University of Bergen
Department of Informatics

Object Tracking Approach for Catch Estimation on Trawl Surveys

Author: Peter Liessem
Supervisors: Ketil Malde and Vaneeda Allken

UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

June, 2023

Abstract

In the Norwegian Sea, coordinated multinational surveys are regularly undertaken with the aim of assessing the size and composition of marine life populations - a fundamental practice for ensuring long-term ecological sustainability. The role of trawling in these surveys is pivotal, as it offers a direct, fisheries-independent, sampling method. This direct approach enables an accurate assessment of the abundance and diversity of fish populations, providing a clearer picture of the marine ecosystem's health. However, traditional trawling leads to increased bycatch mortality rate and can hurt biodiversity.

Scantrol Deep Vision is a company that focuses on the development of advanced underwater vision technology. They have launched a product known as "Deep Vision" which aims to revolutionize marine research by providing an eco-friendly method for fish sampling and stock analysis without the need to bring the catch onboard. The technology takes pictures of marine life during trawling. Our project attempts to estimate the marine life count and distribution based on these images.

Previous work, by Allken et al., on this problem involved fine tuning a RetinaNet model to detect and classify four categories of fish: blue whiting, herring, mackerel, and mesopelagic fishes. They ran the model on images from 20 trawl stations and trained a linear regression model for each species, except mesopelagic fishes, on the resulting object detection count generated form each station and their respective catch count. They used the R-squared metric to quantify how well the regression models fit the data and got the scores 0.74, 0.62, and 0.84 for blue whiting, herring, and mackerel, respectively. Mesopelagic fishes are generally too small to be caught by the trawls and were not part of any regression.

In our project, we aim to enhance the precision of estimation on marine life count and species distribution. We employ object tracking to a dataset generated by the same RetinaNet model used in previous studies for object detection. Subsequently, we apply linear regression to the count derived from these tracks. Our most effective model demonstrates promising results, on the 20-station dataset used in previous work, with R-squared scores of 0.84, 0.96, and 0.87 for blue whiting, herring, and mackerel, respectively. These results underscore the potential efficacy of object tracking in addressing this problem.

Acknowledgements

My sincerest thanks go to my supervisors for their encouragement, insightful comments, and challenging questions, all of which have broadened my understanding and perspective.

I am also deeply grateful to my fellow master's students who have provided support, companionship, and invaluable insights throughout this process.

To my family and friends, your optimism, understanding and constant support have been a beacon during challenging times. The moments of joy and relaxation you provided helped me step back when needed and made the entire process more manageable and far less solitary.

Peter Liessem
Thursday 15th June, 2023

Contents

List of Figures

List of Tables

List of Equations

Listings

Chapter 1

Introduction

This chapter introduces the project relating to the thesis and goes through related works.

Multi-Object Tracking (MOT) encompasses the process of monitoring multiple entities across a video sequence over a period of time. This field has seen a considerable advancement in research over the past few decades, fueling significant progress in technology and algorithms. MOT plays a vital role in diverse applications, including surveillance, traffic analysis, and the domain of autonomous vehicles.

However, the implementation of MOT is not devoid of challenges. Common obstacles encountered during the process include class imbalance, occlusion, changes in appearances, misclassification, and low frame rates.

This paper showcases our investigation into enhancing the precision of agent counting in image sequences, utilizing a pre-existing dataset derived from an object detection algorithm on images captured by trawlers. These trawlers photograph marine life as they enter the net.

Our primary objective is to improve the count and distribution prediction accuracy of various fish species, specifically blue whiting, herring, and mackerel. Achieving this aim would facilitate a more dependable analysis and evaluation of fish populations in targeted regions.

1.1 Thesis Statement and Context

In the Norwegian Sea, regular multinational surveys are conducted to estimate the stock size for marine life, ensuring long-term sustainability. Trawls play a crucial role in gathering fisheries-independent abundance data for stock assessments because they provide a direct sampling method to assess the abundance and composition of fish populations [12].

Although trawling allows us to directly sample fish population, it does have some shortcomings. When trawling, different species entering the trawl are accumulated and mixed in the codend, the end part of a trawl net where fish and other catch are collected and retained during fishing operations. This means that the precise location and distribution of individual species within the water column may not be accurately represented by trawl data.

In addition to this, the trawl nets have large selectivity or catchability biases depending on factors such as the mesh size, effective sampling volume, behavior, and size of the target species. These biases can result in an incomplete representation of the species composition and abundance in the sampled population. Furthermore, when a trawl haul results in a significant catch of marine life, the catch composition is determined from a subsample of the total catch. This approach introduces an additional layer of uncertainty to the data.

To supplement the trawl data, an in-trawl Deep Vision camera system has been implemented 3 meters in front of the codend. The Deep Vision camera system continuously takes photos throughout the trawl operation. Looking at these pictures can give great insight into the composition and abundance of the waters covered by the trawl. However, manually reviewing this extensive dataset is a time-consuming process that lacks scalability.

Leveraging machine learning techniques, we utilize the in-trawl camera system to track agents and approximate the catch data. Previous work has shown that machine learning methods can be used to automate the classification of Deep Vision images into species [2], and to estimate the species distribution within each trawl haul. Comparing the catch data to the prediction counts using machine learning is not straightforward as each fish can be imaged more than once and the number of times an individual fish is captured in consecutive images may be species-dependent. Additionally, a fraction of the predictions were erroneous. Tracking individuals would help improve both the catch estimation and

reduce species misidentification, thereby improving the estimation of species distribution within trawl hauls.

Improving these systems could significantly contribute to more sustainable marine research practices and potentially pave the way for using open codends in trawls. Having an open codend would effectively reduce the mortality rate among rare or endangered species, thereby aligning the work with the United Nations' Sustainable Development Goal 14: Life Below Water [8]. This goal is committed to conserving and sustainably using the world's oceans, seas, and marine resources, and our efforts form an integral part of this global endeavor. Through employing such technologies, we can enhance the sustainable management and protection of marine ecosystems, minimizing bycatch and maintaining biodiversity. In this paper, our primary objective is to examine the utilization of object tracking techniques to enhance the accuracy and efficiency of agent counting estimation.

1.2 Related Work

Previous work on this specific problem [2] involved using an object detection algorithm called RetinaNet, fine-tuned on a custom dataset, to detect the fish species and position and applying regression on the number of detections filtered by probability. In the paper, they filter the object detections on a confidence of 0.47 since this is the average of the optimal probability for each species. They achieved a mean average precision of 0.845 on a test set of 918 images, and an R-squared score of 0.74, 0.62, and 0.84 for blue whiting, herring, and mackerel, respectively, which shows that the technique is viable. Their work used images from 20 stations, which are a subset of the station data we are using.

Westergerling et al. [12] made a comparison between the Deep Vision camera system, acoustic data and catch results. They are responsible for the label data, gathered from three different stations used in this paper.

Scantrol Deep Vision [10] developed "Deep Vision", an innovative tool for marine research. This system utilizes an underwater camera system that allows for fish sampling and analysis without the need for physical catch, promoting sustainability and efficiency in marine research. The Deep Vision camera system was used to create the image data used in our thesis.

RetinaNet [5] is a notable contribution to the field of object detection. This model utilizes the Focal Loss function, specifically designed to address the class imbalance problem in object detection. The model architecture comprises a backbone network, typically a deep ResNet, and two task-specific sub-networks for classifying objects and regressing their bounding boxes. RetinaNet has been widely recognized for its effectiveness in detecting objects across a wide range of scales and its strong performance on benchmark datasets. RetinaNet is presented further in section 2.2.5.

1.3 Thesis Outline

Chapter 2 - Background Theory
This chapter describes relevant theory behind the methods used to solve the objective of the thesis

Chapter 3 - Method
This chapter introduces the dataset and describes the method and the detail of its implementation.

Chapter 4 - Results
This chapter presents the results found from the experiments.

Chapter 5 - Discussion
This chapter discusses the findings in Chapter 4. Suggestions of further work is also presented.

Chapter 6 - Conclusion
This chapter gives a summary and conclusion for the thesis.

Chapter 2

Background Theory

This chapter provides an overview of the key concepts and methodologies underpinning this work, establishing a theoretical background that aids in comprehending the analysis and findings presented in the subsequent chapters.

2.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is a computational structure that is inspired by and shares some resemblance with the biological structure and function of Biological neural networks.

2.1.1 Biological Neural Networks

Biological neural networks are composed of many interconnected biological neurons. These biological neurons can be found in the brain and nervous system and usually consist of three parts: The Soma, the dendrites, and the axon. The soma, known as the cell body, is responsible for the health of the cell and processes the information received from the dendrites. The dendrites are branching structures whose job is to transmit information from other cells to the Soma, where it can be processed. The Axon is a long, thin structure that transmits information from the Soma to other cells. In other words, dendrites send information to the Soma, the Soma processes the data, and the Axon transmits the processed data to other biological neurons.

2.1.2 Non-biological neural networks

Non-biological neurons (we will call them "neurons") are similarly built where they have an input, an activation function, and an output. The neuron input, typically a vector of float numbers, are dot-producted with a vector of trainable float numbers called "weights". We add another trainable float number, the "bias", to the dot product, and that gives us the input to the activation function. Usually referred to as the "activation input" or z in a mathematical context. The activation function is a non-linear function that ensures that the neural network can model non-linear relationships between input and output. Activation functions also commonly limit the values of the output between two values, such as the sigmoid activation Function 2.1 that maps the activation input to a value between 0 and 1. This value is the output of the neuron and is often referred to as a in a mathematical context.

$$\begin{aligned} z &= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b \\ a &= \sigma(z) = \frac{1}{1 + e^{-z}} \end{aligned} \tag{2.1}$$

Equation 2.1: Neuron Computation with Sigmoid Activation Function where each x_i value is a value from the neuron's input vector, each w_i is a value from the weight vector, b is the bias, σ is the sigmoid activation function, and a is the neuron output.

When neurons are stacked and layered as a network they can be trained to calculate complex relationships between the network input and output. In a basic feed forward NN, demonstrated in Figure 2.1, the model is split into three parts: input layer, hidden layer and output layer. The input layer is where the input data is inserted before it propagates through the hidden layer. The hidden layer is highly flexible, it can contain any number of layers, each of which can contain any number of neurons. Through effective training, these layers extract task-relevant information and transmit it to the output layer. The output layer applies an appropriate activation function to compute the final output.

2.1.3 Supervised Learning

Machine learning can generally be split into three branches: Supervised learning, Unsupervised learning and Reinforcement learning. In this thesis, we visit the topic of supervised learning which means that the neural network is trained on labeled data.

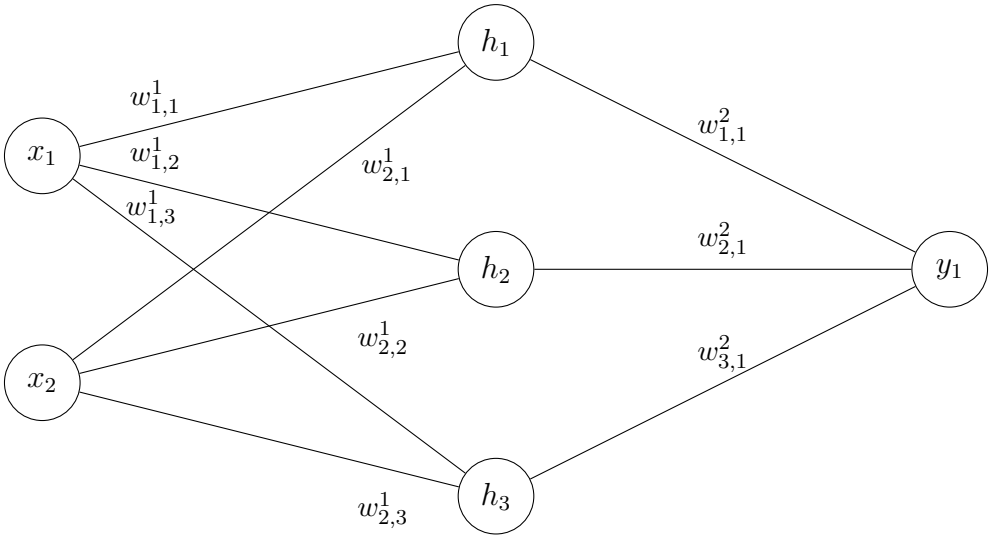

Figure 2.1: Example of a simple neural network.

2.1.4 Backpropagation

Backpropagation is a supervised learning technique that updates the weights and biases of the network. Its aim is to minimize the error between the network's predicted output and the actual data, with the error calculated by the loss function.

2.1.5 Loss functions

The loss function serves as a measure of the discrepancy between the predicted output of a network and the actual label data. A typical instance of a loss function is the Mean Absolute Error, which computes the average magnitude of error across multiple predictions.

2.2 Object Detection

Object detection, which identifies and locates objects within images or videos, is a cornerstone of computer vision tasks. This section probes into key models and concepts like Convolutional Neural Networks, backbone networks, Feature Pyramid Networks, Focal Loss, and the RetinaNet model, all of which significantly contribute to the field.

2.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN), a specialized type of ANN, are notably efficient at processing data with grid-like topology, such as images. CNNs learn hierarchical spatial features from input data using convolutional layers. Unlike traditional ANN layers, where each neuron connects to every output of the preceding layer, each neuron in a convolutional layer only processes data from a specific block of the previous layer. This block of data undergoes transformation by trainable filters, which convert it into feature maps. Subsequently, these feature maps pass through nonlinear activation functions to produce the final features. Finally, these processed features feed into one or more fully connected layers to execute classification or regression tasks. With their proven efficacy, convolutional layers have become a crucial component in numerous state-of-the-art architectures. [4]

2.2.2 Backbone networks

A backbone network refers to the part of the neural network that extracts the higher level features of the input data, often consisting of several layers of convolutions, pooling and activation functions. It is common to use backbone networks that have been pre-trained on large datasets so that it returns generic features that can be fine-tuned to a specific task on a smaller dataset. Using a pre-trained backbone network often achieves a better result than training a network from scratch, especially if the task specific dataset is small.

2.2.3 Feature Pyramid Networks

Feature Pyramid Networks (FPN) stand as a widely adopted neural network architecture, predominantly employed for object detection tasks. The architecture leverages convolutional layers and max pooling to create a series of feature maps, each with a different resolution. It then uses upsampling operations to merge these feature maps, generating a "pyramid" of features that encapsulates object information at various scales. This multi-scale representation significantly enhances the effectiveness of object detection, since it accounts for variability in object sizes and contexts.

2.2.4 Focal Loss

Focal Loss [6] is a loss function that addresses the problem of class imbalance in machine learning tasks. Focal Loss places emphasis on hard-to-classify examples by down-weighting easy examples that contribute to the majority of the loss. This is particularly useful in scenarios where there is a significant class imbalance with a large number of easy examples and a relatively small number of hard examples. By reducing the contribution of easy examples to the loss, Focal Loss enables the model to focus more on the hard examples, leading to improved accuracy on the difficult examples.

Object detection problems often have a skewed class balance, with more easy examples than hard ones, and can therefore benefit from using Focal Loss.

2.2.5 RetinaNet

RetinaNet, an object detection architecture, incorporates elements of ResNet, FPN, and Focal Loss [5]. Serving as a backbone network in numerous object detection tasks, ResNet is a widely used CNN architecture. By employing skip connections, it effectively addresses the vanishing gradient problem, thus facilitating superior accuracy and simpler optimization. FPN is employed to handle objects at different scales effectively. FPN combines features from different layers of the backbone network such that the network can detect objects of varying sizes. Lastly, the loss function Focal Loss is applied to effectively handle unbalanced data by prioritizing more difficult examples.

The output of the model is a set of bounding boxes where each bounding box has four numbers representing the top left corner (x,y), the width (w) and the height (h). Additionally, each box has a class label and a confidence value. Using a combination of ResNet, FPN and Focal Loss, RetinaNet has achieved great results on benchmark data sets and has been a popular choice for object detection tasks.

2.2.6 Object Tracking

Object tracking extends the concept of object detection by not only detecting objects in each frame but also maintaining the identity of those objects across multiple frames in a video sequence.

To create the fish tracks we use the Munkers algorithm, which lets us find the best fitting boxes from frame to frame based solely on the box coordinates and sizes. A step by step description of the Munkers algorithm is provided in Figure 2.2.

1. Create an $n \times n$ cost matrix, where n is the number of rows and columns representing the number of sources and destinations, respectively.
2. Subtract the minimum value in each row from all the elements in that row.
3. Subtract the minimum value in each column from all the elements in that column.
4. Find the minimum number of lines (horizontal and vertical) needed to cover all the zeros in the matrix.
5. If the number of lines is equal to n, go to step 6. Otherwise, go to step 7.
6. Assign zeros to the uncovered elements in the matrix in such a way that no two zeros are in the same row or column. Go to step 9.
7. Determine the smallest uncovered element in the matrix and subtract it from all the uncovered elements. Add it to all the elements covered by two lines.
8. Go to step 4.
9. The assignments of sources to destinations are the elements in the matrix with the value 0.

Figure 2.2: Steps of the Munkers Algorithm

2.3 Linear Regression

Linear regression is a widely used statistical method for establishing the relationship between a dependent variable and one or more independent variables or covariates. It assumes a linear relationship between these variables, thereby enabling the prediction of the dependent variable based on a linear combination of the independent variables and covariates.

A generalized linear regression model [11], considering multiple independent variables and covariates, can be represented mathematically as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k + \varepsilon \tag{2.2}$$

where Y is the dependent variable or the response variable, $X_1, X_2, ..., X_k$ are the independent variables or covariates. $\beta_0, \beta_1, ..., \beta_k$ are the coefficients, where β_0 is the intercept (the value of Y when all X are 0) and $\beta_1, ..., \beta_k$ represent the effect, or slope,

of each variable (the change in Y for each unit change in the respective X_i). Finally, ε is the error term, accounting for the variability in Y that cannot be explained by the X variables.

Linear regression aims to estimate the coefficients $\beta_0, \beta_1, ..., \beta_k$ that minimize the sum of squared discrepancies between the predicted and actual values of the response variable.

2.3.1 R-squared

How well a linear regression model fits the data is often assessed using the coefficient of determination, also known as R-squared, presented in Equation 2.3. R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables. It ranges between 0 and 1, where a value closer to 1 indicates a better fit of the model to the data.

$$\begin{aligned} R^2 &= 1 - \frac{\text{SSR}}{\text{SST}} \\ \text{SSR} &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ \text{SST} &= \sum_{i=1}^{n} (y_i - \bar{y})^2 \end{aligned} \tag{2.3}$$

Equation 2.3: Formula for R-squared, where SSR is the Sum of Squared Residuals and SST is the total sum of squares. y represents the actual value of the dependent variable, $\hat{y}$ represents the predicted value for the dependent variable, and $\bar{y}$ represents the mean of the dependent variable.

2.3.2 Mean Absolute Error

MAE is a straightforward and interpretable metric used for quantifying prediction accuracy in regression analysis. It calculates the average of the absolute differences between predicted and actual values, providing a direct measure of average prediction error magnitude. The formula for MAE is defined as:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2.4}$$

where n is the total number of data points, y_i represents the actual value, $\hat{y}_i$ signifies the predicted value, and $|...|$ denotes the absolute value function.

Chapter 3

Method

This section describes the methods applied in the project. The dataset is introduced as well as the implementation of tracking, visualization, how we test, and a description of our use of Regression.

3.1 Dataset

The data used in this thesis are split into four parts: images, object detections on the images, catch data and labeled data.

In total, we have catch data from 51 trawl sessions. 22 of the trawl sessions are from 2018 with a frame rate of 5 Frames Per Second (FPS) and 29 sessions are from 2022 with a frame rate of 10 FPS. Our labeled data only covers three stations and they are all from the 2018 dataset [3], with catch data.

3.1.1 Image collection and catch count

The images are collected by trawls across the northern Atlantic Ocean. During the trawl, the fish are funneled through a Deep Vision camera system [9] that takes pictures throughout the entire trolling period with two cameras at a frame rate of 5 to 10 FPS. These pictures make up the picture dataset. To avoid adding another dimension to our experiments we elected to only use the images from the left camera, and object detections affiliated to them.

After going through the Deep Vision camera system, the fish are caught in the net and loaded onto the trawler where a random portion of the fish is counted, weighed and identified, and this distribution is scaled up based on the total catch weight. Therefore, if 10% of the total weight is counted and identified as 20 herring and 30 mackerel, it would imply that 200 herring and 300 mackerel are noted as the catch data for that trolling session.

3.1.2 Object detections

In this study, we leveraged the object detection dataset developed from the trawl images in an earlier study [2] on automatic fish species identification.

The earlier study laid a substantial groundwork by utilizing a fine-tuned RetinaNet model to identify and quantify populations of blue whiting, herring, mackerel, and mesopelagic fishes from underwater imagery. The dataset created from the trawl images involved detailed annotations denoting the species/group name and bounding box coordinates for each fish. To train the model they used a set of labeled images, these images were systematically segregated into three different sets: training, validation, and testing. The training phase was enhanced by the use of synthetic data, that were generated via image transformations and placement of fish crops on the background images.

The performance of the trained model was evaluated through mean Average Precision (mAP), a reliable metric that quantifies the area under the precision-recall curve for different confidence thresholds. Subsequently, the model with the highest mAP on the validation set was used to evaluate the test set. The trained model was then converted into an inference model to make predictions on unannotated data. The confidence score threshold for these predictions was set based on maximizing the F1 score. This model was used to generate the 2018 object detection data, and was subsequently retrained for the 2022 predictions, with a semi-supervised approach.

The output of the model is Comma Separated Values (CSV) files that denote bounding boxes that are meant to enclose the fish as well as a species, datetime for the picture, and the algorithm's confidence that the bounding box is accurate. These CSV files make up the object detection data.

3.1.3 Label data

The labeled data is created by field experts going through the data image by image, labeling the data for: species, number of frames each fish is visible to the camera, and marking the front and rear end of the fish for the frame it is most visible in. If the fish is never fully visible, the two points will be put near the center of the fish. If the fish is curved, additional points are added along the body such that one could draw a curve through the points to follow the fish. The label data contains more species than those on which the object detection algorithm is trained. However, in this project, these extra species will be ignored.

Labeled data is time consuming to make and therefore we only have label data from three different stations.

3.2 Tracking

In our tracking implementation we apply the Hungarian algorithm, also called the Munkres algorithm, to temporally connect the object detections from frame to frame, creating several tracks. A track refers to a sequence of object detections that are identified as the same object moving from frame to frame. The Hungarian algorithm is a well-established optimization algorithm widely used for solving the assignment problem. In our case, the assignment problem involves finding the best association between tracks and detections based on their distance from each other.

In our application of the Hungarian algorithm, we use a library which can be found in the stracks repository developed by Malde [7]. The implementation leverages a tailored distance metric that accounts for both the spatial separation between detected objects and the difference in bounding box sizes. The "scale" parameter plays a crucial role in calibrating the impact of size and positional variances between two bounding boxes on the overall distance computation.

When a larger scale value is set, the decay rate for size and positional differences is slower. This implies that higher scores can still be achieved even with larger discrepancies in bounding box dimensions. This configuration, therefore, reduces the sensitivity of the metric to size variations.

On the other hand, a smaller scale parameter introduces a quicker decay rate, resulting in lower scores for larger size and positional differences. This heightens the sensitivity of the metric to size variations. Consequently, the scale parameter provides a tunable mechanism to balance the influence of spatial positioning and size in our object matching process.

The tracks are generated several times with different values for the scale parameter. We made tracks for the scale values: 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 9, 10, 11, 12, 13, 14, 15, 16

After generating the tracks, we employed a weighted majority vote approach to assign a class label to each track. This process involves determining the species present in the majority of frames within a track while considering the certainty associated with each classification decision. Detections with high confidence will have higher influence on the track label. The objective is to mitigate the influence of misidentified object detections on the overall track classification outcome.

3.3 Visualization

The data is displayed with Matplotlib for visualization and altered using Numpy and Pandas libraries in Python.

3.4 Measuring performance

To assess the performance of the model setup given different parameters, we compared it to the catch data and the labeled data. The labeled data should be much more accurate than the catch data, but we use both data sets since we do not have a lot of labeled data compared to the amount of catch data.

Before creating the tracks we filter the object detections on confidence 0.47 to reduce the number of redundant or bad tracks. The filter value 0.47 is based on a previous paper's findings [2].

After creating several tracks for all stations, We use the generated tracks to create regressions with different scale values. For each regression we have two scale values, one

for 2018 data and another for 2022 data. This is because the data from 2018 is 5 FPS while the data from 2022 is 10 FPS, meaning that the scale parameter for 2022 data should be half of the 2018 data. We use R-squared, MAE and WAR, described in section 3.4.2, to assess the quality of the regression models and select the best regression models and scale values. We also compare our models to the R-squared performance from a previous works [2] on the 20-station dataset used in their paper.

In the evaluation phase of our selected model, we employ the three stations with labels. The tracks, produced with the determined scale parameter, are utilized to generate regression values. These values are then evaluated based on R-squared, MAE and WAR scores. Furthermore, we apply the methodology described in Section 4.4.2 and analyze the resulting data. This evaluation approach not only allows us to measure the model's accuracy but also aids in our understanding of the data characteristics.

We also train each regression model on the 20-station dataset used by previous works [2] to better compare the methods and their performance.

The subsections below describe the different metrics used to compare the data.

3.4.1 Connecting tracks to labels

Throughout many of our experiments we compare track counts, catch counts and label counts. The problem with this is that comparing the total number of agents does not necessarily give information about the individual tracks and their correctness. To get some information about this we use the positions of the labels and track boxes to connect labels to tracks and vice versa.

Each label includes the number of frames an agent is present in, as well as two or more points associated with a specific frame where the agent is most visible. The two points represent the nose and tail of the fish. In some cases, when the fish is curved, there are several points following the shape of the fish. In the case where there are no frames where the fish is entirely visible, two points are placed somewhere close to each other near the center of the fish.

To connect a track and a label, we compute the average of the label points to determine the approximate center of the fish. Subsequently, we verify whether this center point falls within a prediction box of a track. If the center point is within a single box, we select the track associated with that box. In cases where the center point falls within multiple

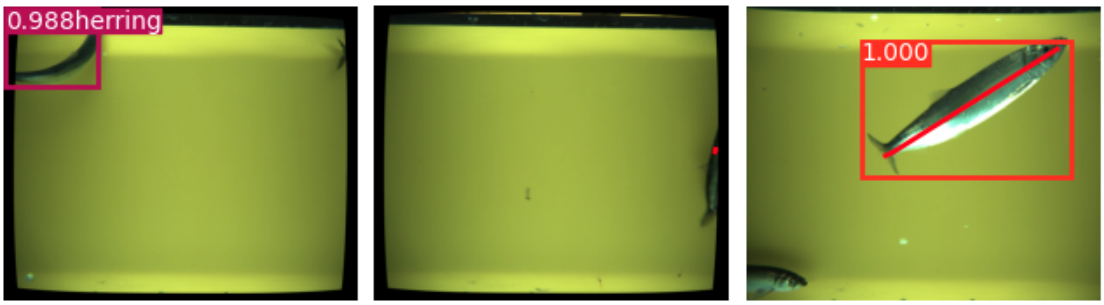

Figure 3.1: Examples of Track Without Label (left), Label Without Track (middle) and track matched with label (right). The box represents the current frames object detection from the track. The red line/point are label data.

boxes, we choose the box that the point is closest to the middle of. In the case where the point does not land inside any boxes it remains trackless. Comparing the species predicted and the species labeled, we can plot the real species vs the predicted species.

After connecting tracks to labels we put the data into four categories that are used for analysis:

Correct identifications : Tracks that are connected to a label of the same species.

Incorrect identifications : Tracks that are connected to a label of a different species

Tracks Without Labels (TWL) : Tracks that have no unused label points inside any of the track boxes

Labels Without Tracks (LWT) : Label points that are not covered by any tracks.

3.4.2 Weighted Average Ratio of tracks and catch data

This study introduces a modified approach to accurately quantify the discrepancy between the predicted count and the catch data, while factoring in the size of the data, Weighted Average Ratio (WAR). To calculate the average ratio of predicted count versus catch count across a species we make sure stations with a higher predicted count or catch-count have a larger impact on the error than stations with lower numbers. To do this we create a weight for each station, per species, that is applied to the ratio of predicted count divided by catch count for that station. +1 is added to the denominator to avoid division by 0 and to the numerator to avoid bias toward the catch count.

The complete Formula 3.1 determines $\overline{w(s)}$, representing the weighted average ratio of the predicted count to the catch data for species s. Here, the weights are influenced by the magnitude of the predictions and catch data. For example, consider station x with $p_{h,x} = 100$ and $c_{h,x} = 150$. The impact of this station on the weighted average herring ratio is significantly higher than that of another station y with $p_{h,y} = 10$ and $c_{h,y} = 15$. Additionally, a station z with $p_{h,z} = 0$ and $c_{h,z} = 0$ would not influence the ratio.

$$
\begin{aligned}
s &\in [h, bw, m] \\
\overline{w(s)} &= \frac{\sum_{i=1}^{n} w_{s,i} \left(\frac{p_{s,i}+1}{c_{s,i}+1} \right)}{n} \\
w_{s,i} &= \frac{z_{s,i}}{\frac{1}{n} \sum_{j=1}^{n} z_{s,j}} \\
z_{s,i} &= p_{s,i} + c_{s,i}
\end{aligned}
\tag{3.1}
$$

Equation 3.1: The Weighted Average Ratio, where $p_{s,i}$ is nr of predicted occurrences, and $c_{s,i}$ is the catch nr, for species s on station i. $\overline{w(s)}$ is the Weighted Average Ratio between the number of predicted tracks and catch count for species s.

3.4.3 R-squared

To measure the performance of the regression model we use the R-squared metric, also called the Coefficient of determination, which is a statistical metric that provides an indication of how well a regression model fits the observed data.

3.5 Regression

To improve the accuracy of our species count estimates, we carry out three primary linear regression analyses, along with one baseline analysis, described in Table 3.1. The regression lines are fitted to the catch data and corresponding track counts, both of which are logarithmically transformed, for each species in the catch data. We assume the number of tracks for a species is linearly correlated to the catch data, and therefore chose linear regression for this task.

The "Baseline model" is a regression model on the track count of a species and that species' catch data, and it is the only model that does not filter tracks on depth before fitting. The other models filter their object detections based on the likelihood of finding a particular species at certain depths before performing regressions. This probability is determined using data from Fishbase [1]. For all species other than blue whiting, anything at or above 0m was removed. blue whiting were filtered on 150m.

Of the three primary regressions: the first approach, referred to as the "Reduced model", uses the logarithm of the track count specific to a species to predict that species' catch data. Conversely, the "Full model" leverages the logarithm of track counts from all species to predict the catch quantity of a single species. Lastly, the "Full model-sum version" employs the logarithm of the track count of the species being predicted, combined with the logarithm of the sum of the track counts of all other species to predict the catch quantity of the species being predicted.

Model	**Depth Filter**	**Input Variables**	**Dependent Variable**
Baseline Model	False	$Log(T_X)$	$Log(C_X)$
Reduced Model	True	$Log(T_X)$	$Log(C_X)$
Full Model	True	$Log(T_{bw}), Log(T_h),$ $Log(T_{mac}), Log(T_{meso})$	$Log(C_X)$
Full Model-Sum Version	True	$Log(T_X),$ $\sum_{s \neq X} Log(T_s)$	$Log(C_X)$

Table 3.1: Summary of the four regression models. T_X is the track count, and C_X is the catch count, for species X. bw=blue whiting, h=herring, mac=mackerel and $meso$=mesopelagic

Furthermore, this method enables us to compute the R-squared metric, providing insight into the performance of the models.

In our regression models, each data point represents an individual station. We use these trained regression models to predict both label data form the labeled stations, and catch data from the same stations. This is not a proper test set, since it is only six data-points, but we use the test to gain insight on the models. The models predict new data by using the slope and intercept derived from the regression to map the number of tracks to a more accurate estimation. To eliminate potential bias, none of the stations possessing label-data were included in the training of any of the models that were subsequently tested. R-squared, MAE and WAR metrics are utilized to assess the performance of the models against the label and catch data.

Chapter 4

Results

This chapter presents the results of our experiments, as well as reference results from related works.

4.1 Reference results

Previous work on this problem [2] created two types of linear regression models for each species. Their Reduced model predicted the number of object detections on a species based on the catch data for that species Their Full model predicted the number of object detections based on the species' catch and the sum of catches from the other two species as covariates.

For all species the Full model performed best. The models showed that a species' count was increased when there was a high count in other species. To measure the performance they calculated the R-squared metric for each model. The R-squared values for the Full model were: 0.74, 0.62, and 0.84 for blue whiting, herring, and mackerel, respectively.

4.2 Depth filter

Depth-based filtering was implemented with the objective of minimizing a species' misclassifications. As shown in Figure 4.2, blue whiting benefits the most from this approach, seeing an improvement in the R-squared metric across all scales. Herring only shows a

modest improvement for certain scales, while performance of Mackerel remains relatively unchanged, with negligible variations that oscillate between minor improvements and slight decay depending on the scale.

Unless mentioned otherwise, subsequent plots and experiments with track data is filtered on depth.

4.3 Tracking performance on catch data

4.3.1 MAE and WAR

When comparing catch data to the track count for each species, MAE shows a strong preference for low-to-medium scale-values between 0.5 and 4, with minor variations for each species. The WAR metric mirrors this tendency.

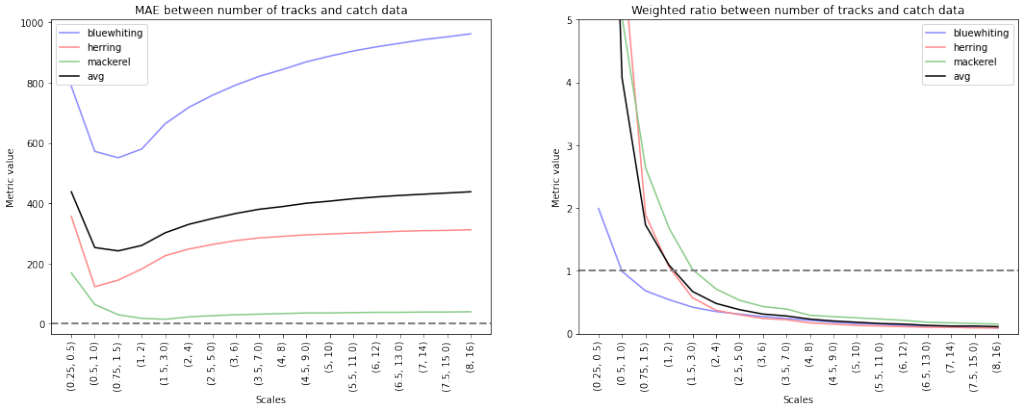

Figure 4.1: MAE and WAR scores across the tested pairs of scale values for each species.

4.3.2 Regression

The regression analysis serves as a statistical tool to evaluate the relationship between catch data and the number of tracks for every pair of scale values. As evidenced by Figure 4.2, all species demonstrate unique behaviors and performance is largely dependent on model. In the performed experiments, the Full model consistently beats the other models for all species except blue whiting where, at higher scales, it is outperformed by the Reduced model.

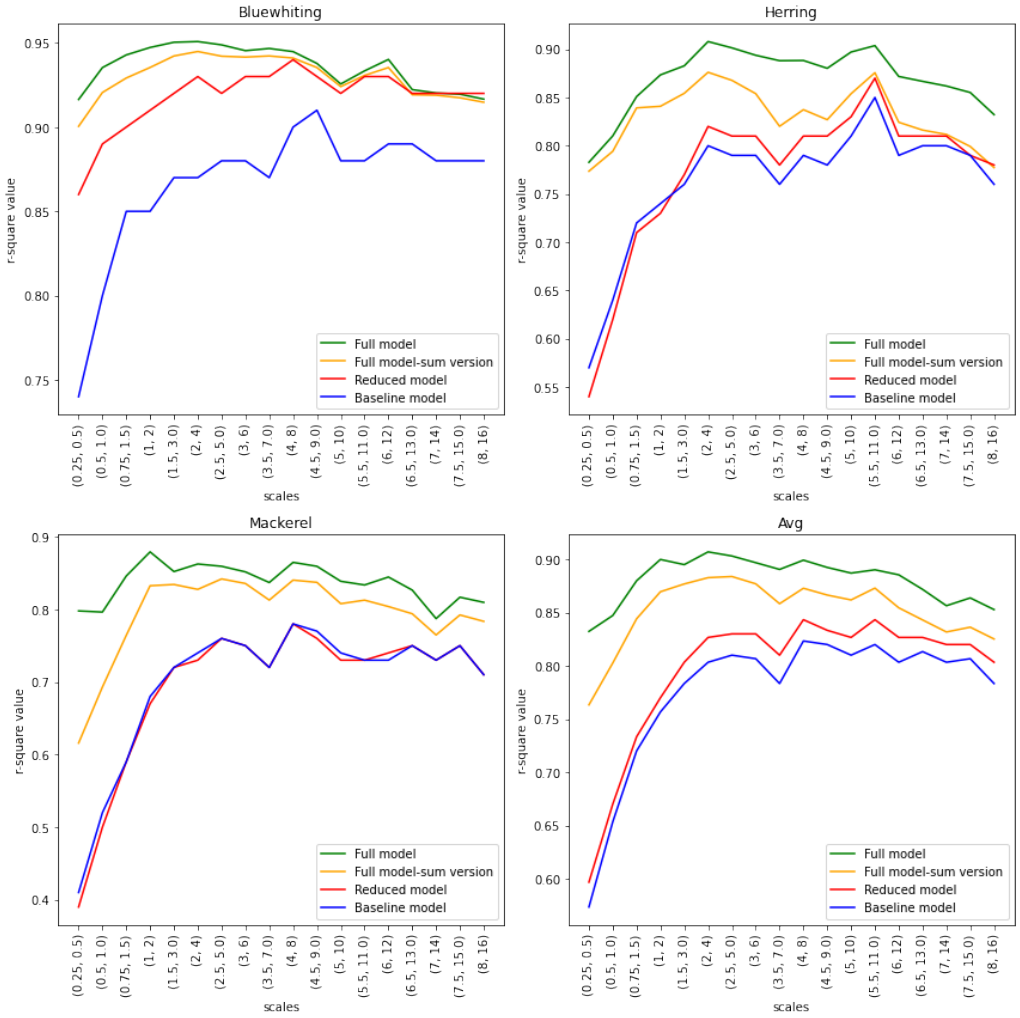

Figure 4.2: This plot contrasts the R-squared values derived from the depth-filtered and non-filtered reduced models, as well as the depth-filtered full model, for each individual species' regression analysis.

The scale values that optimize the R-squared values is the same for blue whiting and herring at (2,4), but the mackerel regression maximizes its R-squared value at scales (1,2). The best scale value for the average of the three species is (2,4) with an average R-squared value of 0.91.

The regression analyses for all three species yielded impressive results, with high R-squared values of 0.95, 0.91, and 0.88, as seen in Table 4.1. A comparison of our approach with the prior work, presented in Table 4.2, reveals a distinct improvement across all species for the Full model. This enhancement is especially evident for herring, where our methodology exhibits significant advancement when tested on the same 20-station dataset used in the previous study. Our other two models are outperformed by previous

work when it comes to mackerel, but still result in an average improvement when taking all species into account.

Species	Scale	R-Squared	Intercept	Slope
Blue whiting	(2, 4)	0.951	0.540095	1.110102
Herring	(2, 4)	0.908	0.193570	1.363425
Mackerel	(1, 2)	0.879	0.753532	1.241854

Table 4.1: Best R-square value for each species for Full model

Species	Reduced model	Full model	Full model-sum version	object detection + regression
Blue whiting	0.81	0.84	0.82	0.74
Herring	0.91	0.96	0.93	0.62
Mackerel	0.76	0.87	0.82	0.84

Table 4.2: Our work versus previous work compared with R-Squared results on the 20-station dataset used in previous work. Our regression model was trained with scale 4 for all species.

4.4 Tracking performance on label data

4.4.1 Regression

Based on the results given in Section 4.3, we chose the value 4 for our scale parameter when testing performance, and only moved forward with the two best performing models: The Full model and the Full model-sum version. Note that we only chose one scale parameter since all the label data is 5 FPS.

We applied our regression models to the track counts and compared the results with the label data and corresponding catch data from each labeled station. For each species, we evaluated the results using the R-squared, MAE, and WAR scores. While the MAE values for blue whiting and mackerel displayed a relative similarity between the catch and label data, the MAE for herring was significantly higher in the label data as compared to the catch data. The WAR values indicate an over-prediction for both blue whiting and mackerel, particularly in instances of larger quantities in catch and label data. We also see that, when predicting the herring catch, the models were relatively accurate; however, when comparing label-count, they under-predict quite badly.

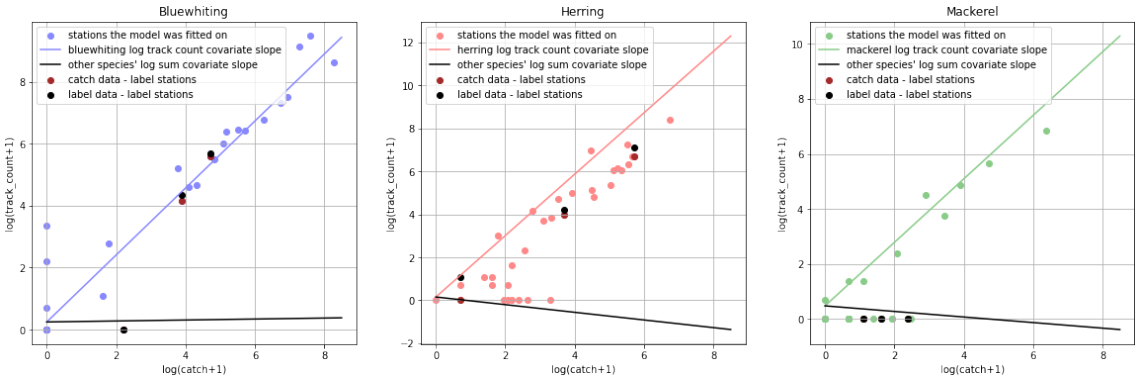

Figure 4.3: Full model-sum version fitted on catch data and tracks using scales 2 and 4 for 2022 and 2018 data, respectively. The colored slope represents the slope multiplied to the log of the track count of the species being predicted. The black slope represents the slope multiplied to the log of the sum of track counts of other species.

Both R-squared values for blue whiting on catch and label data are about the same value for each model. The R-squared metric for herring performs much better on the catch data than on the label data. Mackerel has 0 as its R-squared value for both catch and label data, meaning that the model does not explain any of the response data around its mean for the test data.

The regression lines plotted in two dimensions in Figure 4.3 give some intuision of the difference betweeen the catch and label data of the labeled stations compared to the other stations.

Species	MAE-C	MAE-L	WAR-C	WAR-L	R-squared-C	R-squared-L
M1-blue whiting	18.33	8.67	1.40	1.26	0.55	0.57
M1-herring	11.33	152.00	0.97	0.65	0.91	0.66
M1-mackerel	2.33	2.33	4.57	4.57	0.00	0.00
M2-blue whiting	29.67	34.00	1.39	1.25	0.51	0.54
M2-herring	49.33	190.00	0.84	0.56	0.92	0.66
M2-mackerel	2.33	2.33	5.14	5.14	0.00	0.00

Table 4.3: MAE, WAR and R-squared scores between prediction on track count versus catch data and label data, based only on the three labeled stations. C=catch, L=label, M1=Full model, M2=Full model-sum version

4.4.2 Connecting tracks to labels

Using the labeled data we mapped the tracks to labels, giving us an indication on how the tracking algorithm performed on the different species. Figure 4.4 shows that the most common species to not be tracked or not be correctly identified are the mesopelagic species. Additionally, a considerable proportion of the unused tracks are also attributed to this species. Mesopelagic fishes are also commonly misidentified as blue whiting or herring, and vice versa, as shown in Figure 4.5. None of the three labeled stations had any mackerel in the label or catch data, so we have no data on what mackerel are misidentified as.

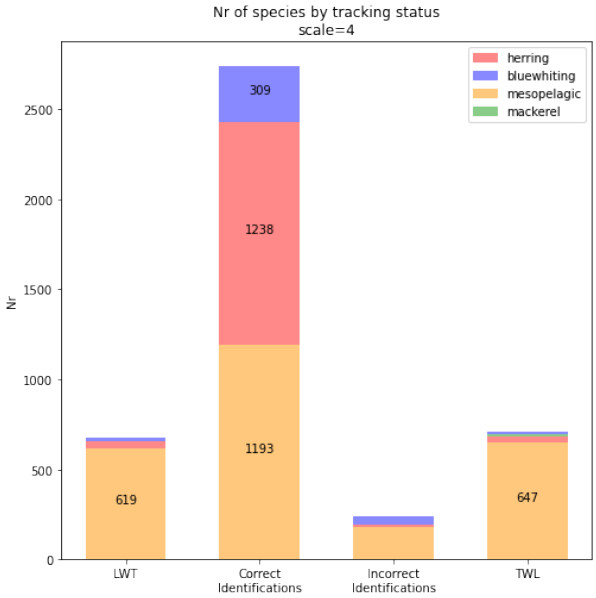

Figure 4.4: Species distribution based on the track-to-label mapping.

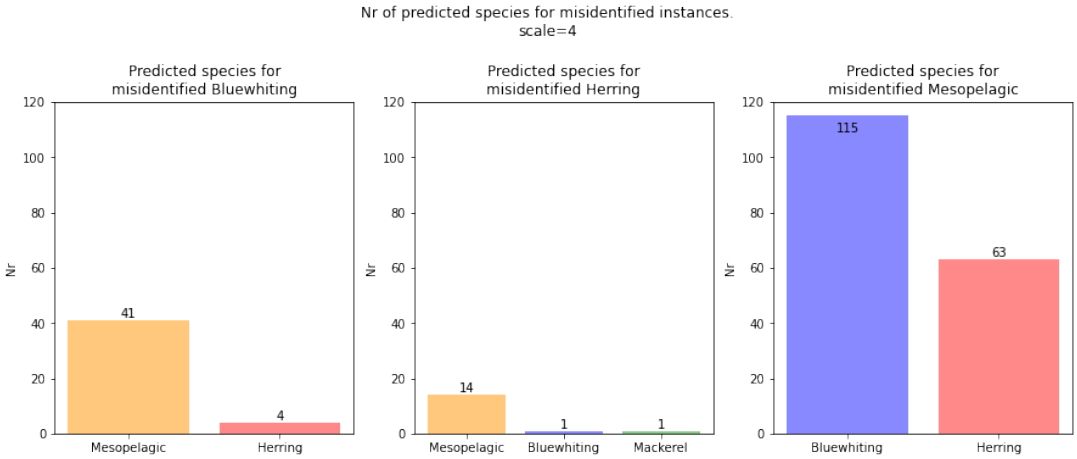

Figure 4.5: Misidentified species and what they were identified as.

Chapter 5

Discussion

This chapter discusses the results and addresses limitations of the methods applied in the experiments. Previous work on this problem is used as baseline for comparison. Suggestions for future work is also presented.

5.1 Limitations of the Dataset

The object detection dataset was generated by a fine-tuned RetinaNet model with a mean average precision of 0.845. As acknowledged in the original study developing the model [2], the technique is vulnerable to misidentifications. This becomes a hurdle when basing the object tracker on the misclassified object detections. Addressing this problem effectively is difficult without resorting to training a new object detector.

Throughout the study we also found that some of the depth data, provided by the depth sensor during the trawl, was inaccurate, implying that some of the depth filtering might be inaccurate too.

5.2 MAE and WAR Metrics

When comparing track count to catch data for each species, the performance of the MAE and WAR metrics showed an increase in performance between the scale values 0.5 and 4, and a drastic decrease in values outside of this range. To understand the reason

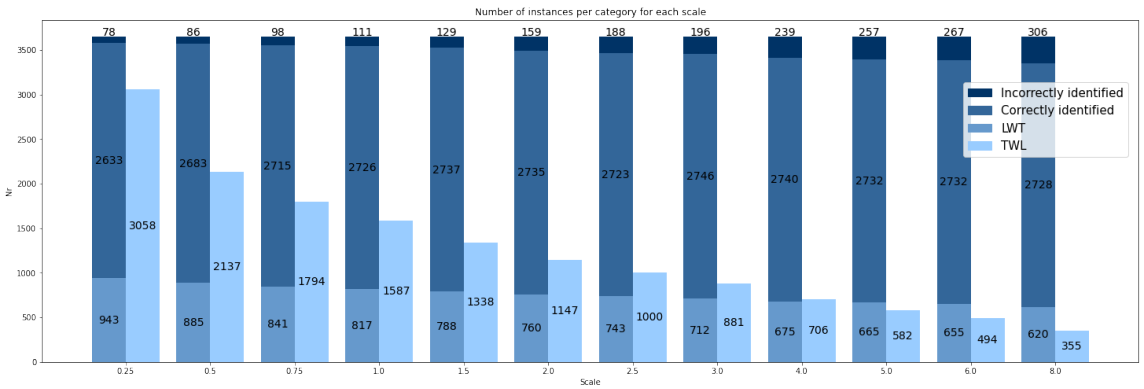

Figure 5.1: Number of instances per category generated by the track-label mapping, for each scale.

behind this, we examined the connection of tracks to labels across all scale values in the label data. Our findings revealed that as the scale increases, both the number of Labels Without Tracks (LWT) and Tracks Without Labels (TWL) decrease. This leads to a greater number of labels being connected to tracks, but at the cost of lower relative accuracy of predicted species, for the tracks connected to a label.

The observed behavior is consistent with our understanding of how scale values operate in the context of object detection. A higher scale value increases the likelihood that an individual object detection will link with another, rather than forming an isolated track. As the scale value becomes more permissive, it allows for the connection of object detections with greater differences, which can subsequently result in excessively long tracks.

In scenarios where multiple species are present, this leniency can lead to an over-representation of the most common species. The reason for this stems from our approach to species assignment. We apply a weighted voting system to determine the species of a track, and when tracks become overly extended due to a higher scale value, the most common species can sometimes dominate the vote, thereby skewing the representation towards the more abundant species.

Figure 5.1 clearly shows how TWL decreases as the scale value is increased. Interestingly, we see that LWT and TWL are very similar around scale 4. When these two values are the same, the total number of tracks will be equal to the label count, even if the tracks themselves might be incorrect. However, this observation positively influences the MAE and WAR values only when the species-distribution of TWL mirrors that of LWT. To explore this possibility, we plotted both LWT and TWL values for each species in Figure 5.2. Our analysis revealed that around scale values ranging from 2 to 5 (varying

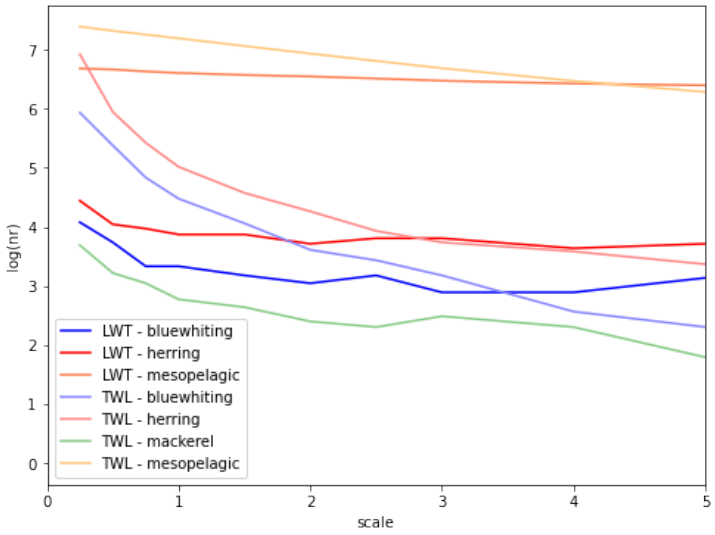

Figure 5.2: Plot from the track-label mapping showing log of the number of Labels Without Tracks (LWT) and Tracks Without Labels (TWL) for each species over some of the scales.

by species), the relative count of instances per species closely aligns. This observation potentially explains the superior performance of the MAE and WAR values around scale values 0.5 and 4 within the catch data, show in Section 4.3.1, but this behaviour could also be a bias in the label data.

5.3 Regression

5.3.1 Regression performance on catch data

The results of the regression models indicated that the Full model outperformed the others for every species. This suggests that the counts of other species play a significant role in predicting any specific species. This observation aligns with our understanding of the misclassifications made by the system, particularly with respect to the mesopelagic species, which are most commonly predicted. In fact, within the model utilizing covariates, the slope for the mesopelagic track count was negative for all blue whiting and herring models, as well as most of the mackerel models across the tested scale combinations.

The slight negative correlation between the number of mesopelagic fishes and other species was also displayed in Figure 4.5, where many blue whiting and herring were

misidentified as mesopelagic fishes. However, even more instances of mesopelagic species were misidentified as blue whiting and herring. This suggests that a high volume of mesopelagic species passing through the Deep Vision camera system can lead to an over-count of the other species.

5.3.2 Comparing to previous work

One of the main reasons for which our setup performs better than previous work is likely due to the massive reduction in difference between the catch data and the covariates, done by the tracker. The 20-station dataset was reduced from ≈352k object detections down to ≈30k tracks. Even though the tracks are imperfect, they provide more information than they remove.

A fish can stay in-frame for anything from 1 to 100+ frames, demonstrated in Listing B.1. A fish that stays in-frame for a long time can result in inaccuracies in a non-tracking algorithm. For instance, if a herring remains in frame for 100 frames, a model based solely on object detection and regression might overestimate this as 10 fish, given that the average herring stays in frame for 10 frames. However, our model would likely register this as less than 10 tracks, resulting in a more accurate prediction.

Another important factor that counts positively for the Full model and Full model-sum version's performance, is their inclusion of the count of mesopelagic fishes as a covariate. As explained in Section 5.3.1, the number of mesopelagic fishes in a trawl has a big impact on the number of tracks for other species. Because of their impact, the inclusion of them as a covariate largely improves the models.

Note that our Full model is five-dimensional, and this can lead to over-fitting, giving us a non-representative R-squared value. The Full model-sum version, is on the other hand three-dimensional and comparable to the previous work using object detection and regression, since this is the same dimensional space they used.

5.3.3 Regression on labelled stations

Our models' performance on the labeled stations varies greatly depending on the species, and whether we use label data or catch data. Overall the performance was better on the catch data reasoned by lackluster performance on the label data regarding herring.

As seen in Table 4.3, the label WAR score is 0.65, which implies that the predicted number of instances is lower than the label. Meanwhile, the WAR score on catch data is 0.97, implying that there is only a minor discrepancy between the catch number and predicted number. These results could be a sign that the catch data is biased in some way, underestimating the real number of species.

Interestingly, by looking at Figure 4.4 and Figure 4.5 we can see that the number of tracks for herring was relatively accurate at about 1300, but because of the high number of mesopelagic fishes, and the negative covariate associated with them, the regression reduced it closer to 1000. Showing how this method can perform differently depending on the composition of the water column.

The R-squared values for blue whiting on the labeled stations are substantially lower than when scoring on the 20-station dataset. This can likely be attributed to the inherent variation from trawl to trawl and the fact that the data from only three stations form a relatively small test set.

The labeled stations had no mackerel in the catch or the labels, and therefore any predictions over 0 results in error. Mackerel is the species with the best MAE score on the labeled stations, which is the most meaningful of the three metrics when the numbers are all 0.

5.3.4 Tracking

It is worth mentioning that the tracker in and of itself is not very accurate when looking at images. This is evident when looking at the median number of frames per species, 15 for blue whiting, 36 for herring, 37 for mackerel, and 2 for mesopelagic fishes. In contrast, the median number of frames per species in the label data is considerably lower: 4 for blue whiting, 6 for herring, and 3 for mesopelagic fishes. Note that the label data does not contain any mackerel, thereby eliminating the possibility of extrapolating mackerel data.

5.4 Suggestions for Future Research

Looking ahead, we can see several potential areas for further study and refinement that could enhance the accuracy and efficacy of our models.

One such area involves refining the object detector. Given the issue of misidentifications that we have observed, retraining the object detector could potentially improve the accuracy of species prediction. This could be achieved by manually labeling data, or by using depth filters and catch data to filter out misclassifications. For example, if a trawls catch data reports zero mackerel, we can use that to remove any detected mackerel from the data generated in that trawl.

The Deep Vision camera system has two cameras inside the camera box. In this paper, we only used the left camera-data, as to not add another dimension to our experiments. However, using both cameras could potentially be used to further reduce the number of misclassified species, and improve tracks.

To better compare the performance of tracking versus object detection on agent counting, specifically on the 51 stations used in this paper, one could train a regression model on the raw object detections used to create the tracks and compare it to the regression models using the tracks as input.

To better the analysis of tracks connected to labels presented in Section 4.4.2, future work could include an extra category for labels that are covered by several tracks. In addition to this, matching a label to a track can be made better by taking into account the size of the tracking-box and the distance between the label-dots. One could also include other species that are not blue whiting, herring, mackerel or mesopelagic fishes, to see what is most commonly misidentified.

Chapter 6

Conclusion

This study primarily aimed to improve the count and distribution prediction accuracy of three key species: blue whiting, herring, and mackerel. We did this by applying object tracking methods to an object detection dataset, and performing linear regression on the resulting number of tracks and the catch data. The results from our series of experiments reveal several valuable insights and improvements over previous studies in both count prediction and species distribution.

Our Full model exhibited superior performance in regression analysis, generating high R-squared values for all three species, outperforming the reference models. The highest achieved R-squared values were 0.95 for blue whiting, 0.91 for herring, and 0.88 for mackerel. This represented a significant improvement over previous work for herring in particular, demonstrating the effectiveness of our approach. However, the Full model could potentially be over-fitting the data given its five dimensional structure. The Full model-sum version performs slightly worse, but still displays a considerable average improvement over previous work.

When the same models were applied to label data, the regression performance showed some variance between species. Blue whiting maintained consistent R-squared values across catch and label data, but herring performed significantly better on catch data than label data, which could be explained by a bias in the catch data. The regression model for mackerel, unfortunately, did not show any explanatory power for the test data, with an R-squared value of 0.00.

The analysis of the tracking algorithm's performance when connecting tracks to labeled data showed that mesopelagic species were often misidentified as blue whiting or herring. However, the absence of mackerel in the labeled stations left us without insights into how this species is misidentified.

Despite these considerations, our results indicate that the combination of depth-based filtering, regression analysis, and tracking algorithms can significantly enhance our ability to predict and monitor the distribution of important fish species. This improvement could provide valuable insights to aid marine conservation and stock assessment, leading to more accurate information gained from trawls.

List of Acronyms and Abbreviations

ANN Artificial Neural Network.

CNN Convolutional Neural Networks.

CSV Comma Separated Values.

FPN Feature Pyramid Networks.

FPS Frames Per Second.

LWT Labels Without Tracks.

MAE Mean Absolute Error.

mAP mean Average Precision.

MOT Multi-Object Tracking.

TWL Tracks Without Labels.

WAR Weighted Average Ratio.

Bibliography

[1] FishBase. https://www.fishbase.se/search.php. Accessed on 7th June 2023.

[2] Vaneeda Allken, Shale Rosen, Nils Olav Handegard, and Ketil Malde. A deep learning-based method to identify and count pelagic and mesopelagic fishes from trawl camera images. *ICES Journal of Marine Science*, 78(10):3780–3792, 11 2021. ISSN 1054-3139. doi: 10.1093/icesjms/fsab227.
URL: https://doi.org/10.1093/icesjms/fsab227.

[3] Vaneeda Allken, Shale Rosen, Nils Olav Handegard, and Ketil Malde. A real-world dataset and data simulation algorithm for automated fish species identification. *Geoscience Data Journal*, 8(2):199–209, 2021. doi: https://doi.org/10.1002/gdj3.114.
URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/gdj3.114.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
URL: http://arxiv.org/abs/1512.03385.

[5] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

[6] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017.
URL: http://arxiv.org/abs/1708.02002.

[7] Ketil Malde. stracks. https://github.com/ketil-malde/stracks, 2023. Accessed: 2023-06-12.

[8] United Nations. Goal 14: Life below water, 2023.
URL: https://sdgs.un.org/goals/goal14. Accessed: 11-06-2023.

[9] Shale Rosen, Terje Jörgensen, Darren Hammersland-White, and Jens Christian Holst. Deepvision: a stereo camera system provides highly accurate counts and lengths of fish passing inside a trawl. *Canadian Journal of Fisheries and Aquatic Sciences*, 70(10):1456–1467, 2013. doi: 10.1139/cjfas-2013-0124.
URL: `https://doi.org/10.1139/cjfas-2013-0124`.

[10] Scantrol Deep Vision. Deep Vision for Marine Research. `https://www.deepvision.no/deep-vision-for-marine-research`, 2021. Accessed: 2023-06-12.

[11] George A. F. Seber and Alan J. Lee. *Linear Regression Analysis*. John Wiley & Sons, 2003.

[12] Eugenie Heliana Taraneh Westergerling. A comparison of an in-trawl camera system to acoustic and catch results for small pelagic and mesopelagic fish. Master's thesis, 2019.
URL: `https://matix.imbrsea.eu/node/3926`.

Appendix A

Tables

Listing A.1: R-square values for each species and pair of scales on the full dataset. The pairs of scales (x,y) correspond to the scale used for 2022 data and 2018 data, respectively.

scales species	0.25,0.5	0.5,1	0.75,1.5	1,2	1.5,3	2,4	2.5,5	3,6	3.5,7
Bluewhiting	0.86	0.89	0.90	0.91	0.92	0.93	0.92	0.93	0.93
Herring	0.54	0.62	0.71	0.73	0.77	0.82	0.81	0.81	0.78
Mackerel	0.39	0.50	0.59	0.67	0.72	0.73	0.76	0.75	0.72
Average	0.60	0.67	0.73	0.77	0.80	0.83	0.83	0.83	0.81

scales species	4,8	4.5,9	5,10	5.5,11	6,12	6.5,13	7,14	7.5,15	8,16
Bluewhiting	0.94	0.93	0.92	0.93	0.93	0.92	0.92	0.92	0.92
Herring	0.81	0.81	0.83	0.87	0.81	0.81	0.81	0.79	0.78
Mackerel	0.78	0.76	0.73	0.73	0.74	0.75	0.73	0.75	0.71
Average	0.84	0.83	0.83	0.84	0.83	0.83	0.82	0.82	0.80

Appendix B

Figures

Listing B.1: Number of frames per species in label data.