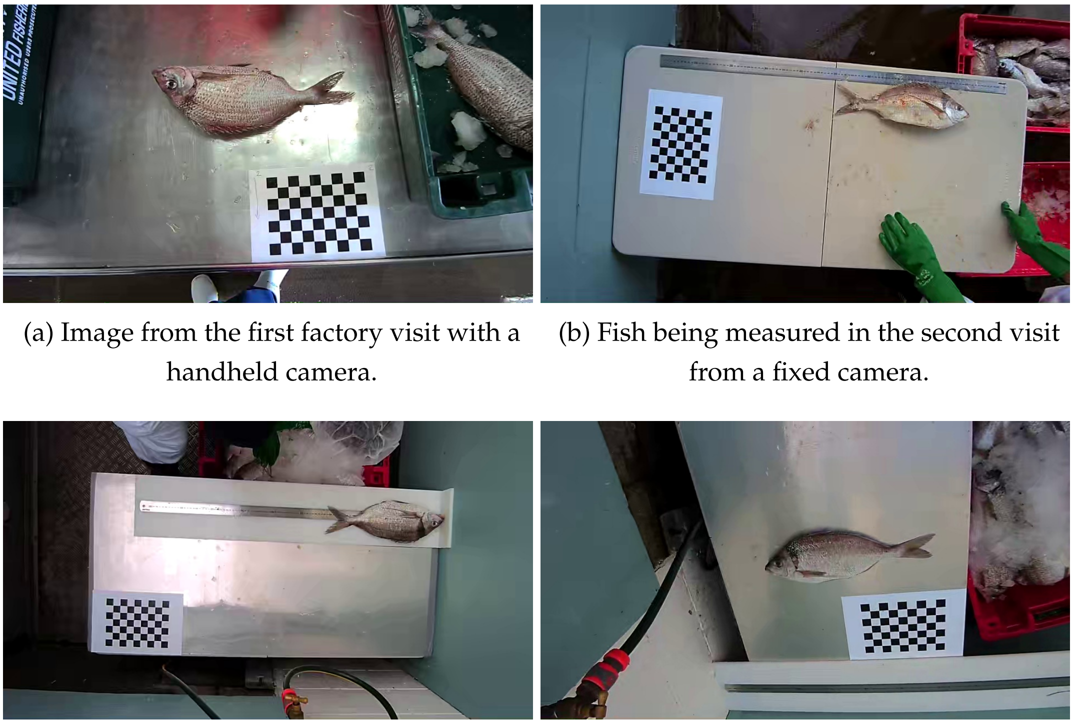
In summary, three different raw image datasets were collected as part of this research. The first utilised a free camera (held in hand) for collecting a large set of image data from a wide variety of observation angles. The second and third visits used a metal frame to suspend the camera above the table on which the fish were measured, allowed a more consistent view of the area of interest and made possible the removal of unwanted objects that were visible in the dataset. The second visit used a plastic trestle table as the surface on which the fish were presented, while the first and third visits used a metal surface, which is more similar to what may be found on a commercial fishing vessel.

Only data from the third visit was used in training the final Stylegans model, though data from the first visit was used in an early implementation. This dataset was chosen for training the generative network as synthetic images would then have a grey metallic background. The fixed camera with a bird's eye perspective of the observation area in this image set meant that images cropped around fish were less likely to contain many background objects. 700 synthetic images were annotated and used to compare model performance when trained on synthesised imagery. The synthetic image generation process is described in subsection 5.2.3.

A separate set of data gathered during the third factory visit was taken while holding the camera in hand, an example is shown in the fourth image (d) of Figure 3.3. This was done to allow a comparison of predicted lengths at different camera angles, and this process is explained in more detail in section 4.3.

Images from the fixed camera datasets, from visits 2 and 3 (example shown in the second (b) and third image (c) from Figure 3.3, respectively), were transformed for use in training the segmentation model, creating two subsets of the data from these visits. A fixed perspective was required for this transformation to avoid having to calculate the parameters for every image (see section 4.2.1 for detail).

Images from the first visit which contained background fish, that were not annotated, such as in Figure 3.3, were cropped around the annotated fish in the centre of the image, to avoid negative training on fish that were not annotated in buckets at either side of the image. Tables 3.1 and 3.2 show a breakdown of the datasets created from the imagery gathered from all three factory visits, and the synthesized imagery.



(a) Image from the first factory visit with a handheld camera.

(b) Fish being measured in the second visit from a fixed camera.

(c) Fish being measured in the third visit with a fixed camera.

(d) Image from the third visit with a handheld camera.

Figure 3.3: Images comparing the different datasets gathered as part of this research. The first image (a) displays an image from the first factory visit in which the camera was held in hand. This example shows the fish currently being observed, in the centre of the table, as well as a secondary fish in the bucket on the right. Images the second (b) and third images (c) show the fixed perspective from the second and third visits. The final image (d), also from the third visit, was taken as part of our test on the impact of different camera observation angles on the predicted lengths of fish, described in section 4.3.

Table 3.1: Summary for image datasets that were annotated

Annotated Datasets Summary			
Dataset	Number	image Size	Use
1) cropped images, visit 1	517	variable	segmentation: training,test and validation
2) Warped images, visit 2	1554	738 × 378	Segmentation: training,test and validation
3) Warped images, visit 3	381	636 × 323	Segmentation: training,test and validation
4) Synthetic images	700	320 × 320	Segmentation: training

Table 3.2: Summary for image datasets that were not annotated

Non-Annotated Datasets Summary			
Dataset	Number	image Size	Use
5) Warped images, visit 2	13247	732 × 376	Images for automating segmentation mask creation. Section 5.1
6) Measured fish, visit 3	56187	960 × 540	frames for extracting lengths from video. Section (6.4).
7) Third visit free camera	1220	960 × 540	Evaluating impact of camera position. Section (4.3).

Chapter 4

Data Handling

Ensuring the quality of data was consistently high was an important challenge that would facilitate achieving our goal of centimetre accurate length estimations. The methods for helping to achieve this goal at the data collection and handling stage are discussed in this chapter. This includes preprocessing methods as well as augmentation of the data to demonstrate an optimal process to achieve this goal.

The augmentation of data in machine learning provides the opportunity to create diversity in a dataset where there may be little, improve the quality of the data and bolster the amount of data available to solve machine learning problems. Augmentation that is conducted with expert knowledge may help reduce the model overfitting to the input data and facilitates the use of smaller datasets when tackling complex ML problems, such as image classification when using deep learning [66].

Augmentation, in this research, was not purely used to reduce overfitting to input data when training the networks. Transforming images from the second and third factory visits allowed us to overcome limitations observed in the first visit. This was done by applying a birds-eye transformation to the data from two of the three visits, allowing us to constrain the visible region, in images, only to the area of interest, this is discussed in subsection 4.2.1.

4.1 Chapter Goals

This chapter aims to present the various augmentations conducted in this research and investigate the importance of camera positioning. To achieve this aim, this chapter will:

- present the processes used in this research for augmenting data prior to training and good practices for handling data from which lengths are derived,
- provide evidence and justifications for the best positioning of the camera for gathering relevant image data by evaluating the various observation angles and the corresponding predicted lengths of fish,
- introduce interpretable techniques and explain the benefit such techniques bring to understanding our data, and how this may be leveraged to improve our systems.

4.2 Augmentation Method

Albumentation [25] is an extensive python library that provides a wide array of tools for fast image augmentation, which was used for augmenting data when training the segmentation model, four examples of this augmentation are shown in Figure 4.1. A total of 14 augmentations were used at this stage, though a limit of 9 could be used for a single example these included:

- Horizontal flip; with 50% probability.
- Shift Scale rotation; randomly applies affine transformations with a scaling factor of 0.5 and shift limit of 0.1. This was kept relatively low to retain the general shape of the fish.

- Padding if needed; ensures undersized images were padded to be 320×320. This allowed us to ensure all our images were the same size when training the model.

- Random crop; always randomly crops inputs to 320×320. The use of a random crop meant that we could avoid scaling all our images to be the correct size while also providing useful examples of partially visible fish, allowing for more consistent inferences when fish are partially visible.

- Gaussian noise; applies glass noise with a probability of 0.2. The use of added noise to our images enables a more robust model when images are of lower quality, or when visibility may be impaired. salt spray is a common example of this on commercial fishing vessels.

- Perspective transformation; applies a four-point perspective transformation of the image with a probability of 0.5.

- 90% probability of one of the three:
  - Contrast limited adaptive histogram equalization (CLAHE), handles amplification of contrast in 8×8 regions of the image to equalize the image, using bilinear interpolation to remove the boundaries of these regions.
  - Random Brightness; randomly changes the brightness of the image between a factor of -0.2 and 0.2.
  - Random Gamma randomly changes the gamma of the image between the limits of 80 and 120. Lighting conditions are highly variable on a partially covered deck out at sea, we adjust brightness and gamma in hopes of representing this.

- 90% probability of one of the three:

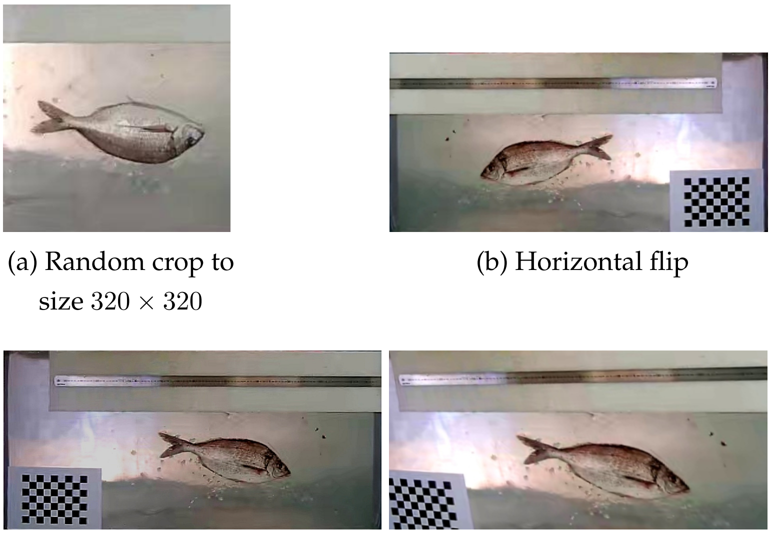
  - Sharpen; displays a sharpened version of the image with visibility in the range of 0.2 to 0.5, where 0 is a fully visible original image and 1 is a fully visible sharpened image.
  - Blur; applies a normalised box filter to the image with a kernel size of 3 to blur the image.
  - Motion Blur; applies blur to the image to give the perception of movement, by applying a kernel with a line of non-zero values, a kernel size of 3 was used here. This was included in addition to our attempts to create motion blur while capturing the data, as many images did not include motion blur.

- 90% probability of one of the two:
  - Random Contrast; randomly changes the contrast of the image between a factor of -0.2 and 0.2.
  - Hue saturation value; shifts the hue between -20 and 20, the saturation between -30 and 30, and the value between -20 and 20. The Perceived colour is variable due to changes in lighting conditions throughout the day. Hue and saturation changes were kept relatively low to recreate this effect when training.

4.2.1 Warping Images

To reduce distortion in our dataset a birds-eye transform of the image was conducted to only include the surface on which the fish was being displayed and measured. This provided us with two significant benefits to the quality of the data. Firstly, by allowing us to reduce the presence of items unnecessary or detrimental to the length estimation or segmentation systems, such as inferences of fish present elsewhere in the image.

The second expected benefit is a more consistent pixel-to-millimeter ratio. The calibration pattern, from which the ratio of pixels to millimeters



(a) Random crop to size 320 × 320

(b) Horizontal flip

(c) Sharpened image with alpha of 1

(d) Perspective transformation

Figure 4.1: 4 examples of the augmentations applied to data used for training the segmentation model.

for a single image is calculated, generally remains stationary. This means that the pixel-to-millimeter ratio is only calculated at a single point in the image, as fish are not placed in the same location as the calibration pattern their true pixel-to-millimeter ratio will be slightly different. Further distortion will be introduced by the angle of the camera, as the camera is unlikely to be positioned perfectly above the table, and from minor optical distortions from the lens design, though these are corrected, some margin of distortion is likely to be present.

The edges of the table were chosen as our ground control point for acquiring the pixel coordinates of the quadrangle vertices required to perform the perspective transformation, Microsoft Paint was used to acquire these point coordinates. Points at the edges of the table were chosen as the table seldom moved during the handling of the fish and its edges being visually separated from the background made it easier to select consistent pixel coordinates across different videos. This process was conducted for every 5 images or where it had visibly changed, these generally remained consistent for individual buckets. A Numpy [6] array was used to store each of these points, in the order of bottom left (bl), top left (tl), top right (tr), and bottom right(br). To calculate the size of the output image from the perspective transformation the maximum size of the widths and heights from the input image had to be calculated. The width and height were calculated using Pythagoras' Theorem to calculate the maximum hypotenuse for both the width at the bottom and top points and the height on the left and right of our corner points.

The width for the bottom of the image was calculated as follows.

$$width_{bottom} = \sqrt{(br_x - bl_x)^2 + (br_y - bl_y)^2} \tag{4.1}$$

where bl is the bottom left and br is the bottom right.

This was repeated using the coordinate points at the top of the table. The maximum width was taken from these two outputs and was used as the width for the transformed image. This process was also conducted on the left and right sides to calculate the maximum height to be used for the height of the transformed image.

The OpenCV function CV2.GetPerspectiveTransform [8] uses both the edge points from the original image and the max width and height from our desired region of interest to calculate a 3x3 perspective transformation matrix [8]. coordinates of the quadrangle vertices from the original image are multiplied with the transformation matrix and the top two values of the generated matrix are then divided by the bottom value to calculate the corresponding coordinates in the destination image.

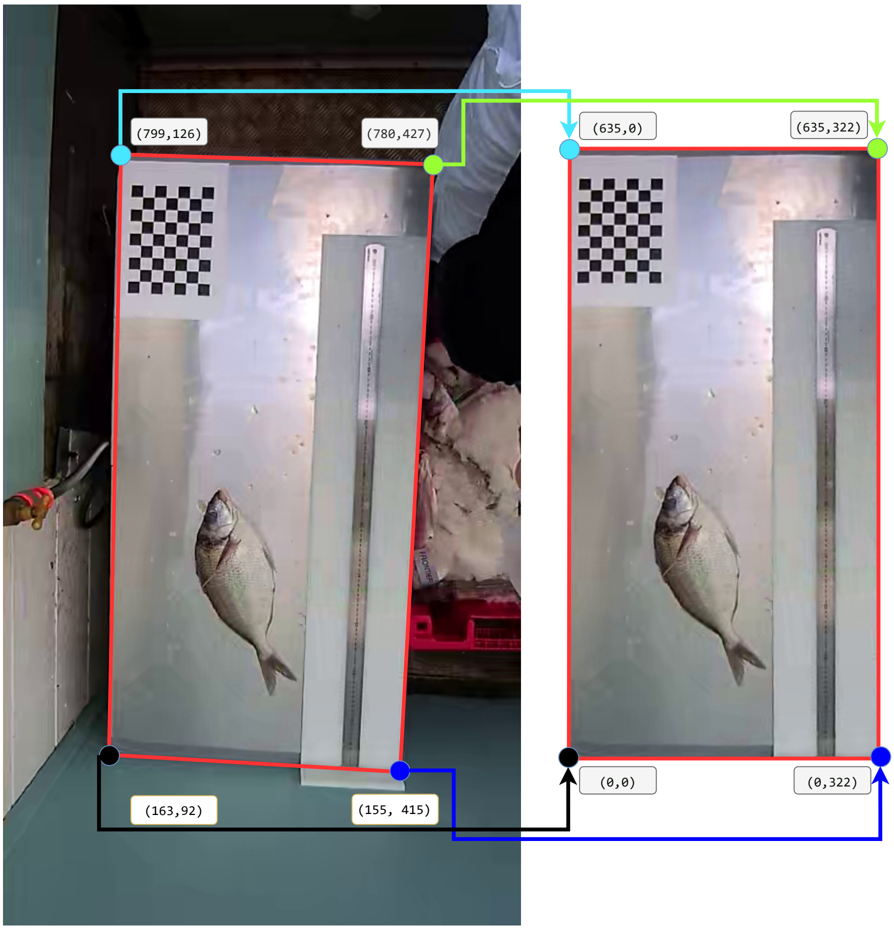


Figure 4.2: Image showing before and after performing a birds-eye transformation to an image from the third factory visit. Pixel coordinates are shown for the edges of the table in the left image and show the edge point coordinates of the second image.

The equation used in CV2.GetPerspectiveTransform [8] is as follows:

$$\begin{bmatrix} t_i x'_i \\ t_i y'_i \\ t_i \end{bmatrix} = TransformationMatrix * \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{4.2}$$

where output(i) = (X'_i, y'_i), source(i) = (x_i, y_i) and i = 0,1,2,3. An example of applying the matrix generated by CV2.GetPerspectiveTransform [8] to the bottom right vertex from the input image in Figure 4.2.1 is shown below.

$$\begin{bmatrix} 9.094e-01 & 2.252e-02 & -1.503e+02 \\ -5.104e-02 & 9.547e-01 & -7.951e+01 \\ -1.004e-04 & -6.126e-05 & 1 \end{bmatrix} * \begin{bmatrix} 780 \\ 427 \\ 1 \end{bmatrix} = \begin{bmatrix} 568.655 \\ 288.357 \\ 0.895 \end{bmatrix} \tag{4.3}$$

Using the outputted matrix by solving for $t_i x'_i = (568.655)$ and $t_i y'_i = (288.357), t_i = 0.895$ provide us with the coordinates of the corresponding pixel location in the output image $(635, 322)$. The OpenCV.warpPerspective [8] function was used to apply the transformation matrix to the entire image.

$$dest(x, y) = src(\frac{M_{11}x, M_{12}y, M_{13}}{M_{31}x, M_{32}y, M_{33}}, \frac{M_{21}x, M_{22}y, M_{23}}{M_{31}x, M_{32}y, M_{33}}) \tag{4.4}$$

Where M is the transformation matrix, dest is the output image, src is the original input image.

4.3 The Importance of Camera Positioning

The camera's role in this research is to map three-dimensional points and map them to a two-dimensional plane from which machine learning may be applied to extract information useful in estimating the length of an observed fish. In order to achieve this the world coordinates of an object must undergo a rigid transformation, from the 3-D world coordinate system to the 3-D camera's coordinate system; represented as extrinsic parameters, and then a projective transformation to 2-D image coordinates, represented as intrinsic parameters [3].

Various factors can affect the size of objects observed through a modern camera. the use of a lens may result in the distortion of an image and focal lengths can affect the perceived size of distant objects [63].

A factor that is of particular importance in this research is the position of the camera relative to the calibration pattern, and how it may alter the length calculation due to variations in the scaling factor at different locations in a three-dimensional space. Objects closer to the camera will require a smaller pixel-to-millimeter ratio as they are represented by a larger number of pixels. Identifying the importance of the relativity of our object of interest (fish) with regards to the source of the scale factor (calibration pattern) was an important part of justifying the use and best position of a fixed camera.

Extrinsic parameters include two operations, a translation and a rotation, these represent the transformation from the world points to the camera coordinate system [3]. The camera's coordinate system measures objects relative to the camera and begins at its optical centre. The Z axis extends from the optical centre outwards from the camera, while the X and Y axis make up the 2-D image plane[3].

The intrinsic matrix parameters consist of the focal length, optical centre and the skew coefficient [3], it is defined in Equation 4.5 and the intrinsic matrix from 20 images is shown to its right. To calculate this matrix the camera parameters must first be estimated, this may also be done by using the calibration pattern.

$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix} = \begin{bmatrix} 505.778 & 0 & 0 \\ 0 & 532.871 & 0 \\ 486.487 & 137.128 & 1 \end{bmatrix} \tag{4.5}$$

where f represents the focal length in pixels, c is the principal point in pixels, and s is the skew coefficient [3].

By finding image points for the observed checkerboard pattern in the image; in the form of pixel coordinates, and comparing these to the known world points of the calibration pattern; projected onto the image, we may calculate the reprojection errors. These are a measure for the euclidean distance between the true position of a point and its projected points in the image. Reprojection errors closer to zero signify that projected image points are close to the true locations of the points (world points).

The world points consist of a matrix of the X and Y coordinates of the key points; points represented by the internal corners of the checkerboard pattern. Our pattern was made up of nine tiles by seven tiles, and thus contained eight by six coordinates for the world points of the calibration pattern. These are calculated by using the known length of the tiles, 23 millimeters, and storing the distance of the top left of each tile, in both axes, from the top left of the image.

By understanding the change of projected world coordinates based on the position of our sensor, we may estimate the sensors position in 3d space. We were then able to analyse the impact of adjustments in the camera's orientation and transformation on estimated lengths by using the sensor's estimated position and the predicted length from the corresponding inference mask. The process of deriving these lengths is explained in section 6.2.

4.3.1 Implementation

To quantify the impact that different locations and orientations have on the predicted lengths of objects, the extrinsic parameters of the camera first had to be calculated. This was done by identifying its position in 3D space, in the form of a translation matrix, and its rotation, in the form of a rotation matrix, relative to the calibration pattern. The camera parameters were calculated by using 20 images in which the camera position varied

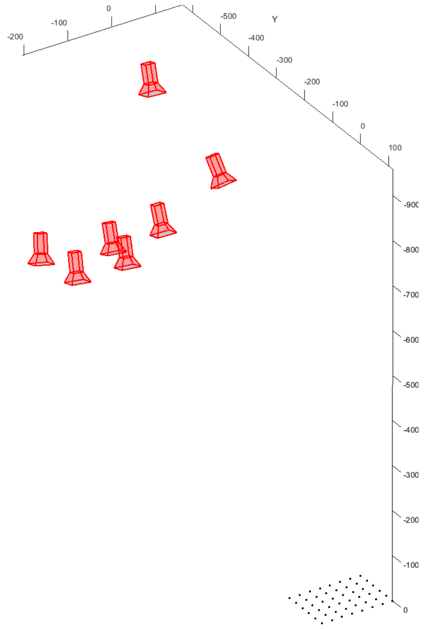


Figure 4.3: Visualisation of some of the camera (red) positions above the calibration pattern (grid on the bottom right).

while the calibration pattern remained fixed on the table below. Images were chosen based on the location and orientation of the camera significantly deviated so that the calibration pattern's location in the 2D image pane would vary from image to image. By varying the position of the calibration pattern in the image we hoped to reduce the likelihood of estimated camera parameters being altered by distortions present in a single area of the image. Barrel distortion and pincushion distortion will shrink, or enlarge objects close to the edges of our image respectively.

After calculating the camera parameters each of the images was read and the extrinsics were calculated. Both the rotation matrix and the translation vector were found and used to calculate the camera's location and orientation in the 3D scene. A visualisation of seven of the camera locations and orientations is shown in Figure 4.3.

The rotation matrix is a 3x3 matrix, that describes a rotation in three-dimensional space. Paired with a translation vector of size 1x3, world coordinates may be transformed into camera coordinates [2]. Both the rotation matrix and translation vector are unique to each image, as the camera was in motion throughout this test. As such, extrinsic parameters for each camera had to be moved into the world coordinate system, which was fixed by having a stationary calibration pattern, to be compared. As shown in Figure 4.3 this origin of our world coordinate system begins at the top left of the calibration pattern and the axes extend out from this location. This was used to calculate the camera's orientation relative to the control point, the pattern. Each orientation matrix was converted into a 1x3 rotation vector, describing deviation from the axes in radians. This 1x3 rotation vector made it easier to explain the deviation from a perfect birds-eye perspective, and therefore evaluate the impact of this deviation on inferred lengths.

Our exploration into the impact of camera pose on length began by looking at the variations in orientation. As image data was ordered by a Unix timestamp for the video and the corresponding frame number for each image, our visualisation of the various camera orientations in Figure 4.4 shows the observation angle changing throughout time. The camera remained fixed until around the 400^{th} index as can be seen by orientations in this range remaining mostly constant. After which the camera's orientation and location were altered, starting at a high position and moving left to right (translation). Care was given to maintaining the optical centre facing towards the fish. Images in which the calibration pattern was not fully visible were removed from the dataset as such instances would not match up with our world coordinate matrix and could affect readings.

Movements were fairly constrained by physical limitations on site, as the table was placed in a corner of the factory to limit its movement in the world coordinate system this meant that the camera could not be moved into positive Y values (below the calibration pattern shown in Figure 4.4)

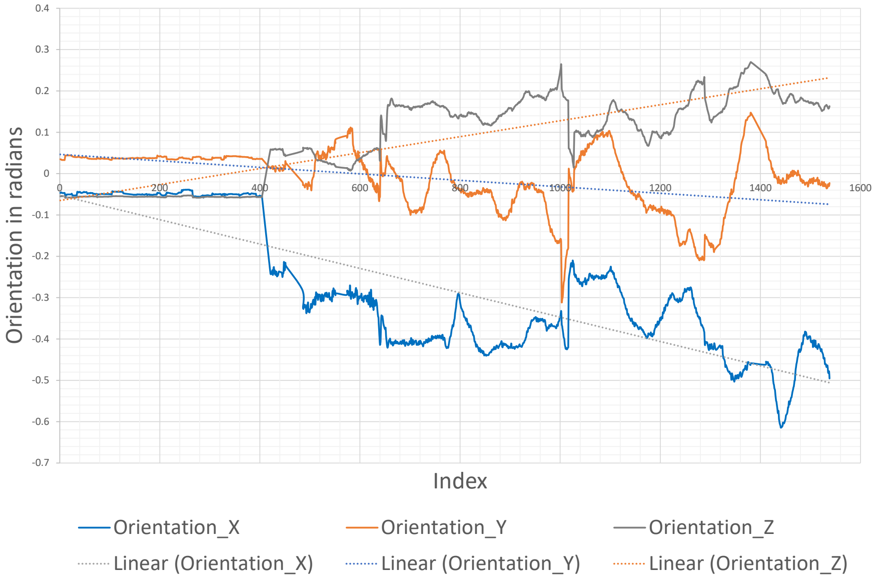


Figure 4.4: Plot showing orientation values per axis plotted on the Y axis, and their index in the table on the X axis.

or further into the negative values of the X axis (to the left of the calibration pattern in Figure 4.4).

Every still image, from which the extrinsic parameters were found, was also passed through the inference process and a length in millimeters was derived. The predicted length of the fish was subtracted from the true measured length to provide us with the difference in estimated length. Predicted lengths that were kept were capped at 3 standard deviations above or below the true length of the fish to keep most of the data points but reduce the number of outliers, such as images in which the fish was only partially visible or an unusually poor inference. The dataset used in this test consisted of 1383 images, from 3 different fish. Images from a fixed camera that were used in this test did not have a birds-eye transform

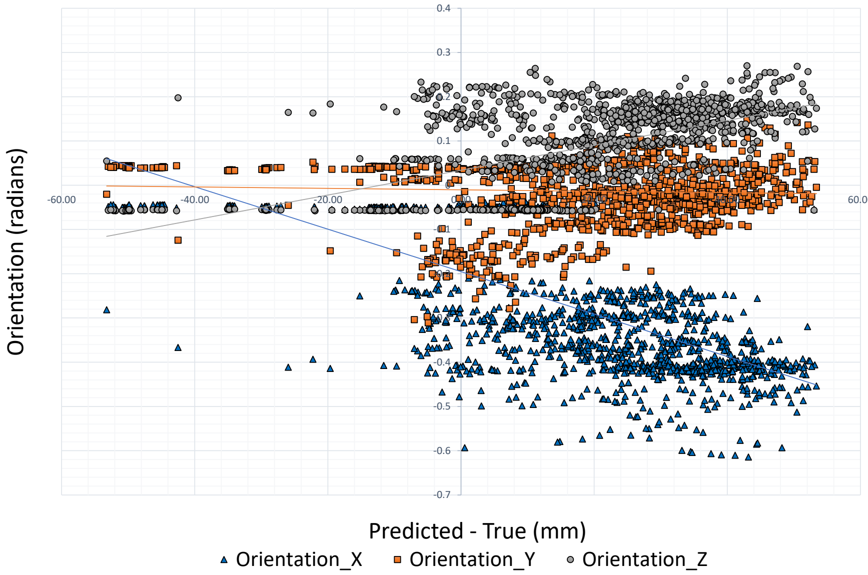


Figure 4.5: Scatter plot showing the difference between the predicted and true length on the X axis, per rotation axis, and the orientation value on the Y axis.

applied as the focus was entirely on camera location and orientation.

Linear regression was plotted from this data to identify whether there was any correlation between camera orientation and the predicted length. Figure 4.5 shows the results of this regression. Greater negative rotations in the X axis and higher positive rotations in the Z axis tended to be associated with a higher predicted length, resulting in larger positive values on the X axis. This was in line with expectations as the camera was placed in positions in which the fish was in the foreground and the calibration pattern was in the background, resulting in an inflated scale factor at the position of the fish, and a larger predicted length. However, due to the restrictions discussed earlier, little data was available for the inverse, when the calibration was closer to the sensor than the object of interest.

Due to the use of signed values in describing the rotation, in radians, of our camera, values close to π but of opposite signs would appear very distant when plotted but in 3D space would be very close. The use of absolute values to represent deviation from a camera oriented perfectly at 0 radians on each axis was considered. However, as our rotational values remain below $\frac{\pi}{2}$ this was not deemed necessary.

Rotation about the Y axis did not present a significant change in the predicted length of our object of interest from a linear regression. Though this value did vary significantly throughout this test as shown in Figure 4.4, its orientation, as measured in radians, did not reach deviate to values as great as orientation on either of the two other axes.

The location in the three-dimensional space of the sensor relative to the pattern was not considered with this simple linear regression, nor were pairwise relationships. In order to explore this the orientation, location and predicted length were paired for each frame and an interpretable generalised additive model (GAM) [49] was trained on the data to find patterns and explain the relationship between orientation, location and the predicted length.

A GAM was used as it leverages ”shape functions”, a non-linear function that describes the relationship between input and target features. Its complexity is arbitrary and so may be used to explain these relationships with greater accuracy than a linear model. A GAM specifically from InterpretML research [62], was used as it considers pairwise relationships, which is not the case in traditional GAM models, which is described in detail in section 5.3.2. For this experiment, we used an explainable boosting regression model to identify the most significant inputs that result in a large gap between true and predicted lengths, as well as the extent of their impact on predicted lengths. Results for the feature importance are shown in Figure 4.6, and an example of the extent of the impact of change in orientation on the predicted length is provided in Figure 4.7.

4.4 Results and Analysis

4.4.1 Image Transformation

Though the method for transforming images proved valuable for providing a cleaner image set, an issue exists with the transformation of images, any change to the size of the calibration pattern relative to that of the fish when transforming images results in a skewed scale factor at the location of the fish.

This did not present issues in early tests on data gathered in the second visit as the camera was well centred above the table. However, depending on the position of the camera relative to the key points this transformation may lead to a reduction in the accuracy of length estimations. This was observed in data from the third factory visit where the size of the fish was increased relative to the pattern, resulting in an over-prediction of lengths of around two centimetres. Camera placement perfectly centred over the area of observation would reduce this loss in accuracy. Additionally, as the increase in size was consistent across the predicted lengths these may be amended by performing linear regression. However, at this stage of our research, we wish to minimise any factors that have a negative impact on the predicted lengths and as such, this transformation was only applied to training images.

Two new datasets were created by warping the images from the second and third visits. These were 13247 images from the second visit (dataset 6 in Table 3.2), which were later used to create a masked dataset (dataset 3 in Table 3.1) in subsection 5.2.1. The second dataset created included 56187 images from the third visit. The images from the third visit were initially to be used for deriving lengths. However, after later discovering the issues with using warped images for length estimation on this dataset, this was instead condensed down to only 381 images (dataset 4 in Table 3.2).

4.4.2 Camera Position

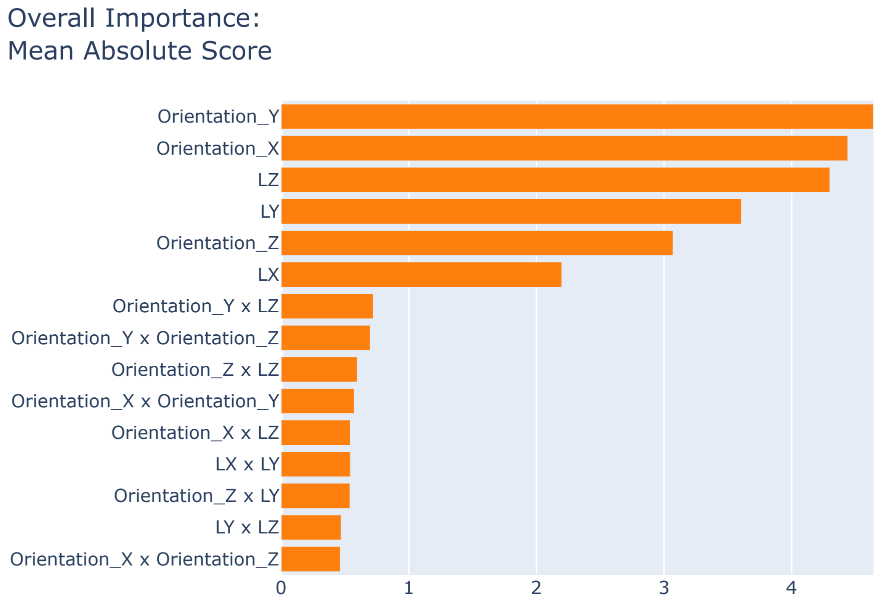


Figure 4.6: Chart for the measured feature importance from training an explainable boosting regression model. The X-axis displays the feature importance found for each of the six input variables, and significant pairwise relationships. Input variables included rotations on the X, Y and Z axes; measured in radians, and translation in 3D space along the X, Y and Z axes, measured in millimeters.

We had expected extreme observation angles to be detrimental to the accuracy of length estimations and our experiment in section 4.3 supported this hypothesis. Figure 4.6 shows the orientations on both the X and Y axis as well as the location along the Z axis are highly important when considering the change in predicted length compared to the true length of the observed fish. Justification for this importance is understood to be caused by a variation in the perceived size of the fish relative to the pattern.

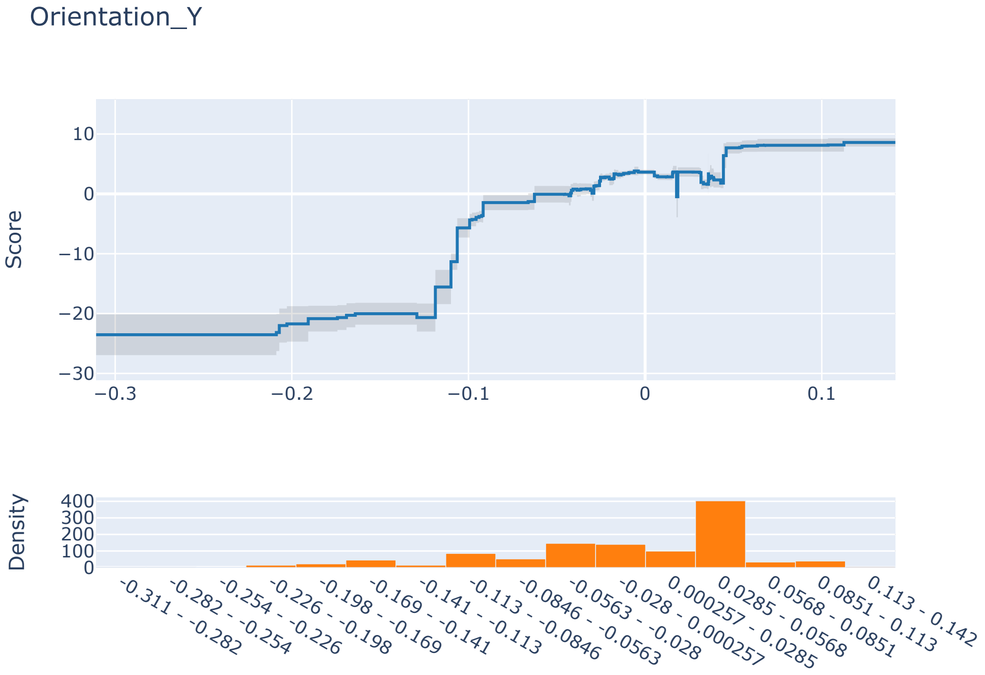


Figure 4.7: Shape function illustrating the orientation of the camera on the Y axis against the contribution this orientation has on the predicted length. Contribution scores are the expected change in predicted length when this level of orientation on the Y axis is observed.

An image in which the camera had a high orientation value, close to $\frac{\pi}{2}$ on the X or Y axis would need to be positioned closer to the table to keep the calibration pattern in view and would therefore have a translation value on the Z axis closer to zero, the inverse is also true. These variables were positively correlated and were found to have a clear impact on the predicted length of the fish.

The impact of the orientation on the Y axis as identified by our interpretable model is shown in Figure 4.7. Camera positions closer to the pattern, represented by higher values for the location on the Z axis (LZ), were found to result in an over prediction of the length of the fish. This is shown

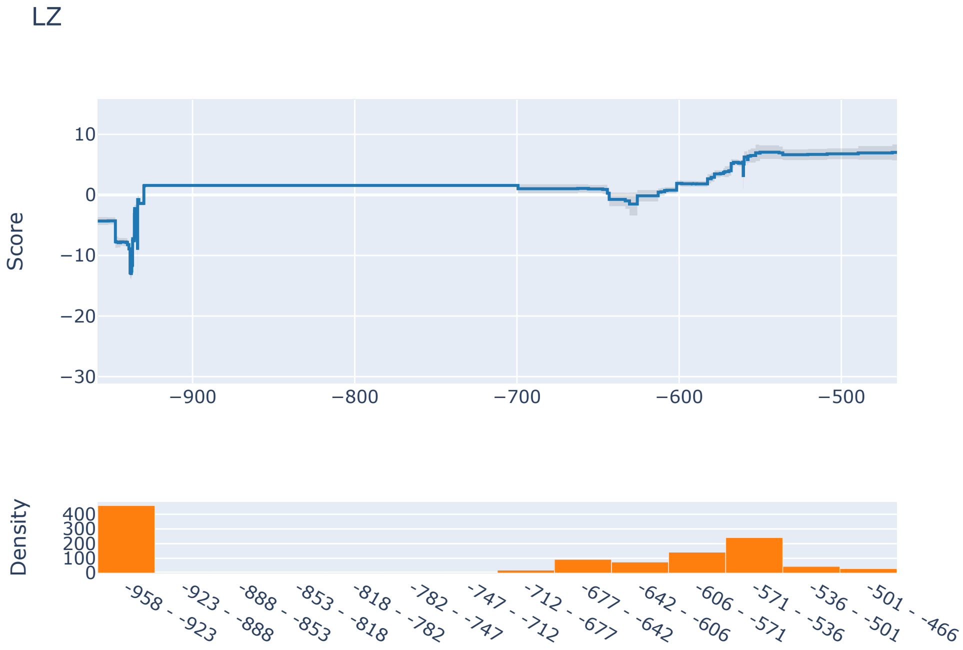


Figure 4.8: Shape function illustrating the contribution score for $predicted - true$ length (Y axis) at different locations of the camera on the Z axis of the world coordinate system (X-axis).

as a positive contribution score at distances above -608 millimeters on the X axis in Figure 4.8. Values below 930 millimeters were found to have a negative impact on the predicted length. As shown by the density plot below Figure 4.8, there are no observations for location values on the Z axis between -712 and -923. As such, the distance at which we would expect to begin seeing under-predicted lengths is difficult to determine, as contribution scores remain within ±2 for distances between -592 and -935.

However, variance in the accuracy of the projection was also likely to have an impact. Higher orientations and lower translation on the Z axis, suffered from greater reprojection errors when mapping the keypoints of the checkerboard pattern that were detected in the image to those known in our world coordinate system. Reprojection error provides us with a qualitative measure of the accuracy in our image of the true points projection [3]. The impact of such variance is likely to be minimal as reprojection errors in both fixed and free camera calibrations were within an acceptable range, with the overall mean error in fixed images at 0.08 pixels and that of the free camera at 0.16.

In contrast to the results from our linear regression orientation on the Y axis was found to be the feature with the greatest importance when used for training an explainable boosting machine. The target class for this experiment was the difference between the predicted and true length so our contribution scores illustrated in Figure 4.7 show the increase or decrease in the predicted length. An orientation on the Y axis of -0.15 radians would suggest an under prediction of 20 millimeters in this model, ceteris paribus. Fewer examples are available for Y values below -0.1, though the 45 examples between -0.169 and -0.141 suggest this is not an anomalous estimation.

4.5 Chapter Summary

In this chapter, we presented our pipeline for data augmentation, by describing the various augmentations applied when training the segmentation model. In addition to this, we explained the differences in handling data for training and for images from which lengths would be derived.

The process for applying a birds-eye transformation on training images was described in detail, and we provided an example of this application on an image from our dataset. This was performed with the intent of removing unwanted objects from the image prior to any augmentation. It was noted that this system was beneficial in removing non-annotated fish from below the (physical) table that were still visible at the edges of the images. However, the application of this process to images from which the length is derived has a detrimental effect on the estimated lengths of fish when the length of the calibration pattern relative to that of the fish is altered.

We introduced the use of GAMs for providing interpretable machine learning methods, providing us with greater insight into the relationships between input and target variables. These models were used to explore the impact of various camera angles on the difference between true and predicted lengths, providing a justification for the use of a fixed camera, placed directly above the object of interest. We established that not only does the image taken from a camera in this position have a scale factor that results in a more accurate millimeter length for the observed fish but also benefits from a smaller re-projection error.

Chapter 5

Image Segmentation

Image segmentation was chosen as it provides a method for identifying the fish in an image with pixel-level accuracy. Though some past research [55] used the width of bounding boxes to derive pixel lengths such an approach is not sensitive to rotation. As only two points may explain the inferred bounding boxes of an object detector, such an approach is limited by the requirement of fish being parallel to the X-axis of the image, as the width of the bounding box is not representative of the length of a fish otherwise.

The use of image segmentation provides a significant advantage in these cases in that the pixel lengths of fish may be directly identified from the two extremes of an inferred mask, described in section 6.2. This is a direct result of having an arbitrary number of points to describe our inference. This does mean that creating the training data for segmentation models requires more granular information than that of an object detection algorithm. However, it also provides us with a more detailed output, a mask of the original image, from which we may derive the contour of the inferred region, and evaluate the shape of inferred objects. This is explored in subsection 5.3.2.

There are drawbacks to this approach. The lack of localisation results in an absence of awareness of the limits of a single object. fish that are overlapping or very close will appear as a single object in the inferred mask from a segmentation model. There are methods to overcome this, such as Mask R-CNN [36] which leverages both the pixel-accurate inferences of a segmentation model as well as the localisation properties from object detection to provide instance segmentation. However, these were not explored in this research as our focus was on the length estimation of partitioned fish, these being individuals represented in the image with no pixels joining those of another fish, so instance segmentation was not required at this stage.

5.1 Chapter Goals

This chapter aims to describe the relevant methods for creating the data for training the segmentation model and will analyse our approach to evaluating the quality of image masks. We specifically aim to cover the following areas:

- how we created our mask datasets to train the segmentation model. This will cover two different approaches for creating segmentation masks.

- how we leverage generative adversarial networks to synthesize additional training data.

- how we automated the cleaning of large datasets by evaluating contours, to target our training to images where the segmentation model performed poorly.

5.2 Segmentation Method

5.2.1 Creating Segmentation Masks from Edge Derived Contours

Due to the consistent camera angle and lack of glare we were able to take many of the processes for data cleaning and contour evaluation developed for handling the data of the first factory visit and improve upon them for better handling of the imagery from subsequent visits. One key benefit was being able to derive the contours of fish by applying an edge detector to the image. As the camera was fixed, known obstacles with a fixed position could be removed.

Edge detection was performed by a canny edge detector. A 5x5 Gaussian filter was first passed over our input image to reduce noise that may produce unwanted edges in our image. A minimum value of 45 and a maximum of 180 were used for hysteresis thresholding. This meant that edge pixels with a value above 180 were considered, pixels between these two values were only considered if they were connected to pixels above the maximum and pixels below 45 were discarded.

These values were chosen to reduce the classification of edges that are not specifically attributed to our object of interest and to capture most of the edges of the fish, an example of the result of this step is show in image (a) of Figure 5.1.

As contours were able to be derived from the images from the second and third factory visits further research was conducted into whether image masks could be generated to expedite the development of training data for the segmentation model. Though the process of automating the generation of mask images from the contours of an edge detected image was possible, it was highly susceptible to any artifacts creating edges near the fish and required a significant amount of cleanup to remove unfavourable masks that did not accurately represent the shape of the fish.

However, it was able to produce accurate masks at a faster rate than drawing polygons by hand.

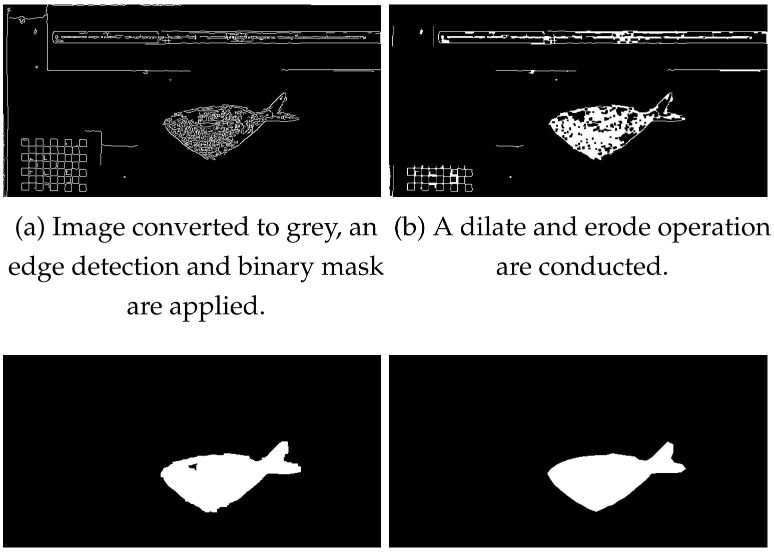
To avoid rough edges and reduce the impact of gaps in the edge detection morphological operations were applied to the binary mask. Two morphological operations were applied to identified contours, a dilate, and an erode to create the binary image masks.

Dilate operations were used to fill in missing sections of the surrounding edges of the target fish. This is done by increasing the boundaries of the white identified contours, reducing the area of the black background. Erode operations were then conducted to reduce the area of the identified fish edges back to a size that was representative of the size of the originally identified object. An image created by these operations is shown in image (b) of Figure 5.1.

From these edges, we identified the contours in the image and kept only those of the object closest to the centre that was greater than 4000 pixels in area allowing us to remove all unwanted polygons. The remaining contour was then filled by setting pixels within the polygon to 255 and the remaining pixels to 0. The result of this process is shown in image (c) of Figure 5.1.

The contour was then passed through a simplify function, which reduced the number of vertices to 50 to be more similar to a human-made polygon, this also helped reduce jagged edges made when identifying contours. A final fill was conducted on these simplified contours creating our image mask as shown in image (d) of Figure 5.1.

This process was conducted 3 times with varying iterations of the dilate and erode operations, as the quality of the resulting mask heavily relied on the edges identified which could vary significantly, even in frames that were chronologically close. Performing many dilates would result in the detail around the tail being lost, it could also then cause nearby edges found from scales or ice to merge into those of the fish resulting in contours that do not closely follow the edges of the fish. Too few dilates and



(a) Image converted to grey, an edge detection and binary mask are applied.

(b) A dilate and erode operation are conducted.

(c) Only one contour is kept and filled.

(d) Exterior coordinates are simplified to 50

Figure 5.1: Process for generating masks from clean images

the missing edges of fish would cause gaps in the mask that our simplify could not amend. As such a thorough final human review was used to select which mask from the three automated datasets best fit each image, images which had no mask that closely followed the shape of the fish were removed.

5.2.2 Annotation

For images in which quality contours could not be created through edge detection, such as for images that failed the shape feature evaluation or those not collected from a fixed camera, the VGG annotation tool [31] was used. The tool was developed by researchers at Oxford and provides a simple and intuitive open-source environment, hosted through a browser, for drawing a variety of annotations such as bounding boxes and polygons.

Polygons were drawn around both the fish in the centre of the frame and in some cases, the fish in buckets were also annotated. Only a single class was used, as polygons were to be used specifically for length there was no need to annotate hands also. Targeted training was used to only annotate images in which the fish had changed position, rotation, or if the view angle had significantly changed.

All polygon annotations were stored as CSV files, with an ID for the image file that was annotated and a list of values which stored all the points of the polygon. These had to be converted into a mask image, a black and white representation of the image with all pixels that were annotated being represented with a white pixel, with a value of 255 and all other pixels being black.

Training data made from images of the first factory visit had a significant shortcoming. As the camera was not fixed for this dataset the location of objects that may adversely affect training was highly variable. The main drawback presented by this approach was the buckets that the fish were being passed between were also placed on the table to allow for quick access when placing each individual on the table. Drawing polygon annotations requires far more points to be drawn than other annotation methods. To draw a bounding box of a fish two points would be required, whereas a polygon would start at around 15 points. Annotating each tarakihi in a frame when there were two buckets visible would have drastically reduced the number of frames that could be annotated. Instead, just the fish that was in the centre of the table was annotated in most images. The second and third factory visits did not have this issue as the camera was fixed and buckets were not kept on the table.

To avoid the negative training that would occur by having non-annotated fish in the buckets present in the training data images with a single annotation were cropped around the annotated polygon. 517 image-mask pairs were created with this process. The remaining images used for training the segmentation model were those created through the automated process, or manually annotated from the data from the second and third visits where multiple fish being in the frame was not an issue.

5.2.3 Synthetic Image Generation

A PyTorch implementation of Stylegans2-ADA developed by Karras et al.(2020) [42] was used to generate synthetic images of tarakihi. GANs generally require large amounts of image data before realistic synthetic images may be developed, $\sim 10^5 - 10^6$ images are required for a high-quality and high-resolution GAN [42]. Smaller datasets often result in the discriminator overfitting, meaning that generated images suffer from a lack of variation [42]. Karras et al. proposed adaptive discriminator augmentation (ADA) in their 2020 work to significantly stabilise training when only limited data was available. Standard methods for reducing overfitting and automating an increase in the quantity of training data for machine learning algorithms in computer vision generally consist of augmenting input data, with techniques such as rotation, flipping or adding noise to the image. Such techniques applied to training data of a generative adversarial network also augment the generated images. Research into ADA sought to automate the level of augmentation relative to the degree of overfitting present in the generated images, without augmentation 'leaking' into the generated images [42]. Automated approaches for quantifying the degree to which the model was overfitting were explored, and an adaptive approach for controlling augmentation was proposed. Traditional methods of measuring whether a model is overfitting to the training data would be to evaluate its performance against a separate validation set relative to the training set, an example of calculating the degree of overfitting with this approach is shown in Equation 5.1. Such an approach requires an additional validation set for measurement. As GANs generally require a large amount of data for training, the requirement of an additional imagery set to reduce overfitting results in a high entry barrier for the practical application of these models.

StyleGANs-ADA literature [42] found that when the GANS model began to overfit the validation sets' behaviour was increasingly similar to that of the generated images. Additionally, the non-saturating loss [34], as used in StyleGANs2 [44], diverges symmetrically around zero for both the real and generated images as the extent of overfitting worsens. This was leveraged to signal overfitting without the use of an additional validation set.

$$r_v = \frac{E[D_{train}] - E[D_{validation}]}{E[D_{train}] - E[D_{generated}]} \tag{5.1}$$

$$r_t = E[sign(D_{train})] \tag{5.2}$$

Equations 5.2 and 5.1 show two overfitting heuristics where discriminator output for the train, validation and generated images is denoted as D_{train}, $D_{validation}$, $D_{generated}$. Their mean over N minibatches is calculated written above as E. Left; shows a traditional approach using a separate validation set. Right; shows an approach that requires no additional dataset. For both methods, there is no overfitting when r=0 and peak overfitting when r=1. The probability of augmentation (p) is initialised to 0 and adjusted every four mini batches based on this value.

This approach for adaptive discriminator augmentation paired with the use of transfer learning was able to generate high-quality (1024^2) images of painted faces with only 1336 images from the METFACES dataset. Training from scratch also yielded significant improvements with $\sim 5k$ images from an animal faces dataset [26], convincing images were generated of cats, dogs and wildlife at a high-quality resolution (512^2).

Despite making use of ADA, which allowed us to reduce the amount of input data required to generate images of fish, a variety of positions and visually different fish would enable a more diverse set of generated images, improving the performance of a segmentation model trained on this data. In order to train a stylegans model a large number of images first had to be generated. The first approach used object detections to target regions of an image to crop from the first factory visit. There were two issues with this approach. The first was that there was a high degree of variation in this imagery, buckets of fish that were visible in the image were also captured by this process. The second issue was that no automated approach for evaluating the quality of object detections existed therefore false positives and inferences of fish in buckets were included in the data created. Despite these shortcomings, this data was used to train a stylegans model and the results were promising, synthetic fish were created. However, the input data required a significant change to produce images that may be used to train a segmentation model.

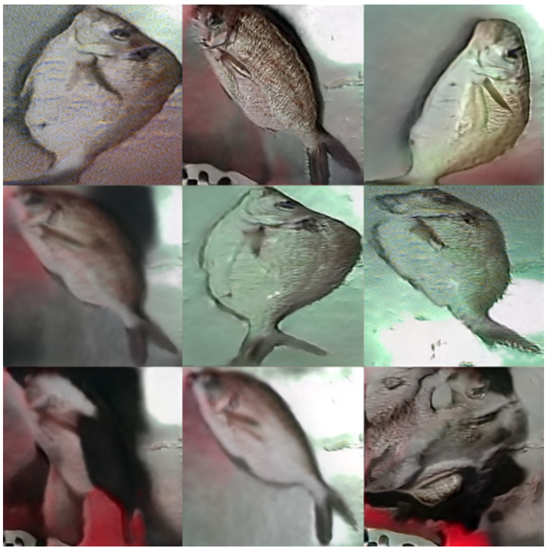


Figure 5.2: Results from Stylegans model trained on object detections

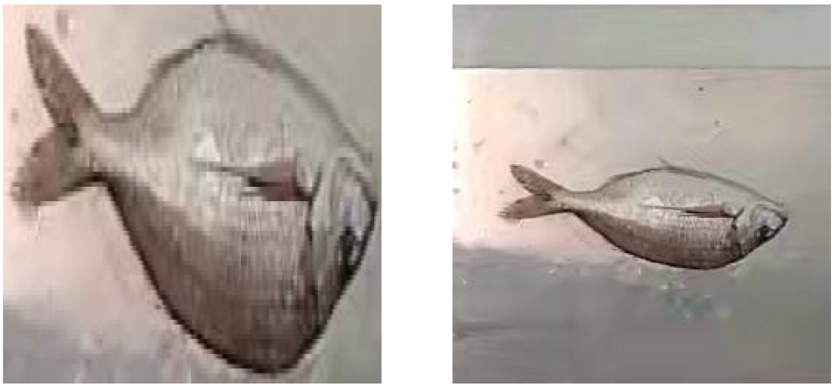
In order to reduce the amount of data cleaning that would be required for imagery from the free camera system, we looked toward the data from the fixed camera dataset. By using the automated process for generating mask contours, described in subsection 5.2.1, we were able to generate masks for 61327 images. These masks were then evaluated using the contour features, described in subsection 5.3.2. By using the mask created from a simple edge detection rather than our model's inference, we allowed a greater number of poor-quality masks to be created, as the edge detection filter would pick up the edges of any object in the scene.

The end result of this approach was that fewer instances would later need to be manually cleaned during the review. Images in which hands were over or touching the fish would result in a mask that included both these items. As this mask would not have contours similar to those of a fish this would then fail our contour evaluation process. The use of a segmentation model would result in a greater number of images passing our contour evaluation in which hands were close to or partially covering fish. Though this is beneficial for estimating length, such examples would add unnecessary complexity to our synthetic images at this stage.

After the contour evaluation, 16284 images remained. These images were cropped using the masks that were generated from the contours of an edge detection. A new mask was generated using the contours of the object closest to the centre of the image pixel values within the contour were set to 255 while those beyond were set to zero. An enclosing rectangle was generated by using the extreme values of the white pixels, this enclosing rectangle was then used to determine the pixel coordinates for the region of the image to crop.

The first implementation of this system cropped images to the region of the enclosing rectangle and then resized images to all be 256^2, an example of this is shown in the image on the left in Figure 5.3. Such an approach did not consider preserving the aspect ratio of the original images. A StyleGANs model trained on such data would generate images of fish with proportions unrepresentative of real fish and may have adverse effects when using the data to train a segmentation model.

The second implementation instead used the enclosing rectangle as a guide for where to take a 256×256 crop of the image. The width and height of the enclosing rectangle were calculated and used to measure the distance from the target crop size. The minimum and maximum pixel



(a) Image cropped to an enclosing rectangle of the contour

(b) Image cropped to a padded enclosed rectangle of the contour

Figure 5.3: comparison of two different approaches for cropping images

coordinates on each axis were then decreased and increased respectively. The quantity these were decreased or increased by was half of the difference between the width or height and the target crop size on the relevant axis, this approach focused on capturing as much of the segmented fish in the cropped region as possible. If the enclosed region was larger than the crop size of 256, on either axis, then the crop region would begin at the minimum value for the enclosed region on this axis and extend from this point to the limit of the crop size on this axis, $minimum + 256$. Enclosed regions that were below the crop size but were close to the edge of the image, so that a 256^2 crop would extend beyond the limits of the image instead had the corresponding pixel coordinate set to this edge of the image.

For example a enclosing rectangle with pixel coordinates $((38, 50), (340, 150)$ would result in a crop region at points $((38, 0), (294, 256))$. An example of an image created by this process is shown in the image on the right in Figure 5.3. This method not only allowed us to preserve the aspect ratio of the fish but also to capture more detail from the scene around the fish, such as scales on the table or hands placed on the ruler. Though these may be unnecessary complexities when simply wanting to generate synthetic images of fish, including this information resulted in a more realistic synthetic imagery set for training the segmentation model and would reduce the number of false positive inferences for such items.

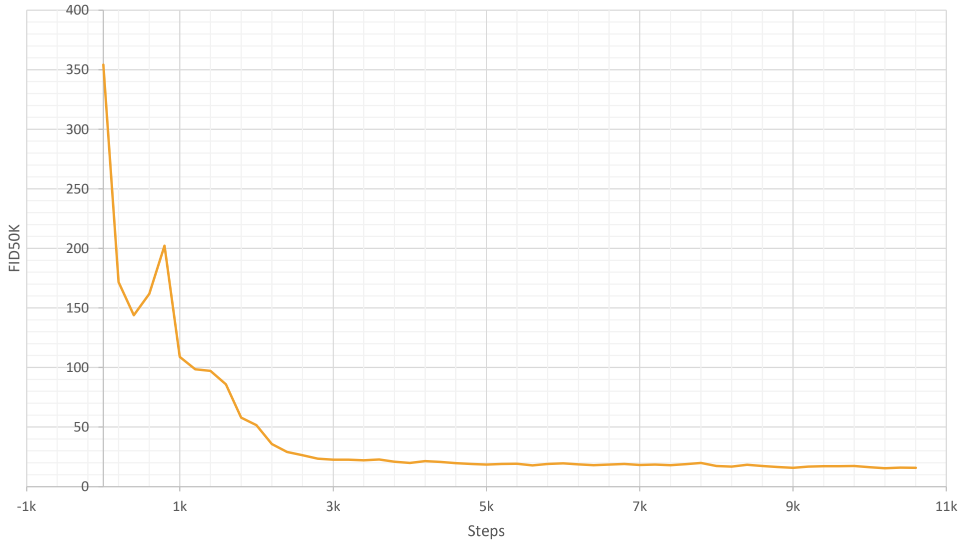


Figure 5.4: Fréchet inception distance 50k

From the created dataset of 16284 cropped images, 5853 images were removed. Images were removed if hands were covering any part of the fish or if they were more than a fifth of the way into the image. As noisy or rotated images were not a concern The resulting dataset was augmented to further increase its size. Images were flipped, rotated and noise and blur were added until the total dataset consisted of 73017 images. These images were then used to train a segmentation model.

A PyTorch implementation of StyleGAN2-ADA [42] was used for generating our synthetic images, the official PyTorch implementation was chosen for its faster training times. A Docker image from the official implementation was converted to a Singularity environment to perform training on a single A100 GPU for a total of 46 hours with a batch size of 16. Fréchet inception distance 50k (FID50K) was used as our chosen metric and the final model achieved an FID50K of 15.49 an example of images generated by this model is shown in Figure 5.5.

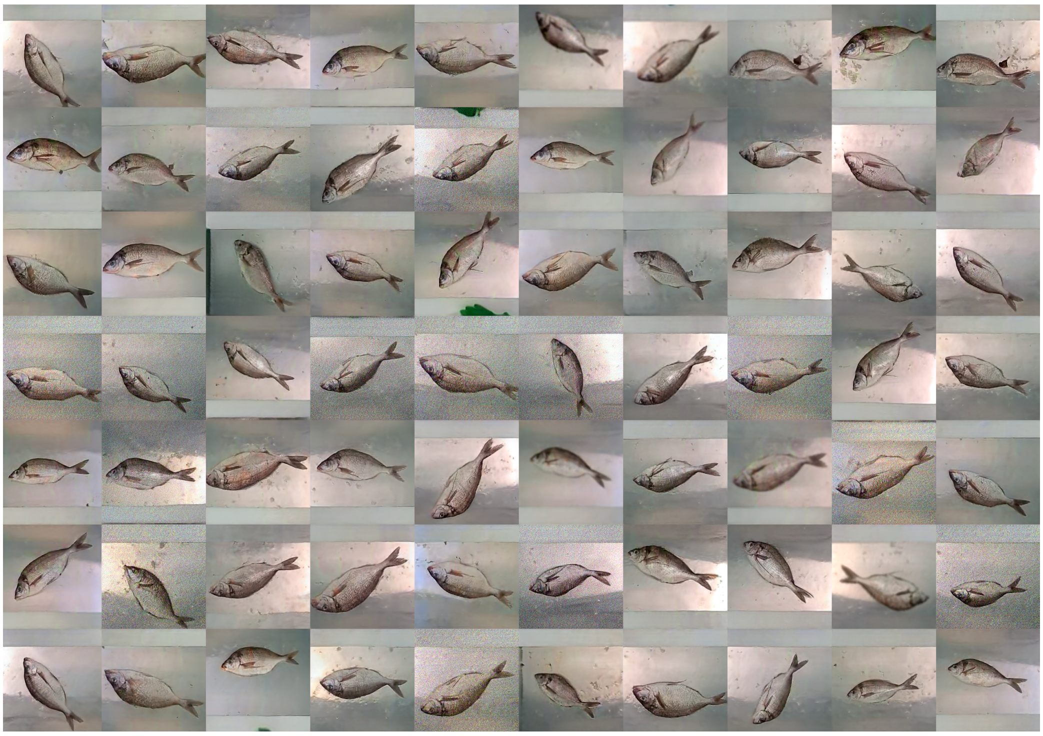


Figure 5.5: Synthetic images generated by the StyleGANs2-ADA model

5.3 Automating the Data Cleaning Pipeline

5.3.1 Object Detection for Removing Unfavourable Frames

As the datasets gathered in this research were all still frames extracted from raw video footage of fish being measured and passed over a table a large quantity of the data had little or no information relevant to identifying or measuring the length of fish, such as those frames in which fish were being moved over the table and the view of the fish was obstructed by hands. There were also many frames where no fish was present an example of which would be when researchers were getting set up with the next bucket of fish. The first method to attempt to remove images in which fish were obscured or not present, in the initial factory visit imagery, was by using an object detector to identify fish and hands, when hand detections did not overlap the bounding box of a fish then we considered this a good image that may provide valuable information when training a segmentation model. As this was an exploratory method to evaluate the performance of an object detector for sorting the large amounts of raw data a relatively small training set was used to gauge the potential. 299 bounding boxes were created and used to train an EfficientDet model. This method was prone to over-predicting bounding boxes for the class 'fish' and proved to be a poor approach to sorting the large dataset.

5.3.2 Interpretable Methods for Evaluating Contours

Introduction

The second approach that was explored on data from the first factory visit was sorting images based on the quality of an inferred segmentation mask. This approach required the entire dataset to be inferred on by the weak segmentation model, which was trained on only 153 images. The resulting inference masks generated from this were then used to signal whether the original image had a clearly visible fish, an obstructed fish, or no fish at all. This approach was far less prone to false positives than the initial approach using an object detector.

To evaluate the quality of inferences all contours were retrieved from the inference masks. OpenCV's findContours function [7] was used to identify these and describes contours simply as a curve joining all continuous points along a boundary with the same colour or intensity.

The target contour in the image was found by comparing the areas and distance from the centre of each contour. The contours with an area between 4,000 and 60,000 pixels were considered and the one closest to the centre was selected as the contour most likely to be the fish currently being captured. This contour was then further analysed to predict its quality. This process was intended to reduce the manual sorting requirement for this large dataset, an automated procedure was developed to sort images based on the quality of their inference. This was done as a proxy for entropy-based sampling, this approach approximates a sampling method for images which will provide the greatest information gain to our model with further training.

A separate mask was generated containing only the inference in the centre of the image. Various features were calculated from the contour found from the remaining inference and were analysed using interpretable machine learning techniques developed in research conducted by Nori et al. [62].

Contour Datasets

Two datasets were created for evaluating this system, one for high-quality segmentation masks, and another for low-quality segmentation masks. Segmentation masks were deemed high-quality if they were of a standard suitable for training, this meant that the entire fish would need to be included in the mask and minimal false positive pixels would need to be present.

The low-quality segmentation mask class was a broader category. Masks unsuitable for training were included, as well as masks from false positive inferences or masks generated from multiple fish overlapping that did not represent the shape of an individual fish. This allowed us to focus our exploration on which features the contour of a high-quality segmentation comprises.

The algorithm was then trained on various features calculated from these two datasets. Some features were raw inputs derived from OpenCV's contours [7], others were calculated by using geometric statistics common in particle, and remote sensing. Input features were gradually reduced until there were only 3, keeping the system simple and reducing the degree of overfitting to the training data. Fewer inputs meant that inferences were always justified based on a few input features and were, therefore, more easily interpreted.

InterpretML

InterpretML [62] was chosen as it provides insight into both global and local features. Research into interpretable machine learning has been gaining traction in recent years [56], interpretable methods allow for greater trust in the systems in which they are integrated, and decisions are less likely to be challenged if their justification is clear. By using interpretable machine learning techniques we can not only develop robust automated systems to distinguish between good and bad polygons but we can also explain why these decisions were reached, and which factors contributed to the final result.

As part of InterpretML research, a new algorithm was developed with a focus on interpretability, the Explainable Boosting Machine (EBM). EBM is a generalised additive model (GAM) and uses modern machine learning techniques, one of which is gradient boosting, to learn feature functions. Care is given to restricting the boosting procedure to a single feature at a time, learning from features is done iteratively, so feature contributions will be diminished if there are other features that are very similar. However, a very low learning rate is used and as a result feature order does not matter.

$$g(E(y)) = \beta_0 + \sum f_j(x_j) \tag{5.3}$$

Where g adapts the GAM to either classification or regression, via a link function. f represents some function explaining the relationship between the variable j and the target class, it is denoted as:

$$f(x) = \sum_{k=1}^{k} \beta_k b_k(x) \tag{5.4}$$

The equation for GAMs is similar to that of logistic regression. Where it differs is in the explanation of relationships between features and the target class. GAMs do not require a linear relationship but instead support arbitrarily complex functions [48]. EBMs go one step further than this and detect pairwise interactions when evaluating features. These are automatically included in the model, if found to be significant, and are displayed in the feature importance summary (example shown in Figure 5.6. The inclusion of these pairwise interactions in the GAM architecture was first explored by Lou et al. [49] and is found to increase the accuracy of the models in [62] without sacrificing intelligibility. This form of GAM is called GA2M [49] and is represented by Equation 5.3.2.

$$g(E(y)) = \beta_0 + \sum f_j(x_j) + \sum f_j(x_i, x_j) \tag{5.5}$$

The effect of the input, at its various values, in predicting the likelihood of the positive class is more accurately represented by using this equation when a non-linear relationship exists between the input variable and the target class.

As EBM is an additive model feature scores are easily interpreted due to their modular contribution to the prediction[62]. The contribution of each feature may be visualised by plotting $f(x)$ from the Equation 5.3.2. This non-linear relationship is described as shape functions in the research conducted by Lou et al. [48].

Global scale interpretability provides the importance of each feature, this is calculated by using the mean absolute score, allowing us to ascertain which features are most important in determining whether a contour is highly likely to be that of a fish or not.

On the local scale each input feature has a score to determine its impact on the predicted class, this score can be both positive; increasing the likelihood that the observed data belongs to the positive class, or negative; decreasing this likelihood.

Shape Factors

Several features were calculated from the contours found in the segmented image, in the form of shape factors, to better evaluate their shape. Shape factors are used for analysis in a variety of fields such as image analysis and microscopy, where the shapes of objects must be differentiated. The notable geometric properties of fish contours that were explored included circularity, compactness and complexity. Podczeck (1996) [67] explored the use of shape factors to assess the shapes of particles. As part of this assessment 3 of the considered parameters for describing shapes analysed the deviations from standard shapes such as a square, triangle and circle. The deviation from a circular image was identified as a potentially useful factor and was included in our analysis of viable features.

Deviation from a circular image [67].

$$\frac{area}{\frac{\pi}{4}s_m^2} \tag{5.6}$$

Where s_m is the longest side of a minimum enclosing rectangle.

Compactness is a measure of the simplicity of the boundary of a shape [11], equidistant vertices from the geometric centre of the shape are an indicator that a shape is very compact, and a circle is the most compact shape. Two measures of compactness were considered:

the Richarson compactness ratio [71].

$$2 * \sqrt{\frac{\pi * area}{perimeter}} \tag{5.7}$$

Reock compactness ratio [69].

$$\frac{area}{\pi * radius^2} \tag{5.8}$$

A popular measure for complexity is the Schwartzberg score, often used in geographical information systems [52], which calculates the ratio of the perimeter of a shape to the circumference of a circle with an equal area.

Schwartzberg complexity [76].

$$\frac{perimeter}{2 * \sqrt{\pi * area}} \tag{5.9}$$

Finally, two additional features were calculated using the characteristics of the external border of our shape and the pixel length of the contour, calculated from a minimum enclosing circle, as a scale factor. These were: Podczeck's elongation calculation [67]

$$\frac{perimeter}{length} \tag{5.10}$$

The second equation was an adaption of Podczeck's elongation, instead exploring the number of vertices relative to the length of the shape, the name coordinate complexity is used to refer to this calculation and is calculated as.

$$\frac{ext.coords}{length} \tag{5.11}$$

Implementation

The data used in training the explainable boosting machine had to be carefully selected using a mixture of both inferred and manually created segmentation masks. 160 of the masks included in the positive class were

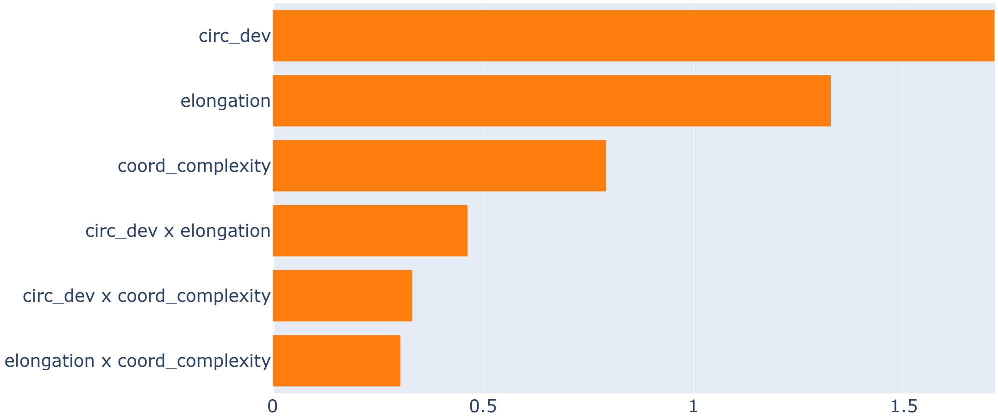


Figure 5.6: feature importance from three inputs for the final GAM trained on 1100 contours. Circle deviation was the most important feature, followed by elongation and finally coordinate complexity.

manually drawn, and the remaining masks were inferred. In earlier iterations, human-made masked images made up roughly 50% of the positive class. A larger number of masked images that were created from human annotated polygons were included in the training data to ensure quality contours were being used. However, this resulted in inflated feature importance being given to the number of exterior coordinates, due to clear differences between the two methods of creating masked images. The first run of the EBM used a total of 35 features. A larger number of features than desired was used in early implementations, this was done to evaluate feature relationships and analyse the input features. By including many features and identifying which are best at explaining the contours of a fish.

Additional features included:

1. Aspect Ratio
2. Perimeter relative to the area

3. Area
4. Perimeter
5. number of vertices
6. All 24 moments [7]

After running interpret ML [62] with many input variables, those with a considerable contribution in identifying the target class were kept and other variables were removed. As interpretML iteratively calculates feature contribution scores; variables that are highly correlated will subsequently have a diminished score. For groups of highly correlated inputs, only a single variable was kept. This process was repeated until only 3 features remained; circle deviation, elongation and coordinate complexity. The feature importance of a model trained on only these 3 features is illustrated in Figure 5.6. Several pairwise interactions are also identified, one of which, circle deviation and elongation are shown in Figure 5.7.

5.4 Results and Analysis

5.4.1 EBM Shape Function Interpretation

This subsection showcases the interpretability of the EBM by illustrating the shape functions learnt during training on the shape features (Circle deviation, elongation and coordinate complexity). The graphs shown here are a direct visualisation of the contribution to both the positive class (high-quality contours) and negative class (low-quality contours) within the range of observed values present when training the EBM. A contribution score greater than zero means that the relevant shape feature value was found to be associated with a high-quality contour, while a shape feature value with a contribution score of less than zero is associated with a low-quality contour.

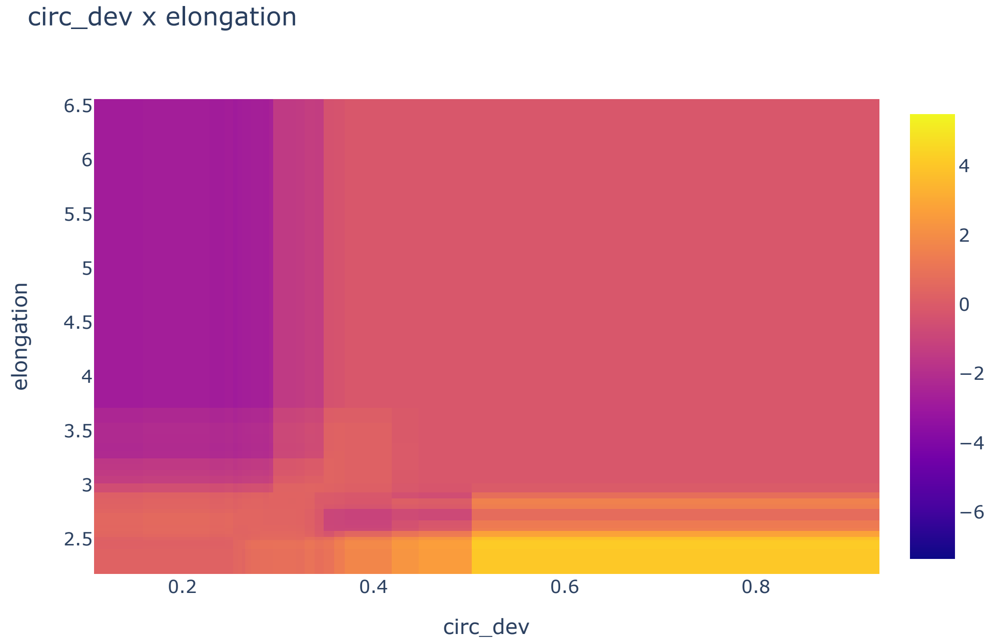


Figure 5.7: Graph showing the pairwise relationship between elongation and circle deviation. A colour map for contributing scores is shown on the right

From the visualisation of the pairwise interaction between circle deviation and elongation, it is shown that Objects with an elongation value greater than three and a circle deviation below 0.3 have a negative score. This suggests a contour with a circle deviation and elongation in this range is likely one that is not representative of a fish. A low elongation and low circle deviation are associated with a slight positive score, contributing toward the model deciding the features are those of a good fish contour. Finally, an elongation with a value below 2.5 and a circle deviation above five was found to be a strong indicator of a good fish contour, with a score greater than four. This is an important distinction to make, as the circle deviation shape function (Figure 5.8) alone, shows a negative response at

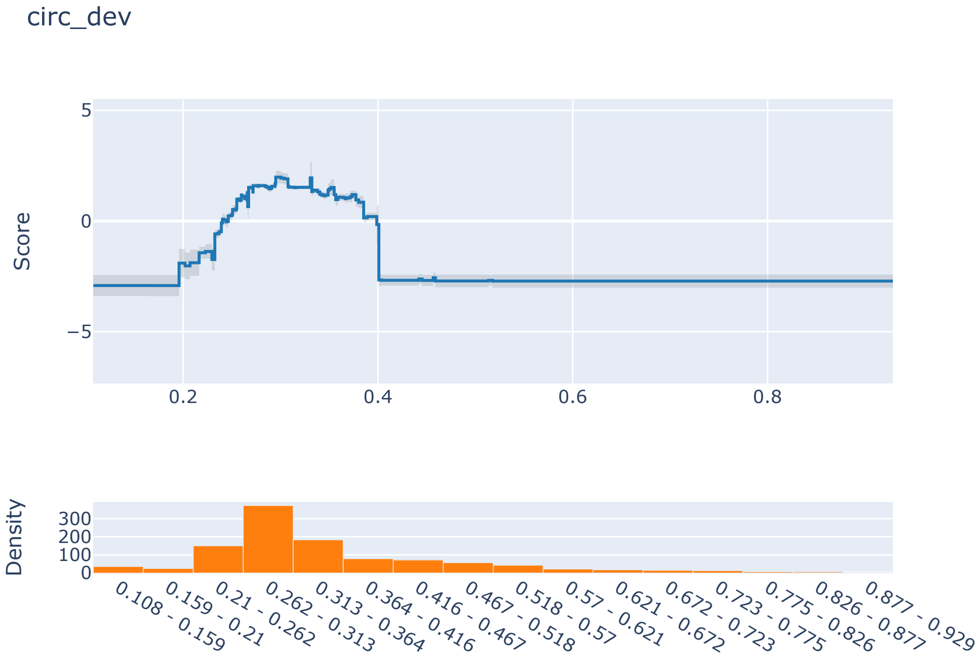


Figure 5.8: f(x) from Equation 5.3.2 plotted for circle deviation

higher circle deviation values. By bringing the instances that are more difficult to distinguish into a score function with greater dimensions we are able to better explain their contributions and improve the model's accuracy.

Circle deviation ranges between zero and one, values close to one signify our contour is highly similar to a circle whereas values close to zero signify a very dissimilar shape. The circle deviation graph, Figure 5.8, shows the contribution scores found from the training set of calculated contour features. The varying values for the deviation from the shape of a circle are shown on the X-axis, while the contribution is shown on the y. Contours with a very high circle deviation, with values less than 0.24, are commonly found for contours that are not of a quality fish segmentation. Very circular contours, with values closer to 1, had a similar score of -2.41.

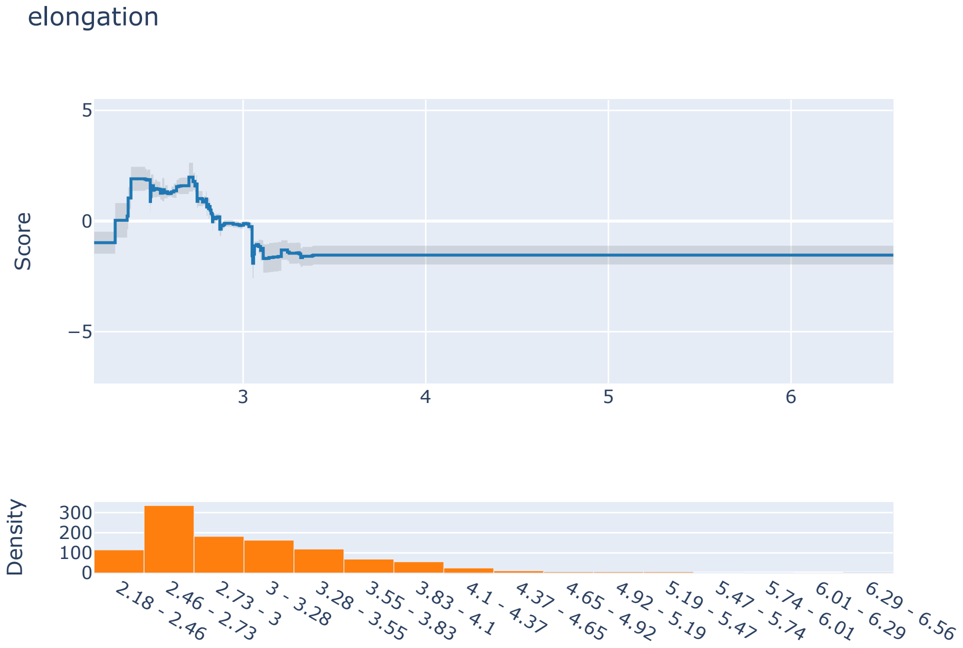


Figure 5.9: f(x) from Equation 5.3.2 plotted for the elongation

Circular deviation scores between 0.24 and 0.4 had a positive contribution to the likelihood of our shape being that of a quality fish segmentation.

The elongation shape function also shows that lower values are attributed to a good contour of a fish. Tarakihi contours had an average elongation of 2.70 while contours of poor inferences had an average of 3.36 meaning a greater perimeter relative to the length. This was caused in some instances by a rougher border of the segmented object. An example of this is shown in Figure 5.10 where the image on the left, a mask created from a hand-drawn polygon, has an elongation value of 2.88 and the image on the right with a far rougher surface has a value of 6.56.

A density plot below the elongation shape function in Figure 5.9 shows

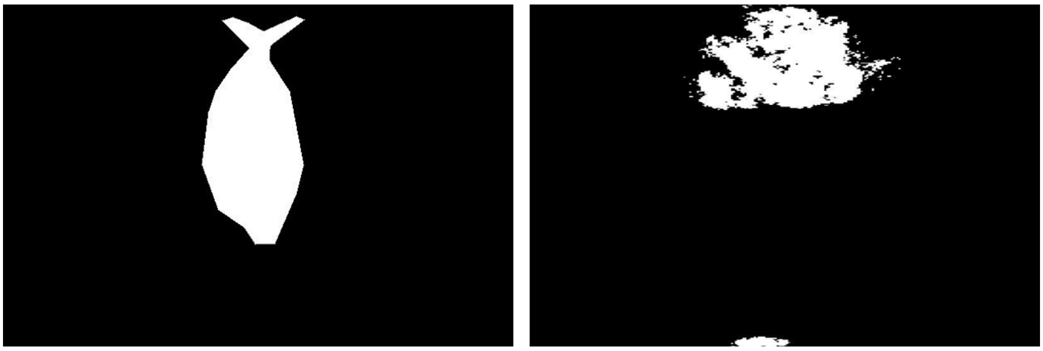


Figure 5.10: Comparison of a mask from a human-drawn polygon to that of an inference of a bucket of fish

the number of observed instances in the range of values. For example, for elongation values in the range $2.46 - -2.73$ there were 335 observations, while contribution scores in the range $4.37 - 4.65$ were based on only 11 observations. This allowed for a better understanding of how reliable contribution scores calculated for these ranges were. Contribution scores displayed at high elongation values (> 4.1) and high circle deviation values (> 0.518) are based on fewer than 25 observations and care must be taken when making decisions based on this information.

Figure 5.11 shows a local explanation of our interpretable model on a single instance. This illustrates the shape features values (Y axis) and their contribution scores (X axis), calculated from a segmentation inference, that resulted in the prediction that the observed contour was that of a high-quality fish.

The high negative intercept means that we assume the observed contour is not a fish unless the shape features are similar to those of a fish. In this instance we see that the circle deviation (0.31) is within the range we would expect of a quality fish contour, as is the elongation (3.41). The paired relationship of circle deviation and coordinate complexity resulted in a negative contribution score however this is not significant enough to suggest our contour is not that of a fish and so the predicted class is a

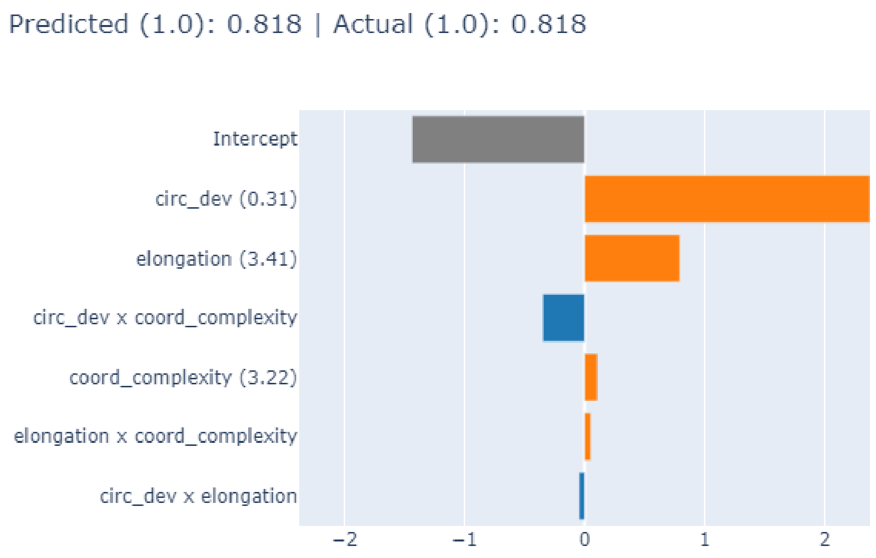


Figure 5.11: Visualisation of the contribution scores for each shape feature for a single inference of the interpretable model. Negative shape features are coloured in blue, positive contributions are orange and the intercept is grey.

high-quality fish contour with a prediction score of 0.818. By using these interpretable techniques, we may demonstrate why certain contours are used in estimating the lengths of observed fish and why others are discarded. As keeping undersized catch may result in a fine, being able to justify why a length was or was not recorded provides a benefit to the stakeholders, in the form of an accountable system that may be tailored to suit their needs. Should new species of fish wish to be added in the future, the explainable classifier may be trained on the contours of these new fish.

5.4.2 Identifying Poor Segmentation Inferences

A second check was created, based on findings from the shape functions, for contours that did not pass the EBM. A contour that failed the first check but still had a $0.2 < circdev < 0.41$ was considered to potentially have a clearly visible tarakihi. This range of circle deviation values was chosen as the contribution scores were still variable, particularly on the lower end of circle deviation values. As this is a classification problem, if all values within this range were of the negative class we would expect to see a consistently low score, as is observed at values greater than 0.4. There was also a large number of observations in this range, so the inference that contours within this range are likely to be those of a fish is based on many examples. Circle deviation was also the most important feature in identifying quality fish contours and should therefore be a good identifier.

Any contour that did not meet these requirements was assumed to not contain a clearly visible fish. This approach meant that an extra neural network for sorting the data was not needed. We could train a segmentation model to infer on the large dataset and iterate improvements on the same model which would also be used for finding the pixel length of inferred fish.

A simple test was conducted on this process, placing images from a test set of 948 images into folders relative to their true and predicted class, Figure 5.12 shows the confusion matrix from this test. Of the 446 images that were predicted as the negative class 94 had a circle deviation between 0.2 and 0.41, these were placed in a separate directory for further investigation. 15 of these images were the false negatives illustrated in the confusion matrix, meaning only 1 false negative did not pass this check. Of the 79 remaining images, 4 were of buckets of fish and 27 were false positive inferences, the remaining 48 images were of fish.

The use of ML models' inferences to sort the data to be used for future training may reduce the performance of the model when just using entropy-based approaches as it may lead to further encouraging model biases if not carefully monitored. If we only consider what items a model is unsure about we would miss the cases in which the model is confidently incorrect, as such, negative predictions were subject to scrupulous manual correction.

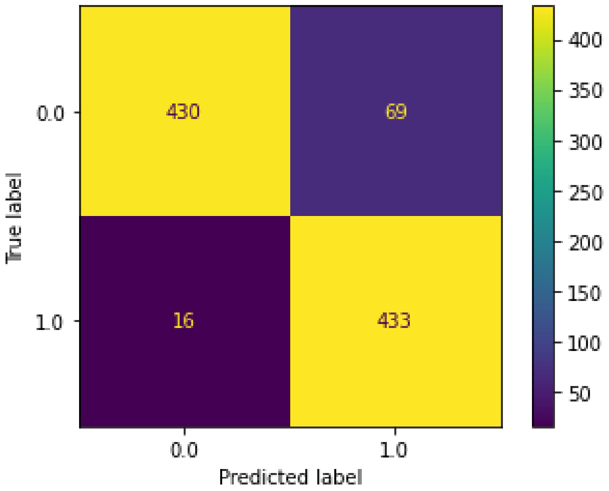


Figure 5.12: Results of the contour evaluation model on a test set of 948 images

The performance of the final model with reduced features is illustrated by the confusion matrix in Figure 5.12. A focus was given to high recall values for the positive class (quality fish contour) in preference of precision. Though the purpose of this method was to reduce the number of false positives affecting length estimations, minimizing the likelihood of not considering contours that would help when calculating the true length of an observed individual was also a priority. Allowing some false positive to pass was less important than falsely predicting a quality fish contour. As such the precision of this model was 0.863 while the recall stood at 0.964.

5.5 Chapter Summary

Our first goal of this chapter was achieved by discussing our two approaches to creating segmentation masks. The first used the contours that resulted from applying an edge detection on images of fish that were not touching any other objects such as the ruler or hands. This approach was successful in creating 1154 masks (dataset 2) for images from the second factory visit. The second approach was used in difficult instances where the automated approach for creating masks was insufficient. This used the VGG annotation tool [31] to create 517 masks from the first visit.

This chapter presented our approach for generating synthetic images. This began by first creating a large dataset of image masks, derived from an edge detection. The contours found from these masks were then evaluated, using an EBM, to determine whether they were likely to be those of a fish. The images in which there was a mask that passed this check were then cropped, based on the size and location of the mask. A PyTorchimplementation of stylegans with adaptive discriminator augmentation was trained and used to generate 700 synthetic images of fish which were also manually annotated, achieving our second goal.

Finally, we demonstrated the use of interpretable techniques for providing greater insight into shape features, that were derived from contours, and how they may be applied to remove inferences that are of low quality or those likely to be false positives. We also explain that this process provides the various stakeholders, such as fisheries organisations or scientists, with an accountable system that may justify why contours were, or were not, used for calculating the length. The shape functions created by the explainable boosting machines allowed us to identify a target range of values for circle deviation, one of the final three shape features included in the model, which was commonly associated with high-quality fish contours. Contours that had a circle deviation within this target range, but were not identified to be good contours by the model, were then flagged as contours that may be of fish but just a poor mask. This allowed us to target our human conducted annotations on the images in which the model was performing poorly.

Chapter 6

Length Prediction

So far we have presented our methods for collecting data and creating segmentation masks. The process for deriving the contours from these masks was discussed and an approach to only keep those of high-quality fish was presented.

Providing accurate lengths is the next stage of our research. We hope that by providing millimeter lengths for all partitioned fish that are presented below a camera, more informed decisions may be made regarding not only the quantity of undersized fish being caught but also the overall catch. Such a system brings benefits for scientists monitoring the catch size and count estimates, by providing more accurate length datasets with a number of samples far greater than current methods are capable of producing [16]. Commercial fishing vessel captains also benefit from these statistics by having a better idea of how close they are to meeting their quota and we hope to better avoid areas where there are large quantities of undersized fish.

6.1 Goals

A major goal of this research was to identify the millimeter lengths of fish from images, and this chapter discusses the methods for ascertaining this by using both a segmentation inference to extract the pixel length of our fish, in subsection 6.2.1), and the use of a calibration pattern for calculating the scale factor to translate pixel lengths to millimeters, in subsection 6.2.2. To achieve this goal, we will investigate the following topics:

- an approach of extracting pixel length from the contours of a segmentation mask.
- how to calculate the scale factor of the image by using a calibration pattern.
- how to gather accurate lengths from a video with multiple frames of each fish.
- how well models trained on the same number of images from different datasets may predict lengths of fish from videos.
- how to adjust final length estimations to account for unknown error in predicted lengths.

6.2 Methods

In order to estimate millimeter lengths, two distinct values had to first be calculated. These were the pixel length and a scale factor that relates the pixel size to millimeters. The pixel length is extracted by calculating the diameter of a minimum enclosing circle of the pixels inferred as the positive class. The scale factor was calculated by relating the projected checkerboard tile size to the known size of 23 millimeters.

6.2.1 Pixel Lengths from Contours

Extracting contours from an inference mask image allowed us to cluster pixels classified by the segmentation model into larger objects and find the extent of an inferred fish. This clustering of inferred pixels is done by following a continuous line around the edge of the inferred mask [7], which is represented by pixels with a value ranging from 0 to 255, pixel values are calculated by multiplying inferred confidences, ranging from 0 to 1, per pixel by 255.

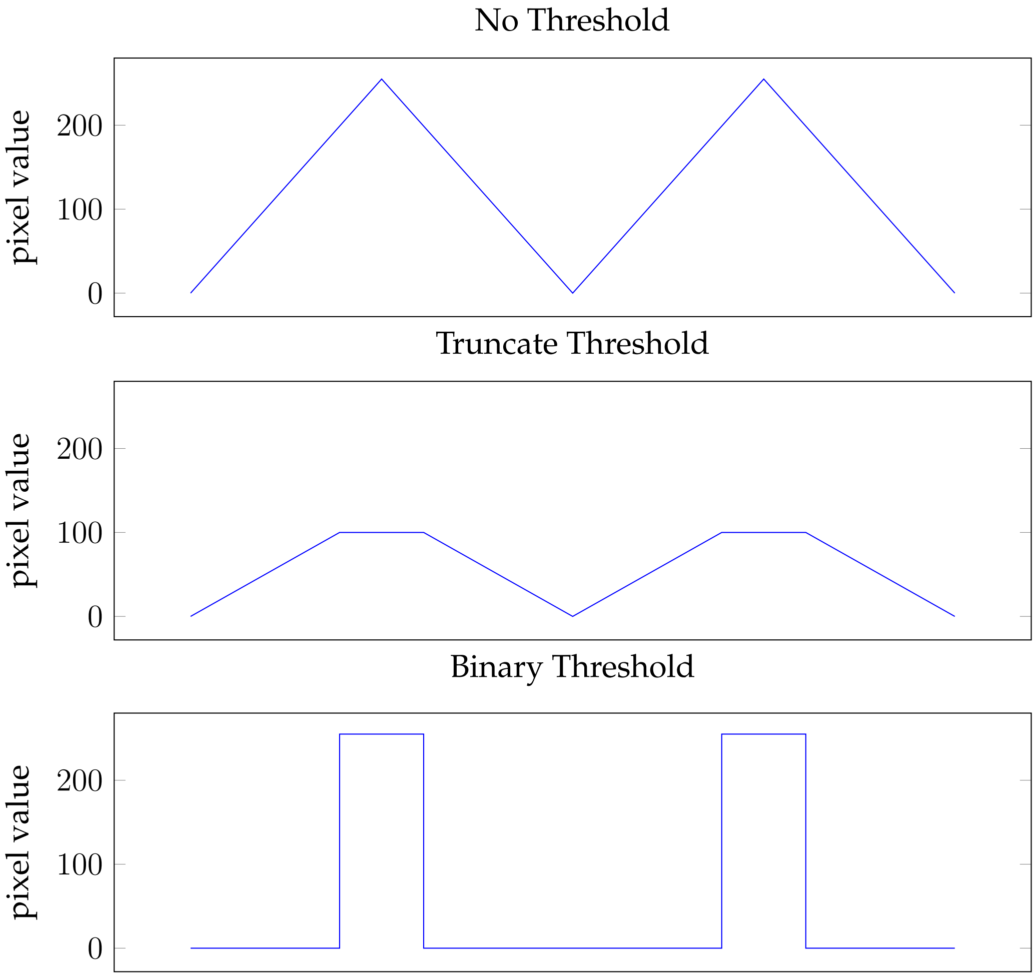


Figure 6.1: Visualisation of how pixel values are altered by different thresholding techniques

A truncate threshold [9] was initially applied to the masked image, this sets the value of all pixels that are above a threshold of 100 to the threshold

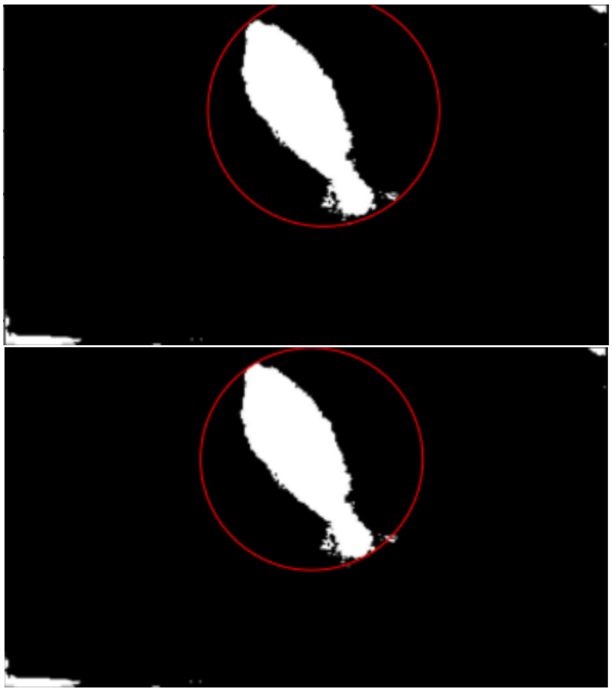


Figure 6.2: Comparison of how a minimum bounding circle is altered when using a truncate threshold (top) and a binary threshold (bottom).

value, while other values remain the same. However, this approach led to artifacts near fish segmentation being included when identifying their contours, and pixels of a low value that connected one segmented object to another resulted in the second object being included in the contours of the original target. To reduce the likelihood of including unwanted objects in our target contour a second thresholding approach was explored, a binary threshold[9]. The binary threshold instead sets all values at or above the threshold to 255 and all other values to 0. This new thresholding method helped remove these artifacts being considered as part of the segmented object's contours.

Figure 6.2 shows the difference between the two approaches for thresholding. The diameter of a minimum enclosing circle is used to calculate the length of our target contour, this minimum enclosing circle is visualised in the example to show how recorded lengths may be affected by poor thresholding. The enclosing circle drawn on the image with the truncate threshold is stretched to include the nearby artifact, resulting in a less accurate pixel length measurement resulting from the skewed circles' diameter.

The moments of each contour were calculated, from which the centroid points were found and spatial moments were used to compare the position of each contour in the image.
Spatial moments were calculated with the following equation [7]:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \tag{6.1}$$

where x and y represent the pixel index of the row and column. I is the pixel intensity at given coordinate x,y.

The area of the contour was calculated as the 0^{th} order moment with:

$$M_{00} = \sum_x \sum_y I(x, y) \tag{6.2}$$

By using a binary representation of the inference mask the result of this equation is a sum of the pixels with non-zero values.

The centre of the contour was calculated as the sum of the non-zero pixels on the X-axis (M_{10}) divided by the total number of pixels (M_{00}). The same is done for the y axis with (M_{01}):

$$(x, y) = (\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}) \tag{6.3}$$

OpenCV was then used to create a minimum bounding circle around the contour closest to the centre of the image, that had an area greater than 4000 pixels and had passed our contour evaluation process described in section 5.3.2. This approach allowed us to select the fish closest to the centre of the image if there was more than one fish visible in the image.

If no contour of this size was within 210 pixels of the center of the image we assume there was no fish visible in the frame. This distance was measured as the euclidean distance between the center of the contour and the centre of the image.

6.2.2 Calculating the Scale Factor

To calculate the pixel to millimetre ratio the known size of an object had to be compared to its size in the image. The internal corners of the checkerboard pattern, if visible in the frame, was first located by converting the image to greyscale, a binary threshold is applied and dilate operations are performed to separate connected black tiles. The contours of the black tiles are extracted and only quadrangles are kept. From these, the internal corner that is positioned optimally between the white and black tiles is extracted, this is a single keypoint from the image.

This was an adaptive process to account for checkerboard patterns (x, y) that were only partially visible, starting with a checkerboard size of 8×6 and decreasing the search parameters for the checkerboard that may be present in an image down to a size of 3×3. If a pattern was not found by this stage the pixel to millimeter ratio was not calculated, instead, the average pixel to millimeter ratio from the dataset was used. The pixel distances of all 4 sides and the millimetre distances were calculated based on the size of the grid that was found in the image (x, y). Estimates for the number of millimetres per pixel were calculated by getting the average of each side's length divided by its pixel length. This value was multiplied by the pixel length of the fish to give its length in millimetres. The pixel to millimetre ratio for a single side of the checkerboard pattern was calculated with the following equation:

$$R = \frac{L_{mm}}{\sum_i N_{pix}} \tag{6.4}$$

The length in millimetres is denoted by L_{mm} and $\sum_i N_{pix}$ is the sum of the pixel distance for the given side (i).

The length calculation for an individual fish from the diameter of the minimum enclosing circle in pixels D_{pix} using the pixel to millimetre ratio from each of the 4 sides $\sum_i R$ of the checkerboard pattern is calculated as follows:

$$L = D_{pix} * \frac{\sum_i R}{4} \tag{6.5}$$

6.2.3 Comparing Measured Lengths to Lengths from Images

To gauge how similar lengths derived from images were to those measured on-site, a preliminary step was conducted prior to the application of machine learning inferences. The two methods applied used a human-supervised system to generate lengths from an image. These lengths were then compared to lengths gathered on-site to determine whether the use of a calibration pattern was sufficient in estimating length.

Images were first paired with the individual fish that was measured on-site. As recorded lengths were ordered chronologically, extracted frames from the videos could be matched up to each measured length by manually grouping all frames of a single fish into a directory. Fish were measured before being placed on the table a gap in visible fish followed by the fish being measured was almost always the sign that a new fish was being observed, notes were taken at the time of measurement to record when unusual activity had happened to assist in distinguishing individuals when later reviewing the footage.

A single frame in which a tarakihi was clearly visible was selected from the collection of frames for each measured fish, to explore whether reliable length measurements could be identified from a still image, the pixel length was extracted from this image, and by using the calibration pattern a length in millimeters was predicted. Two approaches for achieving this are explored in subsection 6.3.1.

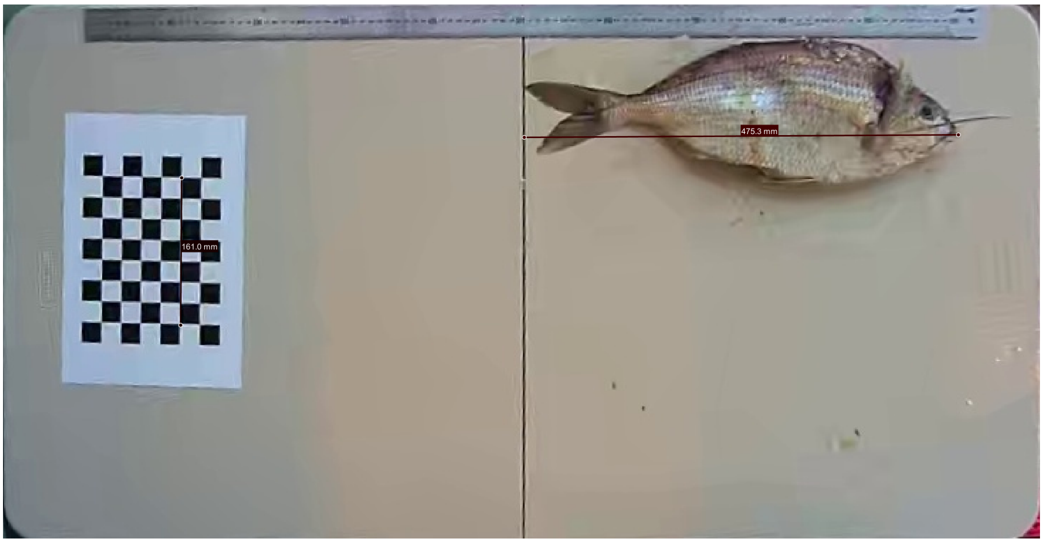


Figure 6.3: fish length measured in MATLAB

6.3 Experiment Design

6.3.1 Manual Length Extraction

The lengths of fish in images were first measured in an application in MATLAB. This allowed a user to place a line on an image by clicking to position its endpoints, a known length in millimeters may then be assigned to this line. This known length is then used to produce the millimeter lengths of additional lines in the image. Using this tool a line was created along 7 tiles of the calibration pattern and the known length of 161 millimeters was assigned to this line, this was calculated by multiplying the number of tiles (7) by the tile size (23). To produce a length for the fish a second line was placed along the fish from one extremity to the other, an example of this is shown in Figure 6.3.

6.3.2 Contour-Based Length Extraction

The second approach for extracting lengths from images used the method for extracting contours from masked images, first described in subsection 5.2.1, we established a human-supervised approach to quickly extract lengths from images. This method provided a faster approach to measuring length than using MatLab. However significant data cleaning had to be done by a human reviewer to remove all frames where hands were overlapping fish, or within two centimeters as this approach was not able to distinguish the class of each contour. The calibration pattern was found using the same approach used in subsection 6.2.2, and contours were derived from the edge detected image.

After the edge detection, edges of known objects in the image were removed, such as the ruler and the calibration pattern. This was done by setting the pixel values to zero in areas of the image where these items were present, as the camera and calibration pattern remained fixed this only had to be done once for the imagery where fish were not being measured. Where fish were being measured the pixels in which the ruler was visible were amended every few images to avoid the ruler's edges being included as part of the fish length measurement. Contours were then produced from this image using the same process of finding contours from segmentation inferences. Figure 5.1 shows an example of this process that was further developed to also generate masks, this process is described in section 5.2.1.

For the images of fish that were measured on-site a predicted length was outputted for every large contour found, using the pixel to mm ratio from the calibration pattern present. This was then passed to a human reviewer to select the correct contour that represented the fish present in the image. As no machine learning was used at this stage, all large contours were identified. As such, a human-supervised process was used to ensure that derived lengths were those of fish and not of other objects present

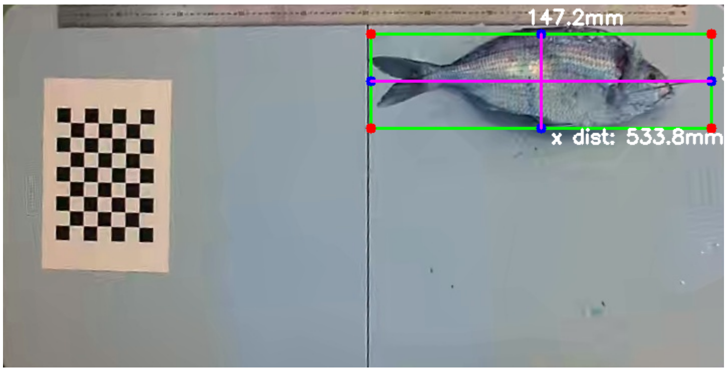


Figure 6.4: example of bad non-ML-derived contours

in the image such as hands, the calibration pattern or the contours of the tables' edges.

The human-supervised approach for selecting the correct contour was done by displaying the image and showing a bounding box around the contour that the predicted length was currently being derived. A user could cycle through each contour by clicking. If the bounding box tightly fit the fish then the length was recorded. When the bounding box did not tightly fit the fish as shown in Figure 6.4 then the extruding pixels in edge detection were removed manually. This approach was more reliable and produced fish lengths from the images that were closer to those measured on-site than the MATLAB application, this was likely due to using multiple points of reference from the calibration pattern to find the pixel to mm ratio, whereas the MATLAB application used only a single side.

6.4 Deriving Lengths from Videos

Inferring lengths of individual fish from multiple frames was an important stage to get right. To show that this system may predict lengths for footage gathered on fishing vessels, lengths inferred from the automated process would need to match up with those lengths measured on-site, when ordered chronologically. This would mean that double counting or missing fish would have a significant adverse effect when comparing true and predicted lengths.

Several challenges arose from deriving lengths from multiple segmentation masks. Prior work in generating lengths from inferred images had always focused on a single example per fish, in most of these images fish were clearly visible and did not have other items partially blocking their view. As a singular length must be recorded lengths over multiple frames, special care had to be taken to ensure the right lengths were derived from segmentation inferences. Partial inferences when hands were covering fish was one instance that would significantly alter measured lengths.

6.4.1 Preparing Data

Length measurements from videos were estimated by first inferring on all frames containing fish from each of the videos from the third visit, excluding the first bucket of fish. Frames from each video were already sorted into the usable frames as all frames containing an individual fish were placed in a sub-directory for that bucket as explained in section 3.4. All images from these sub-directories were copied into a singular folder, creating a large dataset of all the images containing fish, this allowed us to remove large empty sections of the video when no fish were visible, reducing the time needed to process the imagery.

6.4.2 Singulation over Multiple Images

After inference, all images and masks were ordered chronologically, by using the Unix timestamp of the video from which the image was extracted and the images corresponding frame number. Ensuring that important temporal information that may assist in the evaluation of predicted lengths was not lost when reading frames and masks.

Image-mask pairs were read to retrieve the length, in millimetres, of the fish in the image. The pixel to mm ratio was retrieved from the image

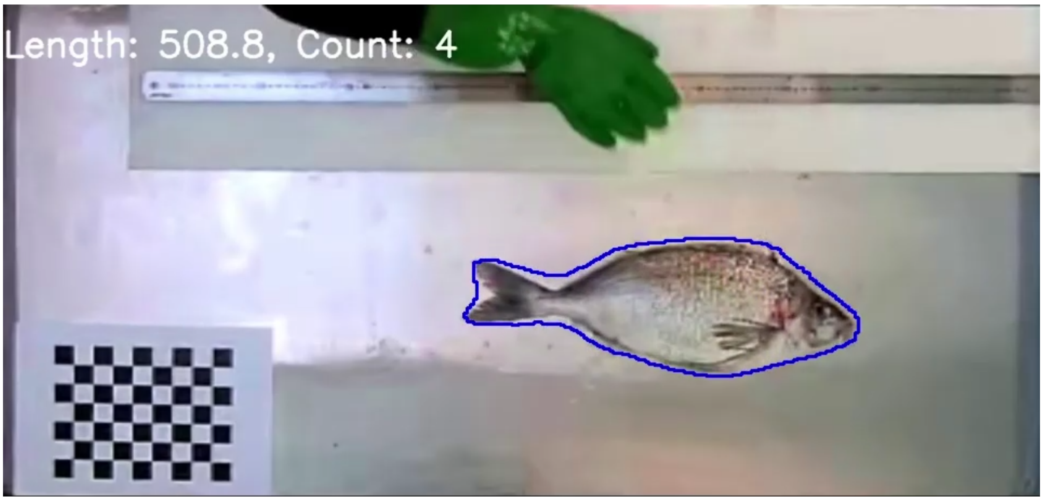


Figure 6.5: Shows a single frame from the video inference process conducted on imagery gathered in the third factory visit

by using the calibration pattern, if no pattern was found then the average from the dataset was used, for the third visit, this was 0.859. Contours of inferred fish from the mask were retrieved and used to retrieve the pixel length of the fish. Each image is saved after adding text positioned in the top left of the image for the current predicted length and the count of fish observed in the current video. A polygon is also added to the image to show the contours of the inferred segmentation mask. Single frames are then converted to a video by using FFmpeg [82], a video encoding software.

The median of similar lengths was used to determine the estimated length of the current fish, the median was chosen as false positives could skew the recorded max length if the measured contour significantly increased in size. Lengths were considered similar if they were within 1cm of the median length. If a length was greater than 1 centimetre above that of the estimated length for the current fish then we would store this value and only if there are 6 consecutive frames with a similar length do we consider this to be the new estimated length. The estimated length was set to zero after consecutive frames where no fish was present to ensure that the estimated length of one individual would not affect the recorded length of the following fish.

Allowing the estimated length for a fish currently in the frame to increase in size but not decrease, allowed us to infer lengths closer to the ground truth by reducing the impact of lengths when the view of the fish is partially obstructed. For instance, when hands were placing the fish down on the table, frames with a consistent length measurement were not uncommon but were not representative of the true length of the fish. By omitting previously measured lengths for the current fish when the measured length significantly increases we can remove the measured lengths from partially visible fish from affecting the recorded length when calculating the median.

False positives are a particular cause for concern as these would increase the lengths of measured fish should the contour of a false positive object join that of a fish. A serious infringement on catch size limits can lead to fines of up to 10,000 dollars and breaches in exceeding daily catch limits can result in criminal charges [14] ensuring that the model does not over predict the lengths of fish and accurately counts the number of fish observed is of the highest importance.

To reduce the number of false positives every contour from the inferred mask is subject to an evaluation through the processes described in subsection 5.3.2. If multiple contours made it through these checks then the pixel length of the one closest to the centre was taken.

Double counting individuals was another issue encountered, to reduce the likelihood of this occurring a check was implemented to ensure that a reasonable number of frames had passed since a fish had been observed before we assume the fish currently present is a new one. There would have to be 50 consecutive frames with no segmented pixels that pass the contour evaluation before we assume that the last fish observed has left the scene.

Once all frames have been processed and text for the length and count have been added images were saved and named according to the order in which they were read, these are then converted to a video with FFmpeg [82], an open source converter for video and audio. FFmpeg uses filenames to order the images when reading files for video conversion, by sorting the image-mask pairs prior to the processing we ensure that each image is saved with a time stamp relative to its chronological position and temporal order is retained in the outputted video.

Two comma-separated values (CSV) files are created as part of this process. CSV files are plain text files only supporting the use of text and numbers, consisting of rows, and columns separated by commas. Storing data in a CSV format allows it to be easily interpreted by both humans and most spreadsheet software such as Microsoft Excel. This format was chosen to ensure easy accessibility to the data created by this system.

The first CSV file contains a row for every frame in the video, with the following columns:

- Frame count: a counter for the number of frames observed in the current video
- Input file: the file name for the current frame
- Frame ID: an identifier for the current frame comprised of the UNIX timestamp of the video and the frame number from that video i.e. 1653948000_1006118
- Inference area: the total area of fish-like contours in the current image
- Total count: the total number of fish seen so far
- Median Length: the maximum length for the current fish
- Current Length: the length measured in the current frame

The second CSV file contained a single line for each counted fish, with just the frame count, input file, total count and the median predicted length. Having a single file with the outputted lengths for each individual allows users to easily gain insight into the length information for all the fish observed. This also provided an easy way to compare predicted lengths from the video to the ground truth lengths from the third factory visit.

6.5 Results

6.5.1 Comparing Methods for Manually Extracting Lengths from Images

A small experiment was conducted to compare both methods for extracting lengths with the human supervised approach. This experiment compared the lengths calculated from both approaches and compared them to the lengths measured on-site, shown in the first column in Table 6.1. The second column shows lengths calculated by using Matlab. The third and fourth columns show the lengths that were calculated using the semi-automated approach of estimating lengths from edge-derived contours, with the third showing those lengths estimated from the raw images and the fourth column showing lengths calculated from images that had been warped by the process described in section 4.2.1.

Table 6.1: compares the results of two human-conducted approaches for calculating lengths from images.

	Predicted		
True length	Matlab	Contour	Unwarped Contour
360	350	343.6	351.7
425	405	406.4	415.8
400	381.5	379.9	389.4
310	294.7	293.5	299.9
395	371.9	371.8	382.7
490	465.9	478.4	478.5
385	347.7	351.7	360.4
345	322.4	322.8	329.4
410	389.3	390.7	400.2
280	260	259.2	263.5
505	463.8	466.6	478.4
415	372	379.5	389
410	381.4	379	386.2
490	458.9	459.4	472.8
420	396.7	395.1	495.6
405	381.6	382.6	391.5
420	404.2	397.8	408.6
480	459.8	457.7	469.4
380	371	368.7	377.3
435	422.2	418.4	429
500	482.5	477.9	493.1
415	396.5	400.5	410.3
375	370.6	367	375.5
340	333.2	333.9	340.4
Absolute difference	21.133	21.162	14.766

The average absolute difference of estimated lengths minus true lengths is shown at the bottom of Table 6.1. These results show that both the approaches for manually extracting lengths from images were on average around 21 millimeters off the true lengths. The contour-based approach using warped images produced length estimates that were closest, on average, to the true lengths.

Though the results from the approach in Matlab achieved slightly better results, Matlab was prone to freezes when loading a new image from which the length was to be calculated. As such, the contour-based approach was used for deriving future manual length estimations from images.

6.5.2 Results: Synthetic Versus Real Datasets

The data used in this test consisted of a combination of real and synthetic images of fish. Each of the three models that were trained as part of this test used 700 images. Real images used in training both the model trained on only real imagery and the model trained on a mix of real and synthetic imagery were randomly chosen from a reduced dataset of annotated images. This image set was created by splitting the entire annotated image library (our holdout set was kept separate and not included in this split) into a training and validation library with an $80/20$ split, respectively. Training images were augmented by applying a horizontal flip (50% probability), rotation, brightness and gamma changes as well as Gaussian noise, to reduce overfitting.

The validation set contained 433 human-made image-mask pairs. This dataset contained images from all three visits and included images taken from both the fixed camera in the second and third visits and the free camera in the first visit. These were used for validating each of the models at the end of every epoch. During training, the model iteration that achieved the best validation loss was saved and used later for comparison.

For evaluating the performance of segmentation models trained on different kinds of imagery, three different training datasets were created. The first dataset contained 700 synthetic images and binary mask images created from human-drawn annotations.

The second dataset contained 700 image-mask pairs of real fish; this dataset was created by randomly shuffling a list of all the real images with annotated masks and selecting the first 700. Imagery in this dataset was entirely from the first and second factory visits.

The final dataset contained 700 image-mask pairs with 350 real images and 350 synthetic images generated by the StyleGANs model, both these datasets were created by selecting image names from a randomly shuffled list.

Table 6.2: shows a breakdown of the data used for each model in the experiment comparing model performance using synthetic data.

Training Data Source Comparison				
Model	Source			
	Visit1	Visit2	Visit3	Synth
Synthetic	0	0	0	700
Real	140	522	38	0
Mixed	83	252	15	350
Holdout	114	144	43	0
Validation	93	283	57	0

Models were trained for 60 epochs, with 87 steps per epoch. An EfficientNet backbone, with the sigmoid activation function, was used for each U-Net model, and Imagenet encoder weights were used to perform transfer learning. Each model's performance on the holdout set was measured and is shown in Table 6.3. The holdout set consisted of 301 annotated images, with 41 images from the first visit, 116 transformed images from the second visit and 144 transformed images from the fixed camera imagery set from the third factory visit.

In Table 6.3 the size refers to the total number of images used when training the model Inference masks were subject to a 50% threshold for both the intersection over union; which used the Jaccard index [40] and the F1 calculations; which used the dice coefficient [30]. Inferred images are denormalised between the 2^{nd} and 98^{th} percentiles and prediction mask values were limited between 0 and 1 before being multiplied by 255.Loss was calculated as a combination of Dice loss and binary focal loss, which introduces a hyperparameter, to penalize difficult classification examples more than easy ones, generalizing binary cross-entropy. The loss was calculated as $dice + (1 * focal)$.

Table 6.3: Compares the performance of the models trained on the three different datasets and a final model that included 2020 real images (all annotated images not in the validation or holdout set) and 700 synthetic images.

Model Performance Comparison				
Model	Metric			
	Size	Loss	IOU	F1 Score
Synthetic	700	0.084881	0.92409	0.95915
Real	700	0.016017	0.9875	0.99367
Mixed	700	0.014691	0.98811	0.99399
All Real	2020	0.016166	0.98554	0.99268
All Mixed	2722	0.014172	0.98747	0.99366

All three models achieved an IOU of over 90%. Performance of the model trained purely on synthetic data surpassed expectations on the holdout set as we had anticipated false positives from other objects in the scene to reduce this score more significantly. The performance of the real and mixed models was comparable. Despite only being trained on 350 examples of real fish (and 350 synthetic images) the mixed model's performance, measured by its F1 score, was marginally ($3.2e^{-4}$) above that of the model that was trained on 700 real images. Its predicted masks compared to those drawn by human annotators for the holdout set also outperformed the model trained on 2020 real images, achieving a lower loss and higher IOU and F1 score.

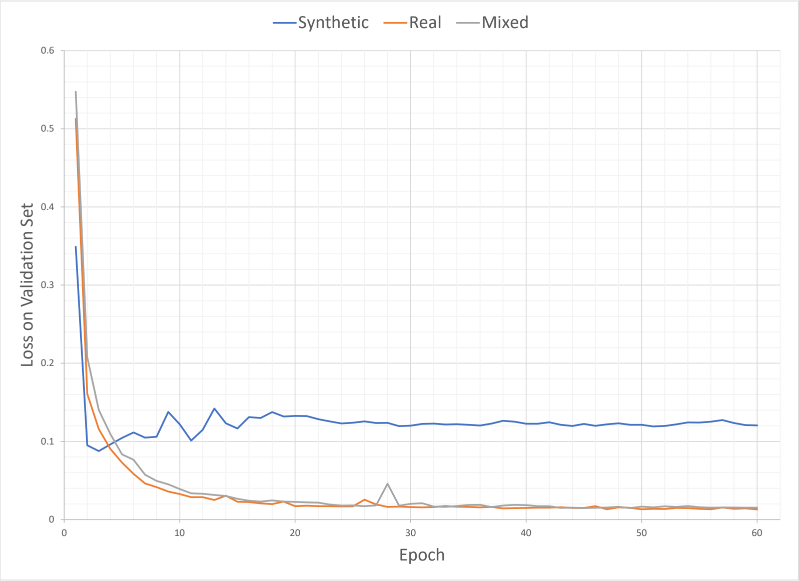


Figure 6.6: Comparison of model loss on the validation set at the end of each epoch

Contours used for deriving lengths were not evaluated by the EBM, as rough inferences by the model trained only on synthetic images meant that some individuals would be double counted. This was due to a large number of successive frames with contours that failed the EBM between observations of quality contours. The circle deviation range for quality contours was instead used, only contours that passed the check were used in our calculation of the median length. However, This meant that false

Table 6.4: Compares the predicted lengths generated by each model through the process described in section 6.4 to the true lengths measured on site in the third factory visit.

Predicted vs True Length Comparison					
Model	Metric				
	Mean (mm)	Absolute Mean (mm)	Absolute median (mm)	Std Dev	R^2
Synthetic	16.937	39.08	21.600	59.1402	0.5947
Real	-14.494	15.09	14.32	11.794	0.9423
Mixed	-22.108	23.39	22.885	15.898	0.9018
All Real	-11.832	13.10	11.265	12.573	0.9508
All Mixed	-10.280	11.25	9.9	10.533	0.9562

Table 6.5: Caption

positive predictions were more likely to affect the predicted lengths.

The image data used for deriving the lengths of these fish consisted of 56,187 frames taken from 128 fish, a single length is predicted per fish. The mean, absolute mean and median, and standard deviation were calculated from the $predicted - true$ lengths and the result is shown under the respective columns in Table 6.4. The R^2 for a linear regression of each model's predicted length against the true length is also shown.

The model trained on synthetic imagery had a significantly larger standard deviation than the other two models that were trained on 700 images, with an average absolute difference between predicted and true lengths of 39.08 millimeters. The synthetic model was the only model to have a mean over-prediction. This was identified to be a result of false positive inferences skewing estimated lengths.

The model trained entirely on real images achieved the lowest absolute mean difference of the three models used for comparison. On average the predicted lengths were 15 millimeters below the measured lengths. It also achieved the lowest standard deviation and highest R^2, suggesting that length estimates derived from this model's predictions best represent the true measured lengths despite the mixed model outperforming it in metrics from the holdout set.

The model trained on all available data (2020 real images and 700 synthetic images) outperformed the model trained on 2020 real images. This improvement was consistent, not only on the holdout image masks but also for the predicted lengths as shown in Table 6.4. The absolute mean of the difference in predicted lengths by the model trained on only real images was 1.85 millimeters higher than the model trained on all data. The difference between the results found from these two models and those trained on a total of 700 images suggests that synthetic data does bring benefits with regard to training DCNNs. These results are particularly evident in the first test which compared the loss, IOU and F1 scores of all the models on a holdout set.

6.5.3 Final Length Estimation

Some margin of error exists between the predicted and true lengths, in tests of imagery from both the second and third factory visits the length prediction process consistently underpredicts the length of the target fish. Though tests that applied a birds-eye transform to the images appeared to rectify this to some extent any transformation to the image that alters the proportion of the calibration pattern differently to the fish will reduce the accuracy of the predicted length.

A test on inferred lengths from two different images of the same fish using the same segmentation mask showed how reducing the size of the image can alter the final predicted length. In this test all single images of the fish measured in the second factory visit were inferred on, generating 101 masks. These masks were then used to output a length per image

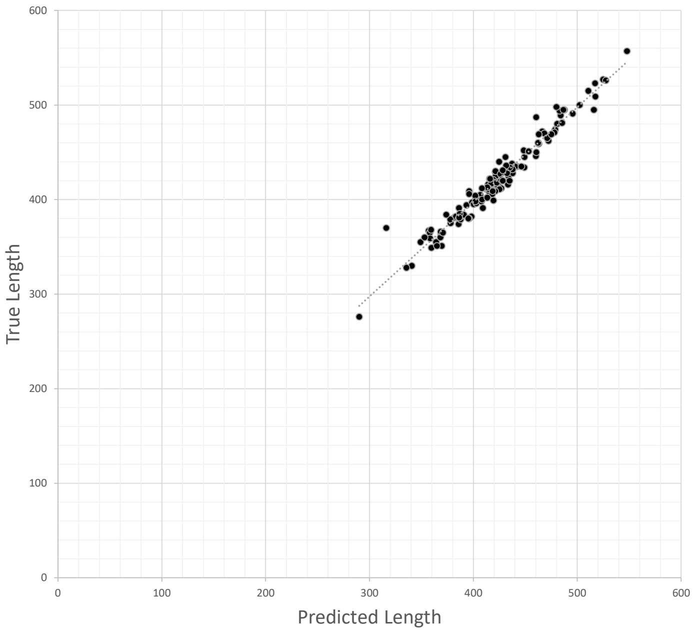


Figure 6.7: True lengths compared to adjusted predicted lengths.

using both the image that had been passed through the data loader for inference and then the original image that had not been altered. The average absolute difference in predicted lengths for images which had been passed through the data loader was 18.82 millimeters while the difference for the raw images was 18.114 millimeters. Though on average this change is less than a millimeter any inaccuracies in the projected size may have detrimental effects on the predicted lengths.

To correct for unknown error that may affect the predicted lengths a linear regression for the predicted lengths for the second visit was conducted providing us with a linear transformation for future predicted lengths to be adjusted. The average absolute difference between adjusted predicted and true lengths, for recorded fish from the third visit, after this adjustment was 7.523 millimeters. A single adjusted length was calculated as:

$$L_i = P_i * 0.9196 + 45.593 \tag{6.6}$$

Equation 6.6: Linear adjustment for predicted values. Where L is the adjusted length for fish i and P is the predicted length for an individual derived from a video. The adjusted lengths are recorded for each fish and compared to the true lengths in Figure 6.7.

6.6 Chapter Summary

In this chapter, we explained how the moments of contours are calculated and used to target the contour that was likely to be a fish that was closest to the centre of the image. A minimum enclosing circle of this fish contour was used to extract a single pixel length for the fish in the current image. We then described how a scale factor, that is calculated from the calibration pattern, is used to translate this pixel length into millimeters.

Our approach for a human supervised approach to manually extract fish lengths from images was explained. The first approach used a Matlab application to draw a single line over an object of known length (the checkerboard pattern) to estimate the length of other objects in the scene. The second approach used an edge detector and some manual removal of unwanted edges, such as those of the ruler, to estimate the lengths of the fish in each image. Both approaches achieved similar results. However, the approach that used contours was far faster at extracting the lengths of fish.

We also described, in detail, an approach to extract accurate lengths from multiple image frames of fish. This used the median of all large lengths to calculate the length for each observed fish. By allowing the resetting of the stored lengths for each individual we were able to remove lengths from partially visible fish from affecting our length prediction.

Lengths predicted from videos, similarly to those calculated through the human-based approaches, were consistently under predicting the true length of the fish. A simple linear regression was calculated on predicted lengths from the second factory visit against the true lengths from this visit. The equation of the linear regression was used to adjust the predicted lengths from the third visit. These adjusted lengths were compared to the true lengths measured for the third visit and were on average 7.523 millimeters away from the true lengths, achieving our goal of achieving predicted lengths within 1 centimetre of the measured lengths.

Chapter 7

Conclusions and Future Work

In this chapter, we present the contributions and a summary of the key findings from the work that was conducted as part of this thesis. Each of the main components of this work is summarised following an overview of the core objectives. Finally, we conclude with a discussion of future research.

7.1 Conclusions

The overall goal of this thesis was to provide a method for accurately measuring the lengths of fish from video footage and provide estimates for both the count and length of these fish. In our pursuit to achieve this goal the use of synthetic imagery to assist in training DCNNs was also explored as image data for the fishing industry can often be difficult to acquire as due to privacy concerns. Interpretable methods for evaluating inference results were also investigated and provided us with a method of reducing unfavourable inferences that may otherwise skew our predicted lengths.

Strict restrictions on reported lengths for undersized catch meant that a key objective of this research was to achieve centimetre accurate length predictions. We were able to achieve this goal on a holdout set of 128 images with an average absolute difference between predicted and measured lengths of 7.523 millimetres.

Concluding that models trained on both real and synthetic imagery will outperform those trained on just real imagery cannot be made from this research alone as there are too many external influences that may affect these results. The randomly chosen images for training each model, for example, may have been better suited for explaining the holdout set in the mixed model than those in the model trained on 700 real images. However, the results from this research are similar to those from other works exploring the use of synthetic imagery [74]. Suggesting that synthetic imagery for training deep convolutional neural networks performs best when used in conjunction with real images.

7.1.1 Interpretable Methods for Evaluating Contours

For accurate length estimation, the use of all contours extracted from raw inferences results in distorted length measurements, poor inferences or false positives can lead to a predicted length unrepresentative of the true length for the current fish. We investigated the use of interpretable methods for explaining features of contours that are useful in determining their quality. These interpretable techniques allowed us to create a process that leverages the power of machine learning to find patterns in our data while also providing inferences that are justifiable.

Various shape features were calculated from the contours extracted from segmentation inferences. These shape features allowed us to extract key geometric properties of high and low-quality inferences which were used to train an explainable boosting classifier. The use of an explainable model not only allowed us to quickly classify the contours of new inferences but also signalled value ranges for the calculated shape features that were commonly associated with high or low-quality contours. These were presented in the form of shape functions.

The shape functions calculated from these features when training the 'glassbox' model were visualised by plotting the calculated shape feature against a contribution score, this explained the extent to which feature values were associated with the positive or negative class. Shape feature values within the range associated with a high-quality contour that did not pass through the EBC successfully were flagged as potential poor inferences. This approach allowed us to target our training on hard-to-classify images, maximising the information gained at each stage of training. Extra care was also given to false negative predictions, negative predictions were subject to thorough analysis to ensure that we did not only reinforce the model's bias.

Shape features were also used to restrict the inferences that contributed to the final length measurement for each individual by only using contours that had a circle deviation between 0.2 and 0.4. All other contours that did not meet this requirement did not have their pixel length calculated.

Interpretable techniques are able to provide a method for evaluating contours that is accountable. If useful contours are excluded from the length estimation of an individual, the features that contributed to this decision can be plotted and the cause of the issue may be understood. As penalties may be incurred for keeping undersized fish, a fish that does not pass the contour evaluation that is below 25cm would cause distrust in this automated approach. By having an explainable model we are able to understand why this occurred and may better implement a resolution.

7.1.2 Application of Synthetic Imagery

In line with data-driven machine learning the use of synthetic imagery in training a segmentation model was explored. A Stylegans model was used to generate synthetic images of fish. 700 of these synthetic images were annotated and were used to train segmentation models.

Results for models trained on entirely synthetic imagery were promising, achieving an IOU of over 92% on the holdout set of annotated images.

The inferences created by the synthetic model for predicting the lengths of individuals from video footage were relatively poor compared to models trained on entirely real data. Though the fairness of such a test is debatable, as synthetic images were limited to a size of 320×320 and therefore lacked the large amount of negative training available in real images, from areas where no fish were present. This resulted in skewed lengths from false positive inferences, our approach of using the median to combat the effects of outliers was not sufficient to produce reliable length estimates for these instances.

The performance of models trained on a combination of both real and synthetic imagery was similar to that of a model trained on the same quantity of only real data. These observations are in line with similar research and suggest that synthetic imagery may be used to assist in the development of data in fields or edge cases where there exists a shortage of data. However, the practical implementation of synthetic datasets requires a thorough understanding of their potential limitations. One such limitation was encountered in our tests in which lengths were predicted from inferred segmentation masks, models trained on synthetic data were not trained on enough data for negative classes, resulting in an over-prediction of the positive class.

The use of synthetic imagery to bolster small datasets, particularly for providing greater samples of uncommon instances, has great potential. Advancements in neural network architecture and the handling of data used in training [42] has significantly reduced the barriers that had limited the development and implementation of synthetic imagery.

7.1.3 Length Estimation

This research was successful in developing an automated system to estimate the length of partitioned objects from an image. This was achieved by using a segmentation model to infer which pixels in the image represent the target. Contours of the segmented objects were retrieved, by using a minimum enclosing circle of the detected contour a pixel length for our target was derived.

The use of a checkerboard pattern as a fiducial marker allowed us to calculate the scale factor to express pixel lengths in millimeters. This was done by retrieving the pixel length for each side of the pattern, by calculating the distance of the extreme key points and comparing this to the known distance in millimeters based on the number of checkerboard tiles identified.

We explored various factors that may hinder the production of accurate length estimates, such as the angle and distance of the camera relative to the pattern. This showed that orientation on the Y and X axes as well as distance on the Z axis is particularly significant in determining the accuracy of our inferred lengths. The use of a handheld camera was useful for collecting a large variety of data. However to produce accurate length estimations a fixed camera placed directly above the area of interest is best.

Effective use of this marker was sensitive to any changes that may reduce its projected size in the image relative to the object being measured. As such, the original image or the earliest possible version should be used to extract the pixel to millimeter ratio. Where no pattern was visible the average ratio for the video was used, avoiding zero predictions for images in which the pattern was obscured.

A system for deriving lengths from partitioned fish in a video was developed, by using the median length derived from a range of the largest segmentation masks that were consistently of a similar size we were able to significantly reduce the effect of outliers on our predicted lengths. Using this process we were able to achieve an absolute mean difference between predicted and true lengths of under 1 centimetre.

7.2 Future Work

Before this system may be placed on a vessel a significant amount of training data should be added. The diversity of images in this research, though gathered in different ways, is not enough to guarantee that these results will be replicated in a less forgiving environment. We hope to use the lessons provided by this research to set a strong foundation upon which future iterations may be built upon.

Including multiple segmentation classes will also be explored, allowing us to identify species as well as length. Enabling all personnel involved to have close to real-time information about the length and count for individual species would provide incredible value and allow for more informed decisions to be made while fishing vessels are still out at sea.

Future work will also explore the use of instance segmentation for counting individuals and storing information about recently seen fish that may have left and re-entered the scene. The current approach for counting and measuring has no resilience against fish that are not manually partitioned. Fish that are close to one another in an image would share the same contour and result in a single length estimation which may be unrepresentative of the length of either individual.

The use of synthetic data for the entire region of interest would provide a better understanding of the use of synthetic imagery in replacing real images. However, we recognise that access to the compute required to generate high-resolution synthetic images is not always readily available. Despite recent innovations in generative adversarial networks meaning far less imagery is required for training There are still some barriers to its practical application in some industries. In recent research by NVIDIA, DatasetGANs [86] was developed, this procedure not only produces synthetic images but also the relevant multi-class masks with minimal human effort. The use of such a model would significantly reduce the burden of annotating synthetic images and potentially allow for a far greater quantity of data to be used in training the segmentation model.

Automating the extraction of the region of interest is another area in which we would like to conduct further research. The use of a calibration pattern under the area in which fish are measured or ground control points that are easily identified such as bright green would facilitate creating an automated approach to narrow down the area in which the desired fish are to be measured. The use of bright green gloves for handling the fish in the second and third factory visits showed the benefits of using contrasting colours as false positives over hands dropped almost entirely.

Bibliography

[1] Department of Conservation new zealand's marine biodiversity . https://www.doc.govt.nz/nature/habitats/marine/new-zealands-marine-biodiversity/. [Accessed 02-Oct-2021].

[2] MathWorks documentation: extrinsicstocamerapose. https://au.mathworks.com/help/vision/ref/extrinsicstocamerapose.html. [Accessed 21-Jun-2022].

[3] MATLAB camera calibrator . https://au.mathworks.com/help/vision/ug/camera-calibration.html. [Accessed 17-Dec-2021].

[4] Ministry for Primary Industries how we count fish . https://www.mpi.govt.nz/fishing-aquaculture/sustainable-fisheries/the-health-of-new-zealands-fisheries/how-we-count-fish/. [Accessed 05-Sept-2021].

[5] Ministry for Primary Industries review of sustainability measures for east coast tarakihi (tar 2, tar 3 and eastern portions of tar 1 and tar 7) for 2022/23. https://www.mpi.govt.nz/dmsdocument/51757-Review-of-Sustainability-Measures-for-East-Coast-Tarakihi-TAR-2-TAR-3-and-eastern-portions-of-TAR-1-and-TAR-7-for-202223-Discussion-document.

[6] Numpy documentation . https://numpy.org/doc/stable/index.html. [Accessed 1-Dec-2021].

[7] OpenCV contours . https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html. [Accessed 18-April-2022].

[8] OpenCV geometric image transformations . https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html#ga8c1ae0e3589a9d77fffc962c49b22043. [Accessed 1-Dec-2021].

[9] OpenCV miscellaneous image transformations . https://docs.opencv.org/3.4/d7/d1b/group__imgproc__misc.html#ggaa9e58d2860d4afa658ef70a9b1115576a147222a96556ebc1d948b372bcd7ac59. [Accessed 19-Jan-2022].

[10] Stark bros 16.8m fishing vessel: Commercial vessel: Boats online for sale: Steel: New south wales (nsw) - south island . https://www.boatsonline.com.au/boats-for-sale/used/commercial-vessel/stark-bros-16-8m-fishing-vessel/234138. [Accessed 05-Mar-2022].

[11] MicroImages geometric analysis: Polygon shape properties . https://www.microimages.com/documentation/TechGuides/81PolyShape.pdf, Sep 2014. [Accessed 02-May-2022].

[12] Fisheries Inshore New Zealand tarakihi management strategy . https://www.inshore.co.nz/fisheries/tarakihi/, 2019. [Accessed 13-Dec-2021].

[13] Anaconda2020 state of data science. https://www.anaconda.com/state-of-data-science-2020, 2020. [Accessed 15-Nov-2021].

[14] Charges and penalties for fisheries offences . https://communitylaw.org.nz/community-law-manual/chapter-34-common-crimes/paua-poaching-and-other-fisheries-offences/charges-and-penalties-for-fisheries-offences/, Sep 2021. [Accessed 13-Jan-2022].

[15] Minderoo Foundation new report finds global fisheries in far worse state than previously estimated . https://www.minderoo.org/global-fishing-index/news/new-report-finds-global-fisheries-in-far-worse-state-than-previously-estimated/, 2021. [Accessed 22-Jan-2022].

[16] Ministry for Primary Industries monitoring and observing fishing activity . https://www.mpi.govt.nz/fishing-aquaculture/sustainable-fisheries/strengthening-fisheries-management/monitoring-observing-fishing-activity/, Jul 2021. [Accessed 11-Nov-2021].

[17] Ministry for Primary Industries digital monitoring of commercial fishing . https://www.mpi.govt.nz/fishing-aquaculture/commercial-fishing/fisheries-change-programme/digital-monitoring-of-commercial-fishing/, Jun 2022. [Accessed 18-Jan-2022].

[18] Álvarez-Ellacuría, A., Palmer, M., Catalán, I. A., and Lisani, J.-L. Image-based, unsupervised estimation of fish size from commercial landings using deep learning. *ICES Journal of Marine Science 77*, 4 (2020), 1330–1339.

[19] Álvarez-Ellacuría, A., Palmer, M., Catalán, I. A., and Lisani, J.-L. Image-based, unsupervised estimation of fish size from commercial landings using deep learning. *ICES Journal of Marine Science 77*, 4 (2020), 1330–1339.

[20] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning* (2017), PMLR, pp. 214–223.

[21] Arnarson, H., and Khodabandehloo, K. *Fish processing using computer vision and robots*. Springer US, Boston, MA, 1993, pp. 26–47.

[22] BAKER, B., KANITSCHEIDER, I., MARKOV, T., WU, Y., POWELL, G., MCGREW, B., AND MORDATCH, I. Emergent tool use from multi-agent autocurricula. *ArXiv abs/1909.07528* (2020).

[23] BROAD, T., AND GRIERSON, M. Transforming the output of gans by fine-tuning them with features from different datasets. *arXiv preprint arXiv:1910.02411* (2019).

[24] BROWNLEE, J. How to implement the frechet inception distance (fid) for evaluating gans . `https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/`, October 2019. [Accessed 07-Dec-2021].

[25] BUSLAEV, A., IGLOVIKOV, V. I., KHVEDCHENYA, E., PARINOV, A., DRUZHININ, M., AND KALININ, A. A. Albumentations: Fast and flexible image augmentations. *Information 11*, 2 (2020).

[26] CHOI, Y., UH, Y., YOO, J., AND HA, J.-W. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 8188–8197.

[27] COLL, M., LIBRALATO, S., TUDELA, S., PALOMERA, I., AND PRANOVI, F. Ecosystem overfishing in the ocean. *PLoS one 3*, 12 (2008), e3881.

[28] DANIEL PAULY, D. Z. Catch reconstructions reveal that global marine fisheries catches are higher than reported and declining. *Nat Commun 7* (2016).

[29] DAVIES, A. J., ROBERTS, J. M., AND HALL-SPENCER, J. Preserving deep-sea natural heritage: Emerging issues in offshore conservation and management. *Biological Conservation 138*, 3 (2007), 299–312.

[30] DICE, L. R. Measures of the amount of ecologic association between species. *Ecology 26*, 3 (1945), 297–302.

[31] DUTTA, A., AND ZISSERMAN, A. The via annotation software for images, audio and video. In *Proceedings of the 27th ACM international conference on multimedia* (2019), pp. 2276–2279.

[32] EILERTSEN, G., TSIRIKOGLOU, A., LUNDSTRÖM, C., AND UNGER, J. Ensembles of gans for synthetic training data generation. *arXiv preprint arXiv:2104.11797* (2021).

[33] GEIGER, R. S., COPE, D., IP, J., LOTOSH, M., SHAH, A., WENG, J., AND TANG, R. "Garbage in, garbage out" revisited: What do machine learning application papers report about human-labeled training data? *Quantitative Science Studies 2*, 3 (11 2021), 795–827.

[34] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., AND BENGIO, Y. Generative adversarial nets. *Advances in neural information processing systems 27* (2014).

[35] HAO, K. The coming war on the hidden algorithms that trap people in poverty . `https://www.technologyreview.com/2020/12/04/1013068/algorithms-create-a-poverty-trap-lawyers-fight-back/`, Jan 2022. [Accessed 26-Jul-2022].

[36] HE, K., GKIOXARI, G., DOLLÁR, P., AND GIRSHICK, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2961–2969.

[37] HEUSEL, M., RAMSAUER, H., UNTERTHINER, T., NESSLER, B., AND HOCHREITER, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems 30* (2017).

[38] HUI, J. Gan — why it is so hard to train generative adversarial networks! . `https://jonathan-hui.medium.com/gan-why-it`

-is-so-hard-to-train-generative-advisory-networks-819a86b3750b, 2018. [Accessed 05-Dec-2021].

[39] IAKUBOVSKII, P. Segmentation models . https://github.com/qubvel/segmentation_models, 2019. [Accessed 02-Feb-2022].

[40] JACCARD, P. The distribution of the flora in the alpine zone.1. *New Phytologist 11*, 2 (1912), 37–50.

[41] JEAN, N. Fréchet inception distance . https://nealjean.com/ml/frechet-inception-distance/, July 2018. [Accessed 07-Dec-2021].

[42] KARRAS, T., AITTALA, M., HELLSTEN, J., LAINE, S., LEHTINEN, J., AND AILA, T. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems 33* (2020), 12104–12114.

[43] KARRAS, T., LAINE, S., AND AILA, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 4401–4410.

[44] KARRAS, T., LAINE, S., AITTALA, M., HELLSTEN, J., LEHTINEN, J., AND AILA, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 8110–8119.

[45] KOBER, J., BAGNELL, J. A., AND PETERS, J. Reinforcement learning in robotics: A survey. *Int. J. Rob. Res. 32*, 11 (Sept. 2013), 1238–1274.

[46] LIN, Y., HUNG, T., AND HUANG, L. T. Engineering equity: How AI can help reduce the harm of implicit bias. *Philosophy and Technology 34*, 1 (2020), 65–90.

[47] LIU, S., QI, L., QIN, H., SHI, J., AND JIA, J. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 8759–8768.

[48] LOU, Y., CARUANA, R., AND GEHRKE, J. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (2012), pp. 150–158.

[49] LOU, Y., CARUANA, R., GEHRKE, J., AND HOOKER, G. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), pp. 623–631.

[50] LUNDBERG, S. M., AND LEE, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems 30* (2017), 4765–4774.

[51] LUO, A. Z. Ocean fish stocks on "verge of collapse," says irin report . `https://www.newsecuritybeat.org/2017/02/ocean-fish-stocks-on-verge-collapse-irin-report/`. [Accessed 22-Oct-2021].

[52] MCGLONE, D. Measuring district compactness in postgis . `https://www.azavea.com/blog/2016/07/11/measuring-district-compactness-postgis/`, Jan 2020. [Accessed 11-Mar-2022].

[53] METZ, L., POOLE, B., PFAU, D., AND SOHL-DICKSTEIN, J. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163* (2016).

[54] MO, S., CHO, M., AND SHIN, J. Freeze the discriminator: a simple baseline for fine-tuning gans. *arXiv preprint arXiv:2002.10964* (2020).

[55] Monkman, G. G., Hyder, K., Kaiser, M. J., and Vidal, F. P. Using machine vision to estimate fish length from images using regional convolutional neural networks. *Methods in Ecology and Evolution 10*, 12 (2019), 2045–2056.

[56] Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences 116*, 44 (2019), 22071–22080.

[57] Nadkarni, A. Tarakihi fishing industry admits to breaking overfishing restrictions . `https://www.stuff.co.nz/business/124622490/tarakihi-fishing-industry-admits-to-breaking-overfishing-restrictions`. [Accessed 16-Jan-2022].

[58] Ng, A. Convolutional neural networks . `https://www.coursera.org/learn/convolutional-neural-networks`, 2019. [Accessed 4-Feb-2022].

[59] Ng, A. Mlops: From model-centric to data-centric ai . `https://www.deeplearning.ai/wp-content/uploads/2021/06/MLOps-From-Model-centric-to-Data-centric-AI.pdf`, 2021. [Accessed 19-Jul-2022].

[60] Nielsen, J., Reines, J. H., and Jespersen, C. M. Quality assurance in the fishing industry with emphasis on the future use of vision techniques. In *Fish quality control by computer vision*. Routledge, 2017, pp. 1–20.

[61] NOAA. Baited remote underwater video station (bruvs) surveys of fish in the hawaiian archipelago from 2012 to 2014 . `https://www.fisheries.noaa.gov/inport/item/25248`, 2014. [Accessed 18-Nov-2021].

[62] NORI, H., JENKINS, S., KOCH, P., AND CARUANA, R. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223* (2019).

[63] ORŠOLIĆ, I. Camera calibration: Explaining camera distortions. `https://ori.codes/artificial-intelligence/camera-calibration/camera-distortions/`.

[64] OYINLOLA, M. A., REYGONDEAU, G., WABNITZ, C. C., AND CHEUNG, W. W. Projecting global mariculture diversity under climate change. *Global change biology 26*, 4 (2020), 2134–2148.

[65] PALMER, M., ÁLVAREZ-ELLACURÍA, A., MOLTÓ, V., AND CATALÁN, I. A. Automatic, operational, high-resolution monitoring of fish length and catch numbers from landings using deep learning. *Fisheries Research 246* (2022), 106166.

[66] PEREZ, L., AND WANG, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017).

[67] PODCZECK, F. A shape factor to assess the shape of particles using image analysis. *Powder Technology 93*, 1 (1997), 47–53.

[68] QIU, C., CUI, J., ZHANG, S., WANG, C., GU, Z., ZHENG, H., AND ZHENG, B. Transfer learning for small-scale fish image classification. In *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)* (2018), pp. 1–5.

[69] REOCK, E. C. A note: Measuring compactness as a requirement of legislative apportionment. *Midwest Journal of Political Science 5*, 1 (1961), 70.

[70] RIBEIRO, M. T., SINGH, S., AND GUESTRIN, C. ”why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of*

the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016 (2016), pp. 1135–1144.

[71] RICHARDSON, L. F. The problem of contiguity: an appendix to statistics of deadly quarrels. *General systems yearbook 6* (1961), 139–187.

[72] RODRIGUEZ, A., RICO-DIAZ, A. J., RABUNAL, J. R., PUERTAS, J., AND PENA, L. Fish monitoring and sizing using computer vision. In *International work-conference on the interplay between natural and artificial computation* (2015), Springer, pp. 419–428.

[73] ROH, Y., HEO, G., AND WHANG, S. E. A survey on data collection for machine learning: A big data - ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering 33* (2021), 1328–1347.

[74] ROS, G., SELLART, L., MATERZYNSKA, J., VAZQUEZ, D., AND LOPEZ, A. M. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 3234–3243.

[75] RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., BERG, A. C., AND FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV) 115*, 3 (2015), 211–252.

[76] SCHWARTZBERG, J. E. Reapportionment, gerrymanders, and the notion of compactness. *Minn. L. Rev. 50* (1965), 443.

[77] SHAHADAT UDDIN, ARIF KHAN, E. H., AND MONI, M. A. Comparing different supervised machine learning algorithms for disease prediction. *BMC medical informatics and decision making 19* (2019), 281–297.

[78] STRICKLAND, E. Andrew ng: Unbiggen ai. `https://spectrum.ieee.org/andrew-ng-data-centric-ai`, 2022.

[79] TAN, M., AND LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (2019), PMLR, pp. 6105–6114.

[80] TAN, M., AND LE, Q. V. Efficientnet: Improving accuracy and efficiency through automl and model scaling . `https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html`, 2019. [Accessed 08-Dec-2021].

[81] TAN, M., PANG, R., AND LE, Q. V. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 10781–10790.

[82] TOMAR, S. Converting video formats with ffmpeg. *Linux Journal 2006*, 146 (2006), 10.

[83] TURING, A. M. Computing machinery and intelligence. *Mind 49*, 10 (1950), 433–460.

[84] VICTORERO, L., WATLING, L., DENG PALOMARES, M. L., AND NOUVIAN, C. Out of sight, but within reach: A global history of bottom-trawled deep-sea fisheries from $>$ 400 m depth. *Frontiers in Marine Science 5* (2018), 98.

[85] YU, F., ZHANG, Y., SONG, S., SEFF, A., AND XIAO, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365* (2015).

[86] ZHANG, Y., LING, H., GAO, J., YIN, K., LAFLECHE, J.-F., BARRIUSO, A., TORRALBA, A., AND FIDLER, S. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021), pp. 10145–10155.

Appendix A

File Structures for Fish Images

The directory structure is shown below.

```
Factory_Visit_3
└─ Bucket4
   ├─ tarakihi_10001_416
   ├─ tarakihi_10002_305
   └─ tarakihi_10003_483
      ├─ 1651781700_frame_1007994.jpg
      └─ ...
```

Fig A shows the structure of the directories for storing our images. These are described in descending order as; the folder for the visit number, directories for each bucket of fish, directories for each individual fish, all the relevant images for this fish.