Linda Helen Sperre

Machine Learning for Automatically Counting Growth-Stunted Fish in Sea Cages

Master's thesis in Simulation and Visualization
Supervisor: Ricardo da Silva Torres
June 2023

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

Linda Helen Sperre

Machine Learning for Automatically Counting Growth-Stunted Fish in Sea Cages

Master's thesis in Simulation and Visualization
Supervisor: Ricardo da Silva Torres
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

ABSTRACT

Sea cages used for salmon farming often contain a large amount of growth-stunted ("loser") fish. The presence of loser fish might indicate insufficient welfare factors or diseases. By monitoring their occurrence, these welfare problems can be identified and addressed. Current monitoring of loser fish in sea cages is mostly performed manually. The objective of this study is to design, develop, and validate a machine-learning solution for automatically monitoring the amount of loser fish in sea cages. The dataset used in this study consists of images depicting healthy and loser salmon from a sea cage. Fish in these images are detected using an object detection algorithm, with a pre-trained fish detection model. Detected fish are then classified into classes (*healthy* or *loser* fish). The number of fish classified as loser fish is identified and counted. The machine learning solutions investigated in our study consider the use of a convolutional neural network feature extractor, principal component analysis for dimensionality reduction, grid search for defining the best configuration of different classification algorithms, and combinations of those classification algorithms. In order to optimise these combinations, the diversity between the classification algorithms is also investigated. This study has resulted in a framework able to detect multiple fish from images in a sea cage, detect 74.5% of the fish detected in the ground-truth, and classify each of the detected fish with an accuracy of 92.9%. The proposed solution is a promising tool for increasing fish welfare, with a significant improvement from manual counting in terms of effectiveness and reliability.

Lakseoppdrettsmerder inneholder ofte et stort antall veksthindrede fisk omtalt som "taperfisk". Tilstedeværelsen av taperfisk kan indikere utilstrekkelige velferdsfaktorer eller sykdommer. Ved å overvåke forekomsten av taperfisk kan disse velferdsproblemene bli identifisert og adressert. Foreløpig overvåking av taperfisk utføres hovedsakelig manuelt. Formålet med denne studien er å designe, utvikle og validere en maskinlæringsløsning for automatisk overvåking av mengden taperfisk i fiskemerder. Datasettet som brukes i denne studien består av bilder som viser frisk og taper laks fra fiskemerder. Fisk i disse bildene er detektert ved å bruke en objektdetekteringsalgoritme, med en forhåndstrent fiskedetekteringsmodell. Detekterte fisk blir deretter klassifisert i kategorier (*frisk* eller *taper* fisk). Antallet fisk klassifisert som taperfisk blir identifisert og telt. Den undersøkte maskinlæringsløsningen i denne studien betrakter bruken av et konvolusjonelt nevralt nettverk for å redusere overflødighet i dataen, hovedkomponentanalyse for dimensjonsreduksjon, grid search for å finne beste konfigurasjoner av forskjellige klassifiseringsalgoritmer og kombinasjoner av disse klassifiseringsalgoritmene. For å optimalisere disse kombinasjonene blir mangfoldet mellom klassifiseringsalgoritmene undersøkt. Denne studien har resultert i et rammeverk som kan detektere flere fisk fra bilder i en fiskemerd, og oppdage 74.5% av de samme fiskene som er oppdaget i referensedataen og klassifisere hver av de detekterte fiskene med en nøyaktighet på 92.9%. Den foreslåtte løsningen er et lovende verktøy for å øke fiskevelferden, med en bedydelig forbedring fra manuell telling angående effektivitet og pålitelighet.

PREFACE

This thesis is written as a master's thesis in the degree of Master in Simulation and Visualisation at the Norwegian University of Science and Technology (NTNU) in Ålesund. This study program is coordinated by the Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering. The work was conducted during the spring of 2023.

What intrigued me about this project was the importance of improving fish welfare and the relevance to several elements learned during the master's degree. Detection and classification problems touch the courses on data processing, artificial intelligence, and machine learning as well as the specialisation course on object detection and the specialisation project related to counting loser fish. Additionally to the relevance of subjects learned in the degree, it provided opportunities for innovation, creativity, and new learning.

I would like to thank my supervisor Ricardo da Silva Torres for providing a very interesting and challenging problem for the thesis, and for assisting during the development of the research. I would also like to thank the Ph.D. community of salmon detection and counting at NTNU, including Lars Gansel, Kana Banno, Mariana Anichini, and Clara Sauphar for the useful discussions regarding my work and for providing the datasets used in my specialisation project during the autumn of 2022 and in my master's thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

List of all abbreviations in alphabetic order:

- **BB** Bounding Box
- **CNN** Convolutional Neural Network
- **DCNN** Deep dilated Convolutional Neural Network
- **IoU** Intersection over Union
- **KNN** K-Nearest Neighbors
- **MCNN** Multi-column Convolution Neural Network
- **MLP** Multi-layer Perceptron
- **NTNU** Norwegian University of Science and Technology
- **PCA** Principal Component Analysis
- **RSPCA** Royal Society for the Prevention of Cruelty to Animals
- **SVC** C-support Vector Classification
- **SWIM** Salmon Welfare Index

CHAPTER

ONE

INTRODUCTION

1.1 Motivation and Context

Sick fish is commonly referred to as growth-stunted, emaciated, drop-outs, or simply "loser" [1, 2]. Even though loser fish might be the most offensive term, it is one of the most commonly used. These are fish that grow too slowly during their first months in comparison to the rest of the group, likely because of a combination of insufficient welfare factors [3, 4]. Loser fish is less likely to survive, often are associated with low animal indicators, and are less likely to grow to an economically profitable size [3, 4].

The occurrence of loser fish in sea cages can indicate problems with feeding strategy, water conditions, diseases, or injury [5]. Such problems cause a lack of welfare and reduce the chance of survival of the fish [3]. The detection of loser fish is important for establishing the condition of the sea cage, and in order to find and fix the problems leading to their presence [4]. Currently, the occurrence of loser fish is detected manually, with no automated system for ensuring consistent or viable surveillance [1]. Therefore, an automated system is much needed for counting the amount of loser fish in sea cages, in order to discover problems leading to their occurrence and, therefore, for improving their welfare.

The main goal of this thesis is to develop a proof of concept of a machine vision system for loser fish counting and to develop a solution that can be implemented in sea cages. The main part of the thesis concerns the identification of suitable models to classify images into two categories *healthy* or *loser* depending on whether the image depicts a healthy or loser fish, respectively. The conducted research also includes the detection of fish in a sea cage and a counting system. A large dataset of labelled data is needed for training the detection model. Since this was not available for this project, the detection of fish consists of a pre-trained fish detection model instead. The proposed solution receives, as input, images obtained from underwater cameras mounted inside sea cages.

This thesis builds upon knowledge and experiments conducted through a specialisation project regarding the same topic, whose associated report can be found in Appendix A. The knowledge gained from the specialisation project has directed the focus of this work, and all the experiments have been extended and redone, with a new dataset, leading to new results and conclusions.

1.2 Objective and Research Questions

The objective of this work is to design, develop, and validate a framework for automatically monitoring the amount of loser fish in a sea cage. The framework is composed of three main components: detection, classification, and counting. The first one uses an object detection algorithm for detecting fish in underwater images obtained from a sea cage. The second one uses a machine learning algorithm for classifying each fish as *healthy* or a *loser*. The third one is dedicated to counting the occurrence of loser fish in the sea cage. In this work, I focus on investigating the detection, classification, and counting results using a new dataset, exploring the correlation and diversity of different classification methods, and looking into how the best-performing and less-correlated classification algorithms can be combined using ensemble methods.

In this work, we address the following research questions:

RQ_1 – What is the performance of a state-of-the-art detector for the fish detection problem?

In this question, the performance of an object detection method will be tested using a pre-trained fish detection model, named fish_detection [6]. This method is chosen as it is trained on Open Images v7,[1] which is one of few large datasets containing underwater images of salmon, which makes the model ideal for detecting different types of fish in a sea cage (especially salmon). This investigation opens up opportunities for the use of transfer learning procedures based on pre-trained models for the problem. The proposed pipeline would then be suitable for scenarios where available labeled training sets are small or are hard to be created.

RQ_2 – Which state-of-the-art classification algorithms has the highest effectiveness performance for the fish classification problem?

In this question, the performance of five different classification algorithms is tested, considering a grid search procedure for determining proper values for their hyper-parameters.

RQ_3 – How much diversity is there between the classification algorithms used in the fish classification problem?

In this question, the diversity between the different classification methods is compared and investigated. The goal is to identify the most promising classifiers to be combined.

RQ_4 – Which state-of-the-art ensemble methods have the highest performance for the fish classification problem?

In this question, the performance of two different ensemble methods for combining the classification algorithms is tested.

RQ_5 – What is the performance of the proposed loser fish counting system?

In this question, the performance of a loser fish counting system is tested.

[1] Google Apis, Open Images Dataset V7 and Extensions. `https://storage.googleapis.com/openimages/web/index.html` (As of June 2023).

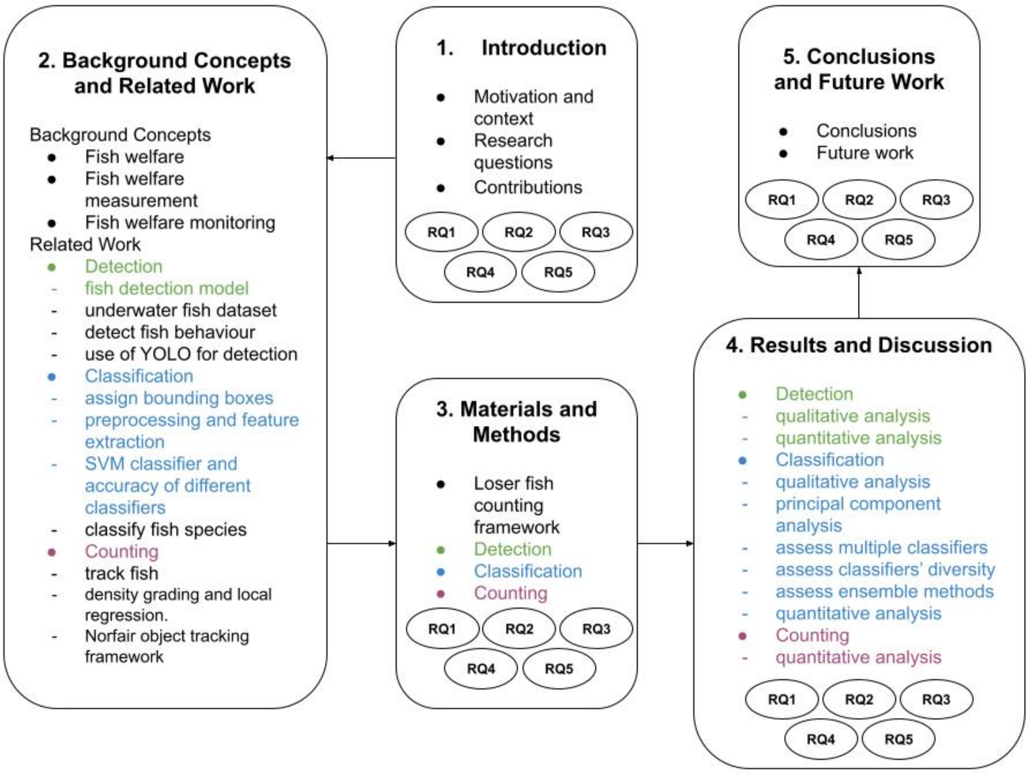


Figure 1.4.1: Outline of the thesis structure.

1.3 Contributions

For this thesis, a new system for counting the amount of loser fish in sea cages is introduced. The system is developed using transfer learning and validated using two different datasets. It consists of three steps: detection, classification (where an ensemble has been assessed using a pretrained detection model), and counting.

The source code of the different components of the system is available at `https://github.com/Linda432/fish-counting-system.git` (As of June 2023).

1.4 Outline

Figure 1.4.1 shows the structure of the thesis. The thesis consists of five chapters: introduction, background concepts and related work, materials and methods, results and discussion, and conclusions. The related content to the main components of the solution: detection, classification, and counting are highlighted with different colours. The figure also indicates which research questions (RQ_1, RQ_2, RQ_3, RQ_4, and RQ_5) are addressed in each chapter.

Chapter 2 - Background Concepts and Related Work provides background information on the problem and describes relevant related work. **Chapter 3 - Materials and Methods** contains a description of the developed methodology and materials used during the conducted investigation. **Chapter 4 - Results and Discussion** presents all the experiments and results of the developed solution, as well as a discussion of the obtained results. Finally, **Chapter 5 -**

Conclusions summarises the main findings and provides recommendations for future work.

CHAPTER

TWO

BACKGROUND CONCEPTS AND RELATED WORK

This chapter introduces background concepts related to the investigated problems. This chapter also overviews relevant related work.

2.1 Background Concepts

This section covers background concepts related to fish welfare and its assessment.

2.1.1 Fish Welfare

Welfare is, according to the Cambridge dictionary, correlated to the physical and mental health as well as the happiness of a being [7]. Physical healthy can be measured based on disease, injury, or illness, while mental health can be measured based on pain, fear, or suffering [8]. There is however a scientific debate regarding the ability of fish to experience pain or fear. Key [9] examined whether the brain structures of fish were capable of conveying senses of pain and consciousness. The conclusions of his study are similar to those of Rose [10]. According to them, such senses are impossible for fish brains [9, 11].

Others, such as Chandroo [12] and Dunlop [13], suggest, however, that there exists evidence that nociception in fish is experienced. For example, fish is capable of experiencing suffering in the form of pain and fear [8]. Rose [10] partially agrees and explains that they display physiological stress responses to noxious stimuli, even though they do not experience pain, fear, or emotions, which is potentially injurious.

Since there is a lot of disagreement regarding the mental senses of fish, it is impossible to know for sure exactly what a fish experiences. However, if they experience mental suffering, the nature of their pain or fear should be assessed, taken into account, and prevented [8].

2.1.2 Fish Welfare Measurement

Welfare can be complicated to be measured but can be split into two welfare indicator categories: direct (animal-based) and indirect (environment-based). The animal-based welfare indicators are more directly linked to the state of the fish, where attributes from the animal itself indicate the lack of welfare needs. Such needs can be identified by the condition factor of the fish, the degree of emaciation, or the existence of damaged gill tissue. Environmental-based welfare indicators have the opportunity to detect a poor welfare problem before the problem becomes visible on the fish. Such indicators might be related to the farming system, such as the water temperature or oxygen levels [14].

Most animal welfare assessment protocols use a combination of animal-based and environment-based indicators. The predicted appropriate indicators are used together with different statistical techniques, such as the monitoring program suggested in the Royal Society for the Prevention of Cruelty to Animals (RSPCA) welfare standards for farmed Atlantic salmon [15], the welfare assessment protocol developed by the Norwegian Veterinary Institute [16], or the Salmon Welfare Index (SWIM) developed by Stien at al. [17] [14].

There is no single measure of welfare for fish, but a combination of physiological, biochemical, and behaviour measures are often used [8]. Ashley [8], Dawkins [18], Broom [19] and Noble et al. [14] all agree that behaviour measurement might be the most important method to assess welfare. Behaviour measurements indicate the welfare of the fish at the point of observation and can be used to assess both physical and mental health. A large amount of salmon sea cages are equipped with underwater cameras, such as fish farmers can use behaviour as a key tool for monitoring the welfare of fish [14].

2.1.3 Fish Welfare Monitoring

Fish have rich body language, which can indicate their welfare. By using behaviour-based methods, including their swimming modes, fin displays, gill ventilation, and skin pigment patterns, their response to food and their position in the water can be observed and analysed. Poor welfare might be indicated by a low response to food, slow swimming, and increased group clumping. Other factors to consider are fish with low condition factors and fish swimming alone. Fish with these factors are referred to as "loser" fish. These are unhealthy fish, which might be caused by a combination of insufficient welfare factors, such as parasites, disease, stress, or environmental factors [4, 14].

About 15-20% of fish in sea cages die before they are big enough to be slaughtered because of insufficient welfare factors. Additionally, there are millions of loser fish with welfare problems among the surviving fish in the aquaculture industry [5]. Loser fish are unwanted from a production perspective since they result in big financial losses and harm the reputation of the industry. They might be able to survive for a long time, but they often get diseases and do not have a satisfying life regarding animal welfare [4]. Loser fish are often therefore removed and euthanized by the fish farmers if detected [3, 4, 5].

This thesis focuses on the automatic detection of loser fish based on images. It can, therefore, be explored for animal-based welfare assessment.

2.2 Related Work

Detection, classification, and counting methods have been broadly studied and developed in recent years. Regarding fish, these methods have been highly utilised for finding the position of fish, creating bounding boxes around detected fish, deciding the fish species, evaluating the fish behaviour, tracking fish trajectories, and counting fish.

Saleh et al. [20] presented a benchmark solution for using visual analysis for monitoring fish habitats as a step towards sustainable fisheries. Their product, DeepFish, is a dataset consisting of approximately 40 thousand underwater images from different marine environments of tropical Australia, created to train and test methods for several computer vision tasks. The dataset uses classification labels, point-level, and segmentation labels, which enables models to learn to automatically monitor fish count, identify their locations, and estimate their sizes.

As a detection method, Muksit et al. [21] developed a deep learning-based fish detection model called YOLO-Fish. Two versions are created where YOLO-Fish-1 fixes an issue with YOLOv3 [1], enhancing it to be able to detect tiny fish. YOLO-Fish-2 adds Spatial Pyramid Pooling to the first model, improving it to be able to detect fish in dynamic environments. Two datasets were used for testing the models, DeepFish, and OzFish, where the models obtained average precision of 76.56% and 75.70%. Their models are more lightweight compared to other versions of YOLO, with similar performance.

Al Aoi [6] also developed a fish detection model. The proposed model was trained on a large number of images of different fish species from the Open Images v7 [2] dataset, and using the TensorFlow [3] library. The model was able to detect a large number of fish from an image and create bounding boxes around them. This model is investigated for loser detection in this thesis.

With the goal of improving fish welfare, Li et al. [22] proposed a novel method of abnormal behaviour detection based on image fusion. This would detect early abnormal behaviour in single fish in real time. Outline information of moving objects was extracted and the position information of the fish was enhanced. A method named BCS-YOLOv5 was developed by adding bidirectional feature pyramid network, coordinate attention block, and spatial pyramid pooling to YOLOv5. This method achieved the best accuracy compared with two other typical models with an average accuracy of 96.69% at 45 frames per second. The BCS-YOLOv5 method improved the extraction of location information and quantitatively detected similar anomalous behaviour, such that abnormal fish behaviour in aquaculture could be detected in real-time.

Wu et al. [23] introduced an effective method for detecting and recognising the starvation-stress behaviour of individual fish in order to ensure fish welfare. Their study focused on the precision-feeding strategy of fish and investigated how stress behaviour could be detected by quantifying the swimming activity of fish based on their swimming intensity. For computing that, the angular information of the fish, including the steering angle, tail-bending angle, and turning speed were considered as key factors. Results obtained via human observation revealed that their detection and recognition method exhibited good performance in the detection of starvation-stress behaviour of darkbarbel catfish. Their study reported an

[1]Joseph Redmon, YOLO: Real-Time Object Detection. `https://pjreddie.com/darknet/yolo/` (As of June 2023).

[2]Google Apis, Open Images Dataset V7 and Extensions. `https://storage.googleapis.com/openimages/web/index.html` (As of June 2023).

[3]TensorFlow, Create production-grade machine learning models with TensorFlow. `https://www.tensorflow.org/` (As of June 2023).

accuracy rate of at least $96.21 \pm 1.42\%$.

Zeng et al. [24] investigated how an Audio Spectrum Swin Transformer (ASST) model using an acoustic signal and attention mechanism could be utilised in order to identify the feeding decisions of fish. The feeding behaviour could be identified based on the acoustics produced by fish chewing feed and activities during feeding. The ASST network for fish feeding behaviour reached an accuracy of 96.16% and could effectively divide the feeding intensity into four grades: strong, medium, weak, and none. The study demonstrated that the model allows for the use of on-demand feeding and provided a basis for developing intelligent feeding machines.

As for classification, different approaches have been investigated. In a study by Alsmadi and Almarashdeh [25], the performance of different fish classification techniques was compared. The performance was based on the availability of preprocessing and feature extraction methods, the number of extracted features, classification accuracy, and the number of fish species recognised. Their study was based on methods gathered from recent works to enhance the understanding of preprocessing methods, feature extraction techniques, and classifiers to guide future research directions and compensate for current research gaps.

Banno et al. [1] developed a fish classification model. Their model was trained on images collected from sea cages of salmon. Each image contained one single fish. Images were labelled according to the presence of healthy or loser fish. They extracted features from the images with a vector size of 2048 features, using a pre-trained Convolutional Neural Network (CNN) model. A binary classifier with Support Vector Machine (SVM) was used to classify the images of fish into two classes: healthy and loser. With this approach, they obtained a classification accuracy of 97.17% for the test data.

Spampinato et al. [26] proposed an automatic fish classification system operating in natural underwater environments for assisting marine biologists in understanding fish behaviour. The fish species classified was Bodianus mesothorax, Chaetodon trifascialis, Chromis viridis, Dascyllus albisella, Dascyllus aruanus, Dascyllus reticulatus, Gomphosus varius, Hemigymnus fasciatus, Plectorhinchus lessonii and Pseudocheilinus hexataenia. The classification system consisted of two types of features: texture features and shape features. An affine transformation was also applied to the images to represent fish in 3D by multiple views for feature extraction. The system obtained an average correct rate of about 92%. A tracking system was combined with a classification layer associating trajectories to fish species. Clustering of these trajectories enabled the detection of unusual fish behaviour.

A Support Vector Machine (SVM)-based technique was utilised by Ogunlana et al. [27] for eliminating the limitations of K-Nearest Neighbour (KNN), K-mean Clustering and Neural Network techniques, as well as improving the classification of fish species. The technique was based on the shape features of fish, where the body and the fin lengths were extracted. The SVM technique obtained a classification accuracy of 78.59%, which was significantly higher than obtained for the other techniques.

Marrable et al. [28] developed a machine-assisted approach for optimising analysis time and providing rapid reporting of the status of marine ecosystems. The approach consisted of assigning bounding boxes in underwater environments containing fish as well as detecting and classifying up to 12 fish species. Bounding box annotations detected and labelled fish with a recall between 70-89% and species were labelled with an F1 score of 79%. In their study, 12% of fish were given a bounding box with species labels, and 88% of fish were located and identified for manual species labelling. Their machine-assisted approach presented a significant advancement towards the applied use of deep learning for detecting fish species with the potential for future fish ecologist uptake but concluded that for now manual labelling and classification effort was still required.

For counting, some different approaches have been proposed in the literature. A lightweight fish counting model called LFCNet was developed by Zhao et al. [29] for fish counting applications of fish farming. Their model consisted of three components: encoder, decoder, and generation head. The encoder utilised density map regression to address the high-density fish issue. The decoder used ghost modules to compress parameters in mobile device applications. The generation head adopted three concentrated comprehensive convolution modules and transposed convolution layers to alleviate the computational overhead and recover the resolution of feature maps. The LFCNet achieved higher counting precision and stability than other comparable methods and provided a balance between accuracy and speed in various fish counting scenarios.

Zhang et al. [30] proposed an automated fish counting method using image density grading and local regression. Fish-connected areas were segmented and four types of image features were extracted from each area. The proposed method performed better than current typical fish counting methods and achieved a mean absolute error of 0.2985, root mean square error of 0.6105, and a coefficient of determination of 0.9607.

An automatic fish counting method based on a hybrid neural network model was proposed by Zhang et al. [31]. This method enabled real-time, accurate, objective, and lossless counting of fish populations in far offshore salmon mariculture. The counting method consisted of a multi-column convolution neural network (MCNN), convolution kernels of different sizes, a deeper dilated convolution neural network (DCNN), and a hybrid neural network model. The counting obtained an accuracy of 95.06%, and the Pearson correlation coefficient between the estimation and the ground-truth was 99%, which represented an improvement from CNN- and MCNN-based methods, providing the counting method as an essential reference for feeding and other breeding operations.

Morais et al. [32] studied the use of computer vision techniques for underwater visual tracking and counting of fish. The method used a Bayesian filtering technique which enabled the tracking of objects of varying numbers over time. Their approach provided relevant information about the characteristics of different fish species such as swimming ability, time of migration, and peak flow rates. The system was also able to estimate fish trajectories over time, which was used for studying the fish behaviour. Performed experiments demonstrated that the proposed method could operate reliably even with severe environmental changes and handle problems such as occlusions, with an overall accuracy of 81%.

The Ocean Aware project is developing a system for monitoring a fish passage observation platform, with the use of detection, classification, and counting. As part of this project, Kandimalla et al. [33] developed and tested an automated real-time deep learning framework, where they used sensors, sonar, and cameras, as well as Convolutional Neural Networks and Kalman filters to classify fish by species in real-time. They were able to accurately detect and classify eight fish species of fish using a YOLO machine learning model and a high-resolution imaging sonar dataset. Fish were also counted using the Norfair object tracking framework with an optical cameras dataset. Their work demonstrated that deep learning models can be used to detect, classify species, and track fish using different types of datasets.

In this work, I use images collected by Banno et al. [1]. I also use part of the CNN model for creating a feature extraction function, and consider the SVM classifier as one of the methods for constructing a classification system. The pre-trained fish detection model by Al Aoi [6] is utilised for detecting fish in images. For detected fish in underwater images, bounding boxes have been assigned, similarly to the approach of Marrable et al. [28]. Similarly to the study of Alsmadi and Almarashdeh [25], the performance of using preprocessing and feature extraction methods is evaluated, and the classification accuracy of different classifiers is compared. As Ogunlana et al. [27], the SVM classifier is also utilised for the classification model. For the counting system, all the papers covered as related work have used methods that are different from the one employed in this study for counting fish.

Different from the initiatives of Li et al. [22], Wu et al. [23], Zeng et al. [24], and Spampinato et al. [26], this study does not investigate fish behaviour. Also, this study is not concerned with the detection of different fish species, such as Aoi [6], Alsmadi and Almarashdeh [25], Spampinato et al. [26], Ogunlana et al. [27], Marrable et al. [28], Morais et al. [32] and Kandimalla et al. [33]. Fish tracking, a problem investigated by Spampinato et al. [26], Morais et al. [32], Kandimalla et al. [33], is not considered in the study described in this thesis.

CHAPTER

THREE

MATERIALS AND METHODS

This chapter provides a description of the research methodology adopted in this study for monitoring loser fish. The investigated solution consists of three pipelines: fish detection, fish classification, and fish counting. Each step of the methodology is described as well as the corresponding materials and methods.

3.1 Loser Fish Counting Framework

Figure 3.1.1 shows the proposed solution for counting fish in sea cages. First, fish in images from a sea cage are detected. Next, the detected fish are classified. Finally, the classified fish are counted. As output, the amount of loser fish in the images from the sea cage is provided.

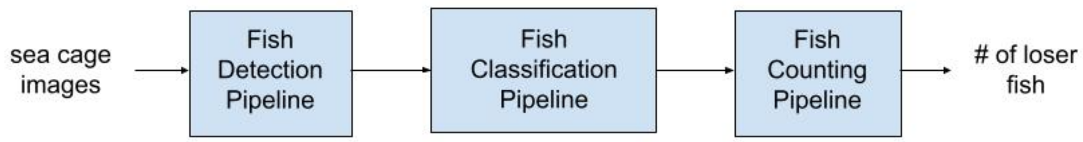


Figure 3.1.1: Pipeline for the proposed solution.

The first component of the framework refers to the fish detection problem. Figure 3.1.2 shows the pipeline adopted for the detection. First, images from the sea cage are provided as input. A bounding box is then created around each detected fish in an image before each bounding box is cropped into a new image.

Figure 3.1.3 shows the pipeline for the classification system. First, cropped images are imported as input. Then, five different classification algorithms are considered. Their results are combined by means of an ensemble method. Lastly, the fish in the cropped images are classified as either *healthy* or *loser*.

Figure 3.1.4 shows the pipeline for the counting step. First, cropped images from the detection method are provided as input. Then, features from the images are extracted. Next, the classification model is applied to classifying loser fish. Lastly, the number of fish classified as *loser* is counted.

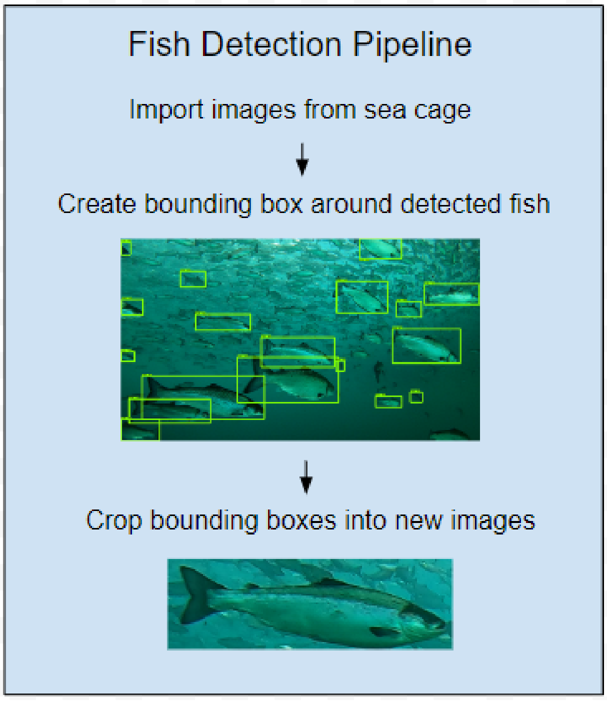


Figure 3.1.2: Pipeline for fish detection.

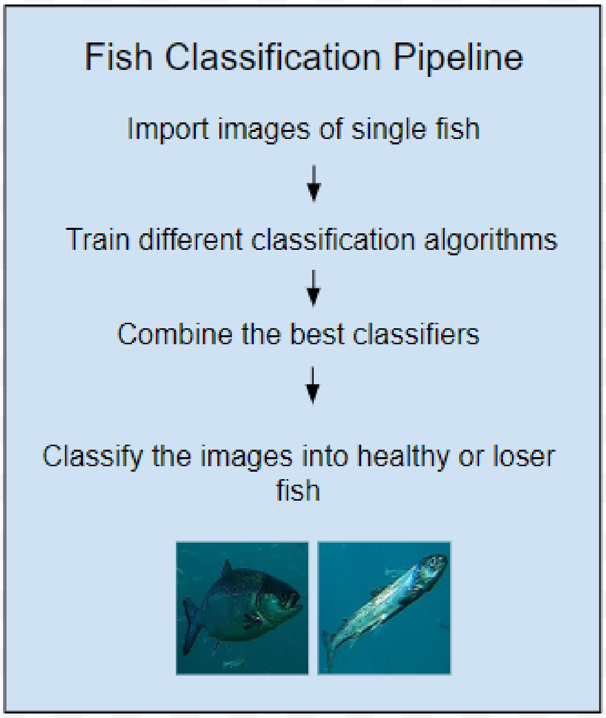


Figure 3.1.3: Pipeline for fish classification.

3.2 Evaluation Protocol

This section provides a description of the evaluation protocol used for addressing the different research questions. They refer to the fish detection (Section 3.2.1), fish classification (Section 3.2.2), and fish counting (Section 3.2.3) problems. The description of the methodology adopted for addressing each problem is organised

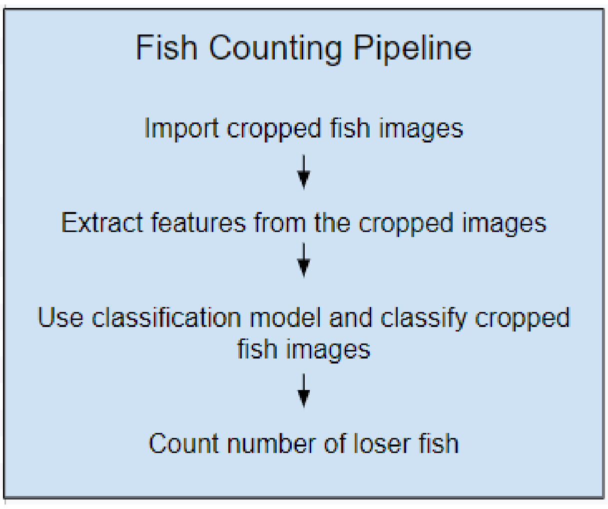


Figure 3.1.4: Pipeline for fish counting.

into three sections: the dataset used, the evaluation measurements considered in the assessment of methods, and the implementation aspects related to performed experiments.

3.2.1 Fish Detection

This section presents the evaluation protocol of the first component of the proposed framework, fish detection. It refers to RQ_1 (Section 1.2 – *What is the performance of a state-of-the-art detector for the fish detection problem?*).

3.2.1.1 Dataset

The dataset was created by Banno et al. [1] and consists of 207 images taken underwater from a sea cage. Each image consists of multiple fish (various amounts of healthy and loser fish) depicted at different sizes and positions. The images were taken from different locations, depths, light conditions, and varying water quality. The dataset is divided into training, validation, and test sets, with 145, 41, and 21 images, respectively; which corresponds to a distribution of 70%, 20%, and 10%. Fish in the training, validation, and test sets were pre-detected and labelled manually by experts. Each of the images was given a corresponding text file of registered fish with a position of four coordinates (enclosing bounding box) and a label (*healthy* or *loser*). All images are of size 1920×1080 pixels.

For the detection part, only the test set from the dataset is used for assessing the performance of the detection model. The number of images and instances (number of bounding boxes) are presented in Table 3.2.1.

Table 3.2.1: The dataset used for the assessment of the detection module.

Dataset	# of images from sea cage	# of bounding boxes
Test set	21	210

3.2.1.2 Evaluation Metric

In order to evaluate the performance of the object detection algorithm, the overlap between detected bounding boxes and pre-detected bounding boxes was computed using the intersection over union[1] (IoU) metric. For each image in the test set, the area of overlap between the automatically detected bounding boxes and the manually detected bounding boxes was computed and divided by the area referring to the combination of both bounding boxes. In cases in which a bounding box is automatically or manually detected (ground-truth) without any counterpart, the intersection over the union score is zero. More formally,

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{3.1}$$

The bounding boxes are defined by an x-axis horizontally, a y-axis vertically, and with the center in the top left corner. The overlap was therefore given by subtracting the smallest x-axis right value from the largest left value and multiplying it with the subtraction of the smallest bottom value and the largest top value. The area of the union was calculated by summing the area of both corresponding bounding boxes and subtracting the area of overlap. The overlap for bounding boxes is given as a percentage score.

3.2.1.3 Implementation Aspects

This section provides an overview of how the fish detection approach was implemented.

- **Fish Detection Model:** In order to detect multiple fish in an image from a sea cage, a fish detection method was created. The dataset was considered to be too small to train a feasible detection model from scratch. Therefore, a pre-trained fish detection model from Al Aoi [6] was used. This model was trained on a large dataset of various fish species using Open Images Dataset[2] and detects objects that are predicted with more than 60% certainty of being a fish.

- **Create Bounding Boxes:** For each fish or a likely detected fish, a bounding box was created around it. Bounding boxes are encoded as a rectangle of different sizes based on the shape of the fish.

  A Tensorflow framework created by Yu et al. [34] was used together with the fish detection model for developing the object detection method. Bounding boxes were created around automatically detected fish, together with the percentage of how likely each detected object was a fish.

- **Crop Bounding Boxes into new Images:** Once the bounding boxes are defined, each bounding box can be cropped into a new image. These new images of different sizes each contain exactly one fish which is centered and covers most of the image, i.e., background included is expected to be

[1] PyImageSearch, Intersection over Union (IoU) for object detection. `https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/` (As of June 2023).

[2] The training set comprises 24,403 fish bounding boxes.

minimal. All the images were labelled with a value of 0 for healthy fish and 1 for loser fish.

For object detection, Tensorflow version 2.11 was used, and protobuf version 3.19.6.

3.2.2 Fish Classification

This section presents the evaluation protocol of the second component of the proposed framework, fish classification. It refers to the research questions RQ_2, RQ_3, and RQ_4 (Section 1.2:

- RQ_2 – Which state-of-the-art classification algorithms has the highest effectiveness performance for the fish classification problem?
- RQ_3 – How much diversity is there between the classification algorithms used in the fish classification problem?
- RQ_4 – Which state-of-the-art ensemble methods have the highest performance for the fish classification problem?

3.2.2.1 Dataset

For the classification part, the whole dataset created by Banno et al [1] is used. The number of images used for training, validation, and test is presented in Table 3.2.2.

For each of the images in the dataset, a bounding box was created around each manually pre-detected fish (ground-truth) based on their given position and manually assigned label. The colour of the bounding box was defined based on the label of the fish. Bounding boxes in yellow are used for healthy fish, while the red ones refer to loser fish. The label is also displayed as a textual label in the middle of the image.

Each bounding box was cropped into a new image, similar to the automatically detected fish. This cropping of images increased the dataset to consist of 1736 images of fish in a sea cage. For the training set 888 images of healthy fish and 306 images of loser fish were obtained. For the validation set, 260 images of healthy fish and 86 images of loser fish were acquired. For the test set, 171 healthy fish and 39 loser fish were obtained.

The images of the training, validation, and test set were further divided into sets of healthy and loser fish based on the cropped images assigned label. The cropped images in these sets were stored group-wise based on their label and were shuffled randomly for splitting up the groups before training the classification model.

3.2.2.2 Evaluation Measurements

- **Evaluating the Classification Algorithms:** For classification, sklearn version 1.2.2 were used, as well as joblib version 1.2.0. The classification algorithms were assessed by taking into account two criteria: accuracy and F1 score (both for finding healthy and loser). Each classifier was trained using the training set, and predictions were performed using the validation

Table 3.2.2: The dataset used for the assessment of the classification module.

Dataset	# of images	# of healthy instances	# of loser instances
Training set	145	888	306
Validation set	41	260	86
Test set	21	171	39

set. The chosen parameters for optimisation for each of the classifiers were also identified. The training and validation scores for each classifier were obtained to check if any of the training was overfitting, e.g., fit the training set too well so that it would not generalise for the validation set.

The classification algorithms were also evaluated based on the accuracy, precision, recall/sensitivity, specificity[3], and F1 score [4], given as percentage scores. The accuracy tells how good the model is at classifying the data, given by:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

where TP stands for true positive, TN stands for true negative, FP stands for false positive and FN stands for false negative.

The precision tells how many of the positive identifications were actually correct, given by:

$$Precision = \frac{\text{Number of correctly classified positive samples}}{\text{Number of classified positive samples}} = \frac{TP}{TP + FP} \tag{3.3}$$

The recall/sensitivity tells how many actual positives were correctly identified, given by:

$$Recall = \frac{\text{Number of correctly classified positive samples}}{\text{Number of positive samples}} = \frac{TP}{TP + FN} \tag{3.4}$$

Specificity tells how many of the actual negatives were correctly identified, given by:

$$Specificity = \frac{\text{Number of correctly classified negative samples}}{\text{Number of negative samples}} = \frac{TN}{TN + FP} \tag{3.5}$$

F1 score combines the precision and recall results, given by:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.6}$$

[3]Wikipedia, Sensitivity and specificity. `https://en.wikipedia.org/wiki/Sensitivity_and_specificity` (As of June 2023).

[4]Wikipedia, Precision and recall. `https://en.wikipedia.org/wiki/Precision_and_recall` (As of June 2023).

The classifier receiving the highest F1 score was considered as the best classification algorithm. The performance of the best algorithm was encoded into a confusion matrix, which contains the results of the model by showing actual and predicted classes for the dataset samples.

The classification model was developed using a standard Google Colab session with an Intel(R) Xeon(R) CPU @2.20GHz, and around 13 GB of RAM.

- **Evaluating the Diversity of Classification Algorithms:** The correlation of the classification algorithms can determine the diversity in their classification choices. By investigating their diversity, we can determine to what extent classifiers provide different views regarding the dataset, i.e., to what extent classifiers assign different labels to the same samples. The diversity of classifiers is computed based on the sample correlation coefficient[5], defined as follows:

$$\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{3.7}$$

  This method takes as input two parameters, which are any two classifier predictions, and compares them for every image to find the correlation between them. x_i is the i-th element of the first classifier predictions, y_i is the i-th element of the second classifier predictions, $\bar{x}$ is the average of the first classifier predictions and $\bar{y}$ is the average of the second classifier predictions.

  The correlation is given as a number between 0 and 1, with a high value corresponding to a high correlation and a low value corresponding to a low correlation. Pair of classifiers with low correlation and high average F1 scores are promising to be combined [35].

- **Evaluating the Ensemble Methods:** The ensemble methods were, similarly to the classification algorithms, evaluated based on the accuracy, precision, recall/sensitivity, and F1 score, given as percentage scores. For all ensemble methods, the parameters giving the highest F1 score were evaluated as the best result, and the results of the best ensemble method were also encoded into a confusion matrix. For the voting-based ensemble method, the highest F1 score is likely to be achieved when voting with a different amount of weight for each classifier. For the bootstrap aggregation method, the highest F1 score is more likely to be found when different numbers of estimators are tested.

- **Evaluating the Classification Model:** The performance of the classification model can be evaluated using a previously unseen dataset. The training and validation sets are already used for creating the model, while the test set is completely new for the model. By classifying fish cropped from images in the test set and comparing them to their manually assigned class, the

[5] Wikipedia, Correlation. `https://en.wikipedia.org/wiki/Correlation#Sample_correlation_coefficient` (As of June 2023).

performance of the model can be assessed. The model is evaluated similarly as during the development of the model, with regard to the accuracy, precision, recall/sensitivity, specificity, and F1 score. Confusion matrices are also computed.

3.2.2.3 Implementation Aspects

- Classification Algorithms
  - **Feature Extraction:** A convolutional neural network (CNN) model was used for extracting features from images. We used a pre-trained model ResNet-101, which is a 101-layer deep CNN trained on more than a million images from the ImageNet[6] collection (containing 1000 classes). This CNN model extracted feature vectors of size 2048 from each image.
  - **Dimensionality Reduction:** ResNet-101 is a general model not specific to this problem, and the vectors might therefore still contain a lot of features that are redundant or unimportant for the classification decision. In our study, we consider the use of principal component analysis (PCA) for dimensionality reduction. By reducing the dimension of the features, the classification process becomes faster as well as the model is expected to be better suited for preventing overfitting. The number of input features should be less than the number of images, which is 1194 for the training set and 346 for the validation set. For both sets, we used the 100 principal components that give the highest explained variance in the sets. The feature vectors for the samples were scaled using the standardScaler[7] function from Scikit-learn, which centralises the feature vectors around 0 with a standard deviation of 1.
  - **Classification Pipelines:** Classification pipelines enable cross-validation for multiple steps, enable tuning of different parameters, and improve the performance of classifiers. Five classification pipelines were evaluated. The first classification pipeline uses a linear perceptron which divides the classes using a single straight line and is sufficient if the classes are linearly separable. The second classification pipeline uses Adaline, which is similar to the perceptron, except that the weights for the learning phase are adjusted based on the inputs instead of the output of the function. The third classification pipeline uses C-support vector classification (SVC), which tries to maximize the margin and incur a penalty when a sample is misclassified, using a one-versus-one approach for multi-class classification. The fourth classification pipeline uses the K-nearest neighbors (KNN) classifier, which performs voting for each class based on Euclidean distances from the N closest samples to the new sample. Lastly, the fifth classification pipeline uses multi-layer perceptron (MLP), which divides the classes using multiple layers

[6]ImageNet. `http://www.image-net.org` (As of June 2023).

[7]Scikit-learn, sklearn.preprocessing.StandardScaler. `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html` (As of June 2023).

and non-linear activation functions, which enables it to separate data that is nonlinear.

All the classification pipelines used a min-max scaling function, which scales and translates each feature to a range between 0 and 1, as well as using a normalisation procedure for centralising the feature vectors around 0 with a standard deviation of 1. A trained classification pipeline with optimised parameters will be referenced to as a *classifier*.

- **Grid Search:** To identifying the best-performing classification algorithm, different configuration parameters should be tested for each classification pipeline. The exhaustive search method gridSearchCV [8] from scikit-learn was used for this purpose, using 5 folds for cross-validation as well as specified parameter values for each estimator. This method has functions to fit the model to training data, evaluate the score of the model, and predict the label of new samples. When training the classifiers through the fit method in gridSearchCV, the desired scoring parameter can be specified.

  All the classifiers used PCA and tested different sizes of principal components as parameters for the gridSearchCV method, of 50, 75, 100, and 125 principal components. For the SVC classifier, it was additionally tested four different kernel types (kernel), linear, rbf, poly, and sigmoid, and 5 different values for the regularisation parameter (C): 0.001, 0.01, 0.1, 1.0, and 10. For the KNN classifier, it was tested 5 different sizes of neighbors (N): 3, 5, 7, 9, 11. The MLP classifier used a hidden layer size of 100 with a maximum of 200 iterations, and 4 different activation functions (identity, logistic, tanh, relu), and 3 different solver methods (lbfgs, sgd, and adam) were considered. The implementation of Perceptron, Adaline, SVC, MLP, and PCA used a random state of 42.

  The classifiers tried to optimise the parameters for two different scoring methods: accuracy and F1 scores in the classification of loser fish. Each classifier was trained twice optimising for both accuracy and F1 score respectively. For each training sequence, the classifiers used the same parameters. The parameter values giving the highest combined scoring for both accuracy and F1 score for each classifier were saved and stored for later use, as well as the corresponding score for accuracy, precision, recall, sensitivity, and F1 score.

- Diversity of Classification Algorithms

  - **Compare Diversity:** Using the diversity measure of the sample correlation coefficient in Equation 3.7, pairwise agreements are put in a 5×5 matrix, to display the level of agreement between each of the five classifiers. Additionally, another such matrix is created, but instead of measuring the agreement of the predictions, we measure the agreement regarding correct predictions.

[8] Scikit-learn, sklearn.model_selection.GridSearchCV. `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`. (As of 27. June 2023).

- **Compare Classifier Pairs:** Using the correlation measure of dos Santos et el. [35] a graph was computed showing the best combination of classifier pairs. This graph displays which classifier pair has the highest average F1 score relative to their calculated correlation.
- **Visualise Each Classifiers Predicted Label:** For understanding the classifiers even better, each prediction for the validation set was explored. Each image was therefore displayed with its correct label and the predicted label of each classifier. Hence, it became easier to see which images were harder to classify, how many samples classifiers assigned correct labels, and for which ones classifiers made similar predictions. Every image that was wrongly classified by all classifiers or for which only one classifier disagreed with the others was displayed with their corresponding label for further inspection. Similarly, some of the images properly labelled by all classifiers are also displayed.

• Ensemble Methods

For improving the classification results of the model further, the results for each classification algorithm were combined.

- **Voting Ensemble:** Voting-based ensemble methods were included for combining the results of the different classifiers' predictions. The hard voting ensemble combines the results of the five classifiers (perceptron, Adaline, SVC, KNN and MLP). By using hard voting, each classifier provides a label for each image, and the final label assigned will be the one related to the class receiving the most votes in total. Different weights were assigned to the algorithms based on their accuracy and diversity, for optimising the ensemble method. All combinations of the five classifiers were tested with different weights between zero and five.

  Soft voting was also used, with the algorithms able to classify using percentages (SVC, KNN, and MLP). These algorithms calculate the probability of how likely the image is to belong to each class, and the image is classified as belonging to the class receiving the highest probability score combined. All combinations of the three possible classifiers were tested with different weights between zero and five. The combination of weights giving the highest F1 score was then identified.
- **Bootstrap Aggregation:** The ensemble method, called bagging or bootstrap aggregation,[9] creates several instances of a black-box estimator based on random subsets of the original training set and aggregates each of their individual predictions to form a final prediction. This method is used for reducing the variance of a base estimator by introducing randomisation before ensemble.

  For the bootstrap aggregation method, a graph is displayed showing the F1 score for each number of estimators. The estimator range is consisting of these numbers: 20, 40, 60, 80, 100, 120, 140, 160, 180, 200, 220, 240, and 260, where an F1 score is calculated for each of

[9]Scikit-learn, 1.11. Ensemble methods. `https://scikit-learn.org/stable/modules/ensemble.html` (As of June 2023).

them. Here as well, the goal is to get the highest F1 score, as such only the first estimator of multiple possible estimators giving the highest F1 score is stored and displayed.

The bootstrap aggregation was used with two types of models: logisticRegression and gradientBoostingClassifier. LogisticRegression [10] models the probability of an event taking place using a linear combination of one or more independent variables [11]. GradientBoostingClassifier [12] for binary classification allows for optimisation of arbitrary differentiable loss functions, where a single regression tree is fit on the negative gradient of the loss function.

- Classification Model

  The test set was used for evaluating the performance of the classification model. Each manually detected fish from the 21 test set images was cropped into new images. Then, 210 new images were created, from which features were extracted before they were classified using the best-performing classification method. For each image, the predicted class was compared with the ground-truth.

3.2.3 Fish Counting

This section presents the evaluation protocol of the third component of the proposed framework, fish counting. It refers to the RQ_5 (Section 1.2, RQ_5 – *What is the performance of the proposed loser fish counting system?*)

3.2.3.1 Dataset

For the counting part, the test set is used. The number of images used is shown in Table 3.2.1. All automatically detected fish as well as ground-truth detected fish from the test set were cropped and used as input of the classification model. This dataset consists of correspondingly 196 images of automatically detected fish and 210 images of manually pre-detected fish.

3.2.3.2 Evaluation Measurements

In order to evaluate the counting system, the actual number of loser fish in the test set should be known. The number of loser fish in the automatically detected test set was compared with the amount of loser fish in the manually pre-detected test set. Annotations were defined by experts.

[10] Scikit-learn, sklearn.linear_model.LogisticRegression. `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html` (As of June 2023).

[11] Wikipedia, Logistic regression. `https://en.wikipedia.org/wiki/Logistic_regression` (As of June 2023).

[12] Scikit-learn, sklearn.ensemble.GradientBoostingClassifier. `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html` (As of June 2023).

3.2.3.3 Implementation Aspects

The fish detection model was used to automatically detect fish in the test set. The detected fish were cropped into new images, and features were extracted for all of them, similar to the procedure for the training and validation sets used for training the classification model. Each image was then classified as *healthy* or *loser* fish using the classification model which received the highest F1 score. The amount of loser fish was obtained by counting the number of fish predicted as a loser. The manually detected fish were also manually assigned a class, such as the amount of loser fish manually detected was obtained by counting and adding each fish manually classified as a loser.

CHAPTER

FOUR

RESULTS AND DISCUSSION

This chapter presents the results for the fish detection (Section 4.1), fish classification (Section 4.2), and fish counting (Section 4.3) methods and provides a discussion about the obtained results.

4.1 Fish Detection

This section contains the results and discussion for the first part of the framework, fish detection.

4.1.1 Fish Detection: Qualitative Analysis

Figure 4.1.1 shows a test image where a bounding box is created around all detected objects with at least a 60% certainty of being a fish. For this image, each detected object has a certainty between 68% and 99%. We also observe that all the detected objects are fish, but there are also more fish in the image that were not detected. The definition of the certainty percentage regulates the trade-off between the number of fish and false positives. In the performed experiments, 60% was identified as a reasonable value for avoiding false positives.

In most cases, the biggest fish was the easiest to detect. For each image, the detected bounding boxes and pre-detected bounding boxes are displayed with their overlap percentage as seen in Figure 4.1.2. Here, the predicted fish is highlighted in yellow, and the ground-truth bounding box is highlighted in white. The percentage of overlap is displayed in the center of the two corresponding bounding boxes in orange. There are six fish from the ground-truth that were correctly predicted. One from the ground-truth was not detected by the algorithm. We can also observe that two correct detections were not annotated. This means that there are some ground-truth fish that are not found by the object detection model, as well as the model is able to find some fish that were not annotated. One of the predicted fish, as seen at the bottom of the image, has received an overlap score of 21%. This is because there is a ground-truth fish that is overlapping, but since it is not the same fish, the overlap is small.

There are also some indications of fish that were not detected nor annotated, such as in the middle of the image, which is zoomed in and outlined in Figure 4.1.3.

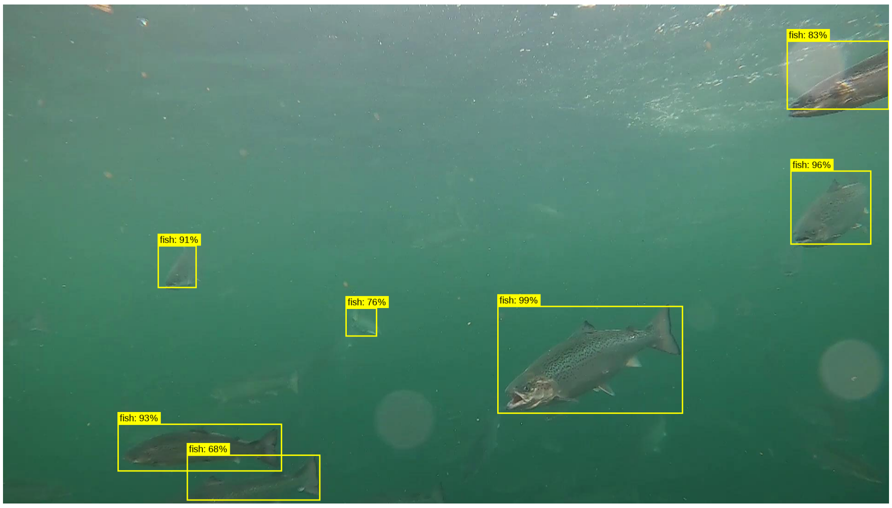


Figure 4.1.1: Detected fish with their certainty of being a fish.

Figure 4.1.2: The overlap of detected and ground-truth fish.

Figure 4.1.3: Undetected fish in object detection.

With no overlap, i.e., cases for which the percentage is zero, the percentage score is not displayed.

4.1.2 Fish Detection: Quantitative Analysis

The accuracy of the object detection pipeline is affected by both the quality of the object detection model and the accuracy of the previous annotations. In total, for all the images in the test set, the object detection pipeline was able to find 196 predicted fish, as shown in Table 4.1.1. Additionally, 146 out of 210 ground-truth fish were detected. This leads to an accuracy of $\frac{146}{196} = 74.5\%$.

Table 4.1.1: Accuracy of the fish detection module.

Fish detection	# of labelled bounding boxes (ground-truth)	# of automatic detections	# of matches
Bounding boxes	210	196	146

4.2 Fish Classification

This section presents and discusses the results for the second part of the framework, fish classification.

4.2.1 Image Cropping: Qualitative Analysis

Once fish are detected, the respective bounding boxes are used to define the region that will be cropped. Figure 4.2.1 shows an example containing an image from the sea cage, containing multiple fish. This image is from the training set, and the fish in the image belongs to the ground-truth. A bounding box is created around each fish, with a colour based on their label. Boxes with a green colour contain a healthy fish, while boxes with a red colour contain a loser fish. The label of each fish is also written in the middle of each bounding box. In this image, we can see that there are seven fish labelled as healthy and one fish labelled as a loser.

Figure 4.2.2 displays two of the cropped images from Figure 4.2.1. Figure 4.2.2a is a cropped image of one of the healthy fish, while Figure 4.2.2b is a cropped image of a loser fish.

Since only fish found in the ground-truth from the train set are cropped, there are some additional fish in the images that are not cropped and therefore not considered in the training process of classification methods. That might affect the quality of the classification model.

4.2.2 On the Use of Principal Component Analysis

Figure 4.2.3 shows the results of using 100 principal components for the training set. The graph in Figure 4.2.3a shows the explained variance ratio in relation to the number of components. By reducing the number of input features from 1194 to 100 and thereby using only 8.38% of the feature extracted training data, a variance of 85.14% is preserved. This means that by using only a small proportion of the dataset, a large amount of information is preserved, making the classification model almost as good as before, and more effective. Figure 4.2.3b shows the

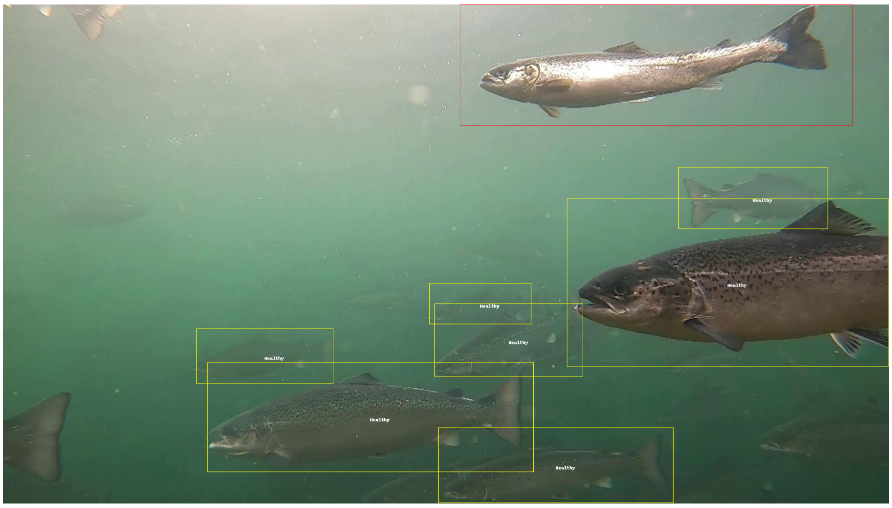
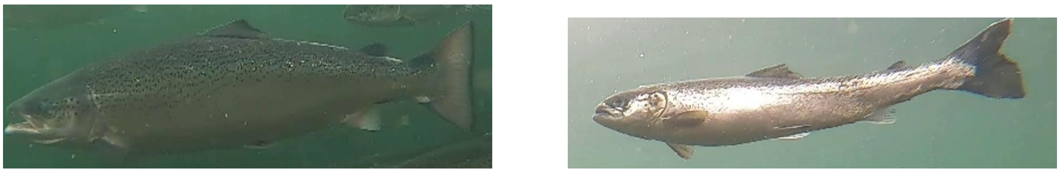


Figure 4.2.1: An image from a sea cage with bounding boxes showing healthy and loser fish.



(a) A healthy fish. **(b)** A loser fish.

Figure 4.2.2: Cropped images of healthy and loser fish.

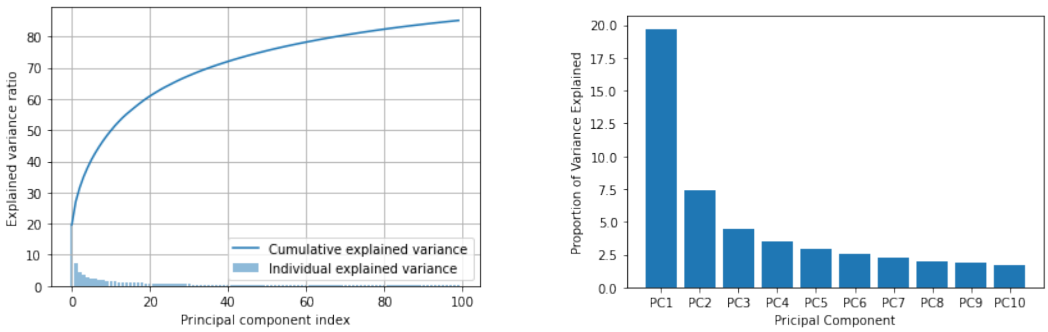
variance of the 10 most significant components, whereas the two most significant components alone represent 27.1% of the variance in the features.

Similarly, the results of using 100 principal components for the validation set are shown in Figure 4.2.4. Figure 4.2.4a shows the cumulative explained variance using up to 100 components for the validation set. By reducing the number of input features from 346 to 100, using only 28.90% of the feature extracted validation data, a variance of 90.02% is preserved. Figure 4.2.4b shows the variance of the 10 most significant components for the validation set, where the two most significant components represent 26.7% of the variance in the features for the validation set.

4.2.3 On the Assessment of Multiple Classifiers

For each of the classifiers, two criteria were assessed: accuracy and F1 scores for loser fish. For both perceptron and Adaline, the highest scores were achieved using a number of 100 principal components. For SVC, a number of 125 principal components, a c-value of 10, and the rbf kernel led to the best results. KNN got the highest scores using 125 principal components and 3 neighbours, while MLP used 100 principal components, logistic activation function, and Adam solver. Their scores for the training and validation sets changed, however.

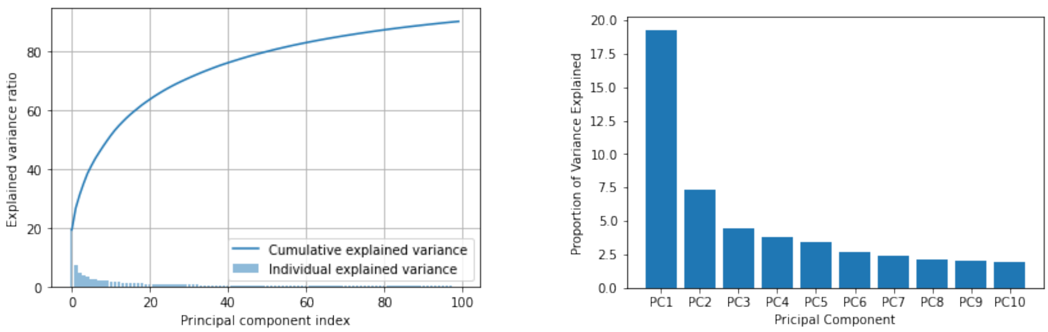
The results for the accuracy score can be seen in Table 4.2.1, and the results for the F1 score can be seen in Table 4.2.2. The effectiveness measure scores for the training and validation indicate if the classification model is overfitting the training data. Concerning accuracy, the scores for the training set are not that



(a) Highest variance using 100 components for the training set.

(b) Variance in the 10 most significant components for the training set.

Figure 4.2.3: Variance explained for the training set using 100 principal components.



(a) Highest variance using 100 components for the validation set.

(b) Variance in the 10 most significant components for the validation set.

Figure 4.2.4: Variance explained for the validation set using 100 principal components.

different from the scores for the validation set. The results in the training set were however more accurate for all the classifiers, especially for the SVC and MLP classifiers where the training set is fitting to 99.8% and 99.9% of the data. Large differences between the scores for training and validation data were not observed. In general, the results suggest that overfitting is not an issue.

Table 4.2.1: Training and validation **accuracy** scores for evaluated classifiers.

Accuracy	Perceptron	Adaline	SVC	KNN	MLP
Training score	91.2	93.0	99.8	93.4	99.9
Validation score	90.2	88.4	91.6	85.8	91.9

Table 4.2.2: Training and validation **F1** scores for evaluated classifiers considering the *loser* class.

F1 score for loser	Perceptron	Adaline	SVC	KNN	MLP
Training score	83.1	85.9	99.7	85.8	99.8
Validation score	79.0	75.6	81.5	66.2	82.7

The five classifiers' scores for accuracy, precision, recall/sensitivity, specificity, and F1 score were all similar regardless of the scoring method during training. Those scores can be seen in Table 4.2.3. The results of the classifiers are evaluated against each other for each row. The cells highlighted in green show the best performance for each evaluation method. Red cells refer to the worst effectiveness scores, while yellow ones relate to medium performance. All numbers are given in percentages. The MLP classifier performed the best for 3 out of 5 methods and also yielded the highest F1 score, 82.7%. SVC also performed well for precision and specificity. KNN performed the worst for the same evaluation metrics for which MLP was the best. KNN yielded the lowest F1 score of the classifiers, with 66.2%. Adaline performed the worst according to the precision and specificity metrics, the ones for which SVC performed the best.

Table 4.2.3: Accuracy, precision, recall/sensitivity, specificity, and F1 scores for the five classifiers in the validation set.

Method	Perceptron	Adaline	SVC	KNN	MLP
Accuracy	90.2	88.4	91.6	85.8	91.9
Precision	84.2	79.5	90.1	81.4	88.2
Recall/Sensitivity	74.4	72.1	74.4	55.8	77.9
Specificity	95.4	93.8	97.3	95.8	96.5
F1 score	79.0	75.6	81.5	66.2	82.7

Figure 4.2.5 shows the confusion matrix of the MLP classifier, which yielded the best results of all the classifiers. This classifier misclassifies 19 images of loser fish as healthy fish and misclassifies 9 images of healthy fish as loser fish. The confusion matrices for the four other classifiers can be found in Appendix B.

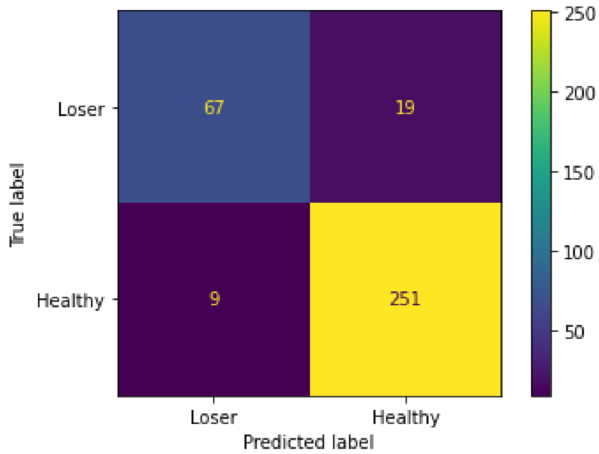


Figure 4.2.5: Confusion matrix results for the MLP classifier.

4.2.4 On the Assessment of Classifiers' Diversity

Figure 4.2.6 shows the agreement matrix for all combinations of pairs for the five classifiers. The matrix is mirrored through the diagonal line. Classifier 1 corresponds to perceptron, classifier 2 to Adaline, classifier 3 to SVC, classifier 4 to KNN, and classifier 5 to MLP. A correlation pair of value 1 indicates a complete correlation, and a value of 0 has no correlation, also indicated by colour in a range from white to dark blue.

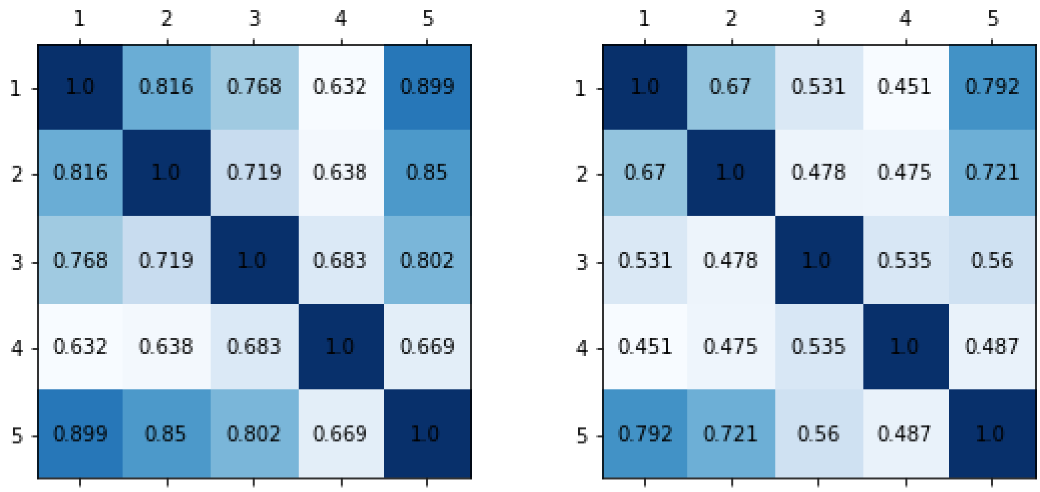
Figure 4.2.6a shows the correlation when comparing the classifiers' label predictions. The graph shows that all the combinations consisting of classifier 4 had the lowest correlations, whereas the correlation between classifiers 1 and 4 was the lowest of them all, and the correlation between classifiers 2 and 4 was the second lowest.

Figure 4.2.6b shows the correlation when comparing whether the classifiers are predicting the correct label. Here, classifiers 1 and 4 still had the lowest correlation, while classifiers 2 and 4 had the second lowest correlation. However, now, classifiers 2 and 3 were associated with the third lowest correlation. The highest correlation was still observed by classifiers 1 and 5, the second one was still classifiers 2 and 5. Now, classifiers 1 and 2 were the third most correlated.

All the correlations are lower when considering the correct predictions of classifiers. This suggests that if the classifiers misclassifies, they have a higher chance of agreeing with each other. Further investigations are needed to discover the benefits of combining the classifiers.

Figure 4.2.7 shows the results observed for pairs of classifiers in terms of their correlation and average F1 scores. The classifiers' correlations are the same as earlier, but now their combined scores can be assessed as well. Classifiers 3 and 5 received the highest combined F1 score, while classifiers 1 and 5 had the second highest, and classifiers 1 and 3 had the third highest. On the other end, classifiers 2 and 4 received the lowest average F1 score, while classifiers 1 and 4 received the second lowest. Since pairs with low correlation and high F1 scores are the most promising for being combined, the results suggest that classifiers 1, 2, 3, and 5 could be considered.

Figure 4.2.8 shows the predicted label for each classifier for each image in the



(a) Correlation based on correct and incorrect predictions.

(b) Correlation based on correct predictions.

Figure 4.2.6: Correlation between classifiers. Classifier 1 corresponds to perceptron, classifier 2 to Adaline, classifier 3 to SVC, classifier 4 to KNN, and classifier 5 to MLP.

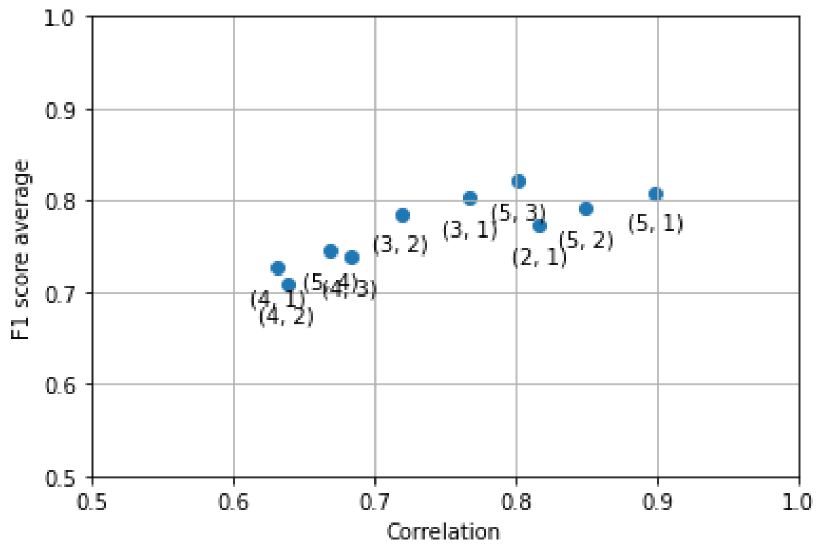


Figure 4.2.7: Correlation versus average F1 score for pairs of classifiers. Classifier 1 corresponds to perceptron, classifier 2 to Adaline, classifier 3 to SVC, classifier 4 to KNN, and classifier 5 to MLP.

validation set, as well as the images' correct label. Each healthy fish is indicated with a white circle, and each loser fish is indicated with a yellow circle. Each column is an image, and each row is a classifier prediction for each image. The green circles indicate that the classifier has predicted the correct class label for an image, while the red circle indicates a wrong prediction. Each row has 44 images, such that 8 tables were required for visualising every image from the validation set.

Based on the predictions, if any classifier assigns a wrong label for an image, there is a high chance that other classifiers also misclassify the same image. These incorrect predictions are mostly for the loser fish. In fact, we can observe that, for many of the loser fish, all five classifiers are incorrect. This indicates that loser fish is harder to classify for all classifiers than healthy fish.

4.2.4.1 Correctly Classified Images: Qualitative Analysis

Figure 4.2.9 shows images containing healthy fish, which have been correctly classified by all five classification algorithms. In total, there are 230 images. In the figure, however, only 21 are displayed. The images are of different conditions and quality regarding water, light, distance, and resolution, which affect how easily the fish are to be detected. Fish in clear water, high light intensity, high resolution, and fish swimming alone are likely to be easier to detect and classify than fish in poorer conditions.

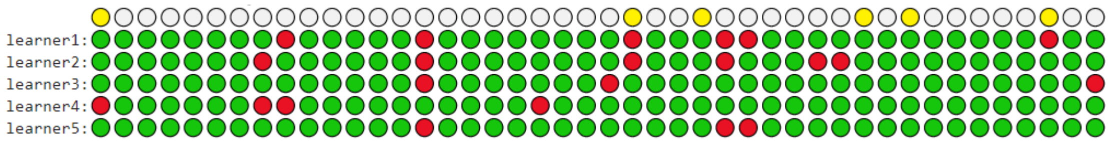
The images displayed in Figure 4.2.10 depict loser fish, which were correctly classified by all the five classification algorithms. There are in total 41 images, but here as well, only 21 are displayed. These images also present different conditions whereas the quality seems to be higher than for the images containing healthy fish in Figure 4.2.9. This might indicate that the images of loser fish require a higher quality for being correctly classified by all classifiers.

4.2.4.2 Partially Correctly Classified Images: Qualitative Analysis

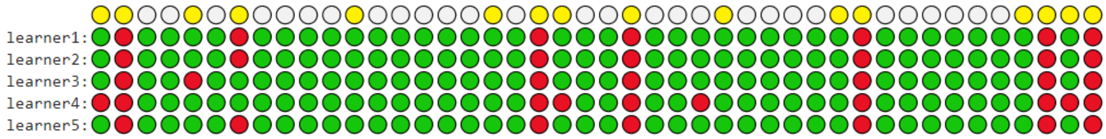
All the images displayed in Figure 4.2.11 are of healthy fish, which were correctly classified by four of the five classification algorithms. There are 14 of these images. For three of those images, the perceptron is the only classifier that classified wrongly. For five of the images, Adaline classified wrongly, for two images, SVC was wrong, and KNN classified wrongly four images. MLP did not classify any of them wrongly.

The images displayed in Figure 4.2.12 are of loser fish, which were correctly classified by four of the five classification algorithms. There are in total 18 images. The perceptron is the only classifier that classified wrongly one of them. Adaline also classified wrongly one of them, while SVC misclassified two of them. Here, KNN stands out quite significantly, with 14 images wrongly classified. Similarly as for the healthy images in Figure 4.2.11 MLP did not misclassify any of them.

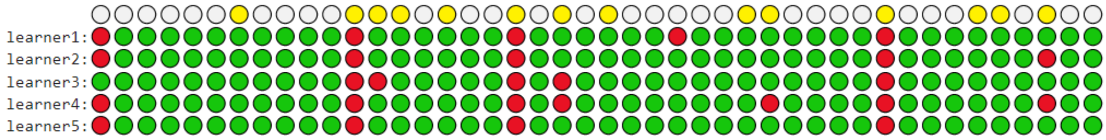
All the images displayed in Figure 4.2.13 were correctly classified by only one of the classification algorithms. There are in total 7 images, consisting of both healthy and loser fish. One of the images is correctly classified as a loser fish by Adaline, four of the images are correctly classified as a loser fish by SVC, one image is correctly classified as a healthy fish by SVC and one image is correctly classified as a healthy fish by KNN. Those results suggest that SVC might be the



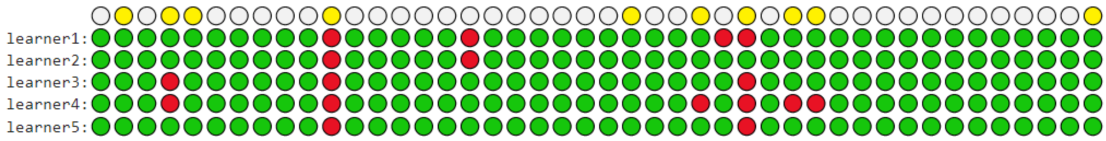
(a) Visualised correctness for image 1-44.



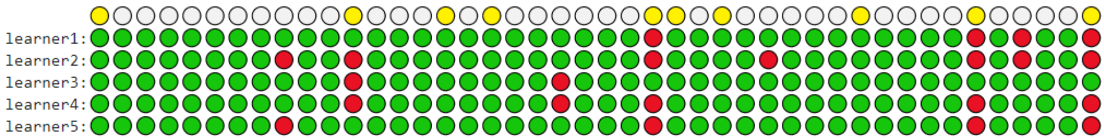
(b) Visualised correctness for image 45-88.



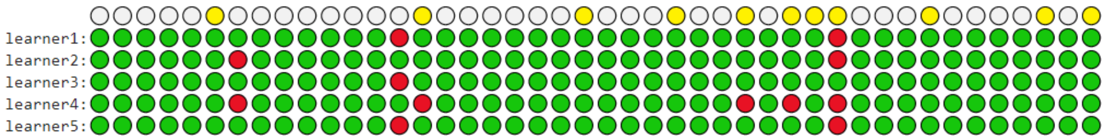
(c) Visualised correctness for images 89-132.



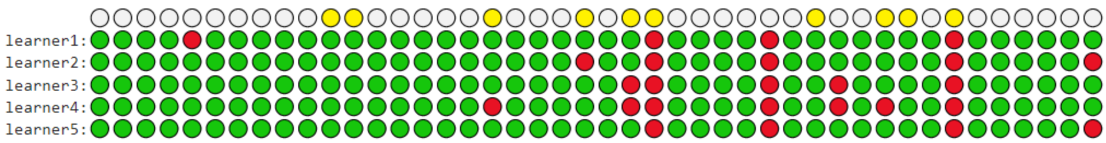
(d) Visualised correctness for images 133-176.



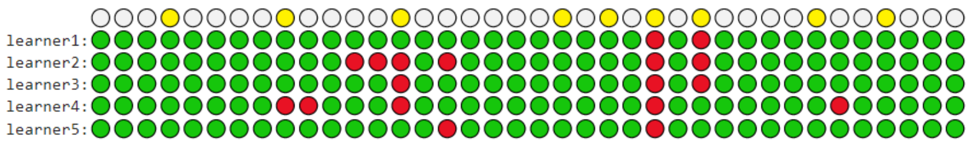
(e) Visualised correctness for images 177-220.



(f) Visualised correctness for images 221-264.



(g) Visualised correctness for images 265-308.



(h) Visualised correctness for images 309-346.

Figure 4.2.8: Classifiers' predictions for different images of the validation set.

(a) Labelled as healthy.

(b) Labelled as healthy.

(c) Labelled as healthy.

(d) Labelled as healthy.

(e) Labelled as healthy.

(f) Labelled as healthy.

(g) Labelled as healthy.

(h) Labelled as healthy.

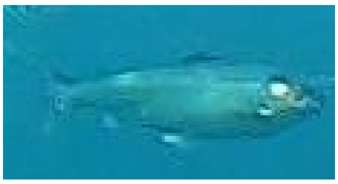
(i) Labelled as healthy.

(j) Labelled as healthy.

(k) Labelled as healthy.

(l) Labelled as healthy.

(m) Labelled as healthy.



(n) Labelled as healthy.

(o) Labelled as healthy.

(p) Labelled as healthy.

(q) Labelled as healthy.

(r) Labelled as healthy.

(s) Labelled as healthy.

(t) Labelled as healthy.

(u) Labelled as healthy.

Figure 4.2.9: Healthy fish correctly classified by all five classifiers.

(a) Labelled as loser.

(b) Labelled as loser.

(c) Labelled as loser.

(d) Labelled as loser.

(e) Labelled as loser.

(f) Labelled as loser.

(g) Labelled as loser.

(h) Labelled as loser.

(i) Labelled as loser.

(j) Labelled as loser.

(k) Labelled as loser.

(l) Labelled as loser.

(m) Labelled as loser.

(n) Labelled as loser.

(o) Labelled as loser.

(p) Labelled as loser.

(q) Labelled as loser.

(r) Labelled as loser.

(s) Labelled as loser.

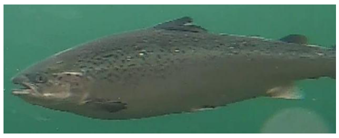
(t) Labelled as loser.

(u) Labelled as loser.

Figure 4.2.10: Loser fish correctly classified by all five classifiers.

(a) Wrongly classified by perceptron.

(b) Wrongly classified by perceptron.



(c) Wrongly classified by perceptron.

(d) Wrongly classified by Adaline.

(e) Wrongly classified by Adaline.

(f) Wrongly classified by Adaline.

(g) Wrongly classified by Adaline.

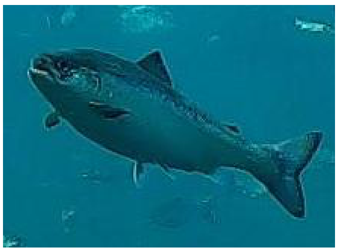
(h) Wrongly classified by Adaline.

(i) Wrongly classified by SVC.

(j) Wrongly classified by SVC.

(k) Wrongly classified by KNN.

(l) Wrongly classified by KNN.



(m) Wrongly classified by KNN.

(n) Wrongly classified by KNN.

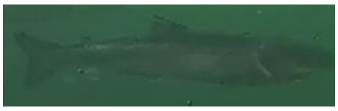
Figure 4.2.11: Healthy fish wrongly classified by one classifier.

(a) Wrongly classified by perceptron.

(b) Wrongly classified by Adaline.

(c) Wrongly classified by SVC.

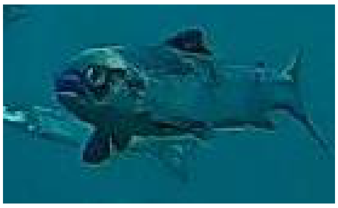
(d) Wrongly classified by SVC.



(e) Wrongly classified by KNN.

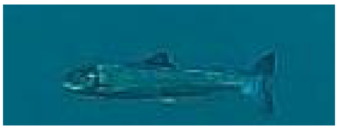
(f) Wrongly classified by KNN.

(g) Wrongly classified by KNN.



(h) Wrongly classified by KNN.

(i) Wrongly classified by KNN.



(j) Wrongly classified by KNN.

(k) Wrongly classified by KNN.

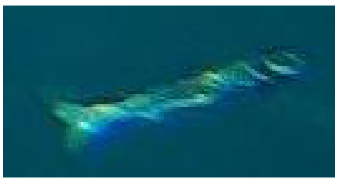
(l) Wrongly classified by KNN.

(m) Wrongly classified by KNN.

(n) Wrongly classified by KNN.

(o) Wrongly classified by KNN.

(p) Wrongly classified by KNN.

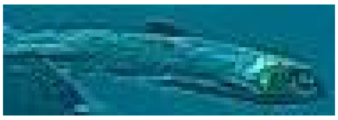


(q) Wrongly classified by KNN.

(r) Wrongly classified by KNN.

Figure 4.2.12: Loser fish wrongly classified by one classifier.

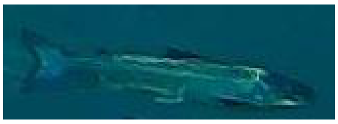
best-suited classification algorithm for classifying correctly when the classifiers predict differently. None of the images were classified correctly by only MLP. This together with the results that none images are wrongly classified by only MLP might indicate that MLP has the least diversity from the other classifiers.



(a) Correctly classified by adaline as loser.

(b) Correctly classified by SVC as loser.

(c) Correctly classified by SVC as loser.



(d) Correctly classified by SVC as loser.

(e) Correctly classified by SVC as loser.

(f) Correctly classified by SVC as healthy.

(g) Correctly classified by KNN as healthy.

Figure 4.2.13: Fish wrongly classified by four classifiers.

4.2.4.3 Wrongly Classified Images: Qualitative Analysis

Figure 4.2.14 depicts fish that were incorrectly classified by all the five classification algorithms. There are 14 of these images, consisting of both healthy and loser fish. Since only one of these images is of a healthy fish, loser fish might be harder to classify for all of the classification algorithms. This might be also due to the fact that the dataset contains more images of healthy fish, such that the classification model has been able to train more on detecting healthy fish than loser fish.

4.2.5 On the Assessment of Ensemble Methods

The accuracy, precision, recall, specificity, and F1 scores for each ensemble method are shown in Table 4.2.4. For each row, the results for each ensemble method are evaluated against each other. The cell containing the best performance for each evaluation method is highlighted in green. All numbers are given in percentages.

(a) Labelled as loser.

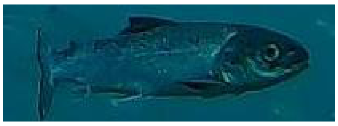
(b) Labelled as loser.

(c) Labelled as loser.

(d) Labelled as loser.

(e) Labelled as loser.

(f) Labelled as loser.



(g) Labelled as loser.

(h) Labelled as loser.

(i) Labelled as loser.

(j) Labelled as loser.

(k) Labelled as loser.

(l) Labelled as healthy.

(m) Labelled as loser.

(n) Labelled as loser.

Figure 4.2.14: Fish wrongly classified by all five classifiers.

Hard voting performed the best for 3 out of 5 methods and also yielded the highest F1 score, 85.0%. Bootstrap aggregation using logistic regression performed the best for precision and specificity.

Table 4.2.4: Accuracy, precision, recall/sensitivity, specificity, and F1 scores for the four ensemble methods in the validation set.

Method	Hard Voting	Soft Voting	Logistic Regression	Boosting Classifier
Accuracy	92.8	92.2	91.6	91.3
Precision	87.7	88.3	90.1	87.8
Recall/Sensitivity	82.6	79.1	74.4	75.6
Specificity	96.2	96.5	97.3	96.5
F1 score	85.0	83.4	81.5	81.2

4.2.5.1 Voting Ensembles: Hard and Soft Voting

For hard voting, the best weight combination for the five classifiers giving the best results was using weight 1 for perceptron, 0 for Adaline, 2 for SVC, 0 for KNN, and 1 for MLP. So, even though MLP had the best result on its own, it is not the most important classifier in the ensemble. This might be since perceptron and MLP has the highest correlation, so their results often overlap with each other. SVC had a lower correlation with both perceptron and MLP and might therefore be more important for complementing the views provided by those in the ensemble.

Figure 4.2.15 shows the confusion matrix which received the highest F1 score for hard voting. Here, 71 out of 86 loser fish were classified correctly, and 250 out of 260 healthy fish were correctly classified. The differences from the results observed for the MLP classifier (see Figure 4.2.5) were that now 4 more loser fish were correctly classified while 1 less healthy fish is classified correctly. This is considered a relevant improvement given the fact that the main goal is to detect as many loser fish as possible.

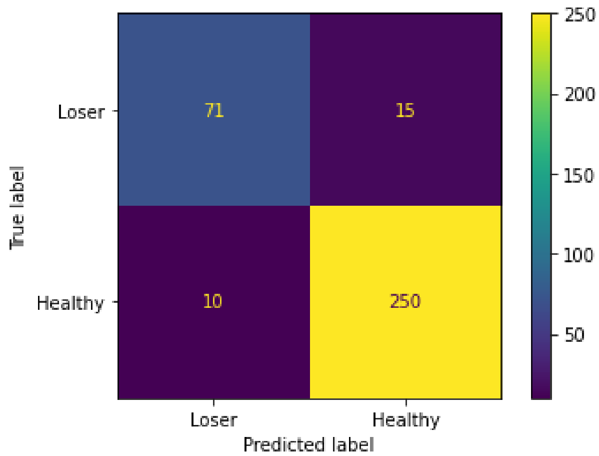


Figure 4.2.15: Confusion matrix for hard voting.

For soft voting, the best combination relied on the use of 1 for SVC, 0 for KNN, and 4 for MLP. Since perceptron is not included here, MLP's results did not overlap much with those of other classifiers. MLP can, therefore, be considered the most important classifier for this ensemble.

Figure 4.2.16 shows the confusion matrix for soft voting, which is among those with the highest observed F1 scores. Here, 18 images of loser fish were misclassified as healthy fish, while 9 images of healthy fish were misclassified as loser fish. This is slightly improved from the MLP classifier, with one more correctly classified image. It is, however, considered worse than the classification for hard voting, since it misclassified more loser fish images.

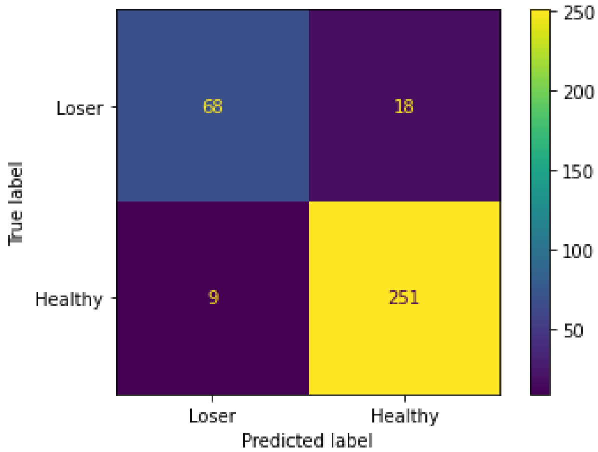


Figure 4.2.16: Confusion matrix for soft voting.

4.2.5.2 Bootstrap Aggregation: Logistic Regression and Gradient Boosting Classifier

Figure 4.2.17 shows the graph that encodes how the number of estimators of the ensemble method affects the received F1 score. The graph consists of a single line for the whole range of 20-260 estimators. This implies that every number of estimators is just as good as each other and gives the same result for an F1 score of 81.5%.

Figure 4.2.18 shows the confusion matrix of bootstrap aggregation using logistic regression with the highest F1 score. This ensemble method seems to be much better at classifying healthy fish correctly, with only 7 misclassified images of healthy fish, but worse at classifying loser fish correctly, with 22 misclassified images of loser fish. Even though this method seems promising, it is considered worse for the purpose of this research, i.e., to detect loser fish.

Figure 4.2.19 shows the graph that encodes how the number of estimators of the ensemble method affects the achieved F1 score. The graph is fluctuating up and down between an F1 score of 80.0% and 82.5% for the range of 20-260 estimators. The highest F1 score was observed when using 40 estimators. This is, therefore, considered the most optimal number of estimators for this method.

Figure 4.2.20 shows the confusion matrix of bootstrap aggregation with boosting classifier receiving the highest F1 score. This method misclassified 21 images of loser fish and misclassified 9 images of healthy fish. It is thereby the ensemble method that classified the most images incorrectly. It however still classified

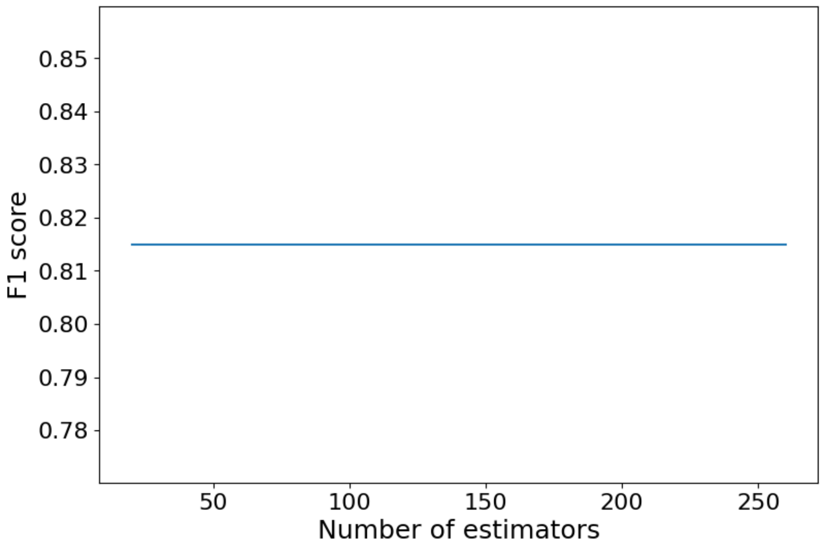


Figure 4.2.17: Optimal number of estimators for logistic regression.

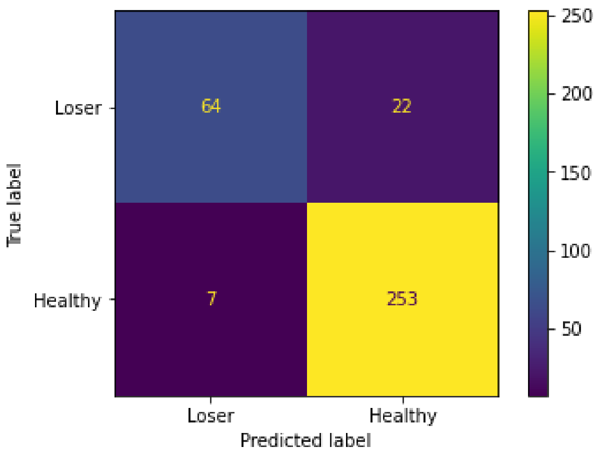


Figure 4.2.18: Confusion matrix for logistic regression.

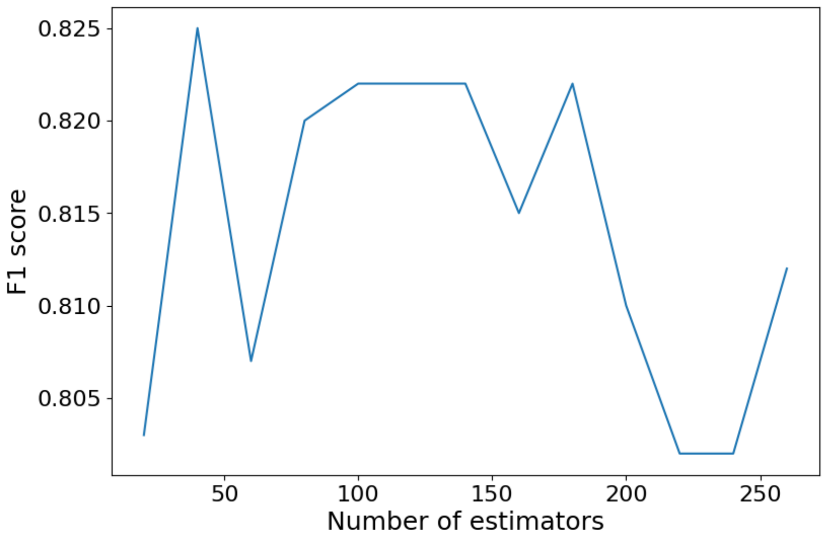


Figure 4.2.19: Optimal number of estimators for gradient boosting classifier.

more loser fish correctly than the method using bootstrap aggregation with logistic regression.

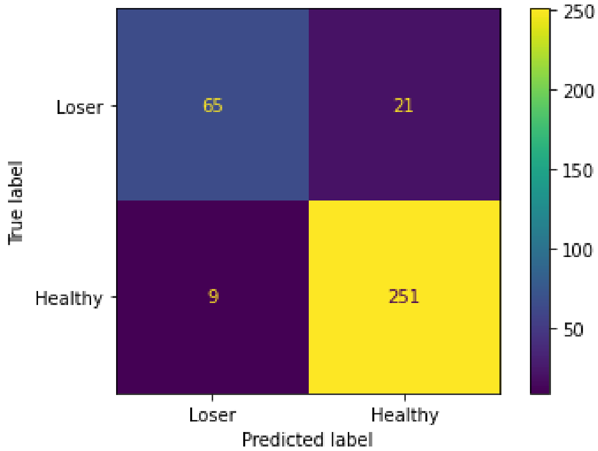


Figure 4.2.20: Confusion matrix for gradient boosting classifier.

4.2.6 Fish Classification: Quantitative Analysis

The classification model is evaluated using the test set, and the methods performing the best for the validation data. Figure 4.2.21 shows the main steps for the construction of the final classification model. The model was built from top to bottom, following the steps highlighted in pink. First, features are extracted using a ResNet-101 model pretrained using ImageNet. Then each of the five classifiers was trained using grid search for optimising parameter values by optimising F1 score in order to classify loser fish. Each of the classifiers was trained considering minMaxScaler, standardScaler, and PCA. Lastly, the classifiers were combined using a voting ensemble, with hard voting. Here, perceptron and MLP received the same weight, and SVC received twice the weight of perceptron and MLP.

Table 4.2.5 shows the results of the evaluation of the classification model on the test set. The results are quite similar to the scores achieved by the model using the validation set. The F1 score reduced from 85% to 81%.

Table 4.2.5: Accuracy, precision, recall/sensitivity, specificity, and F1 scores for the **test** set.

Evaluation Metric	Test set
Accuracy	92.9
Precision	80.0
Recall/Sensitivity	82.1
Specificity	95.3
F1 score	81.0

The confusion matrix for the evaluation of the model can be seen in Figure 4.2.22. We can observe that only 7 loser fish are misclassified as healthy fish,

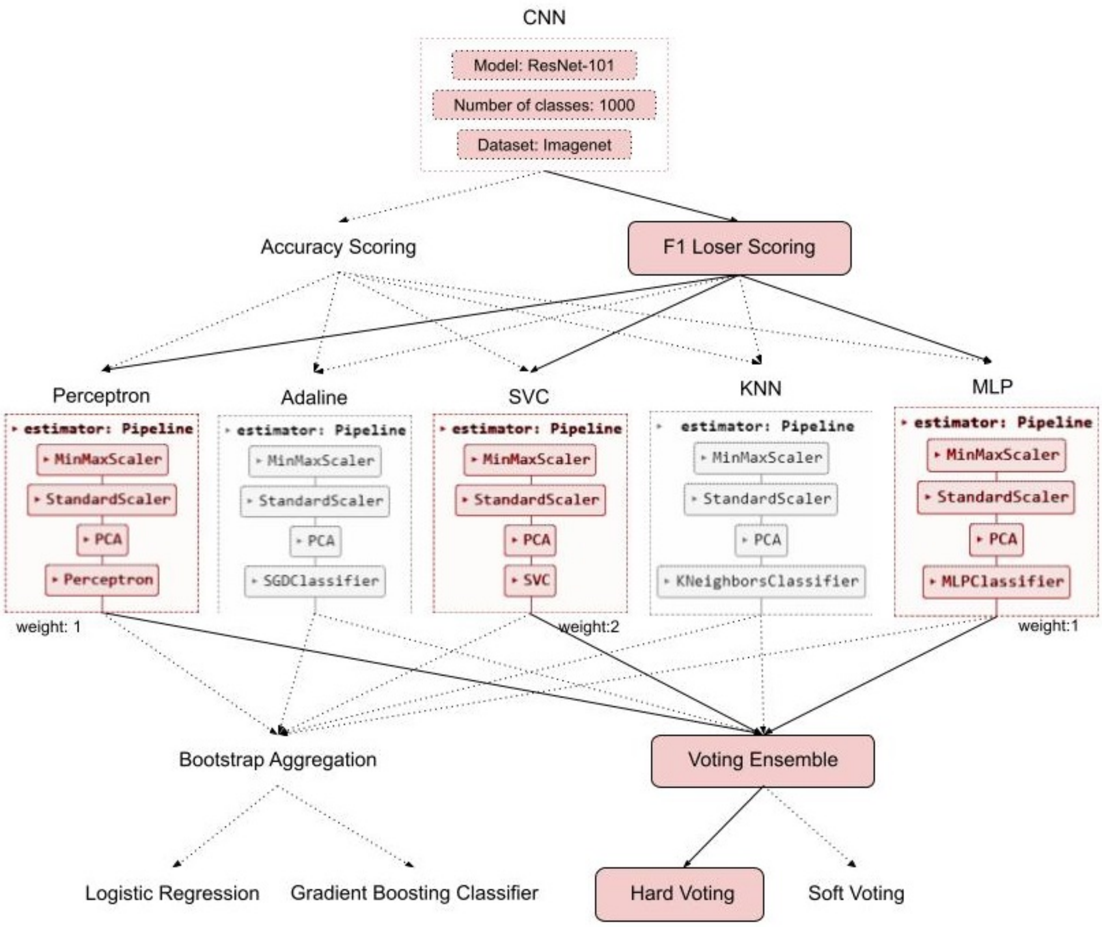


Figure 4.2.21: Main steps for the definition of the classification model.

and 8 healthy fish are misclassified as loser fish. The test set is quite small, consisting of only 210 fish images. The results are however promising, showing that the classification model achieved promising results on completely unseen data.

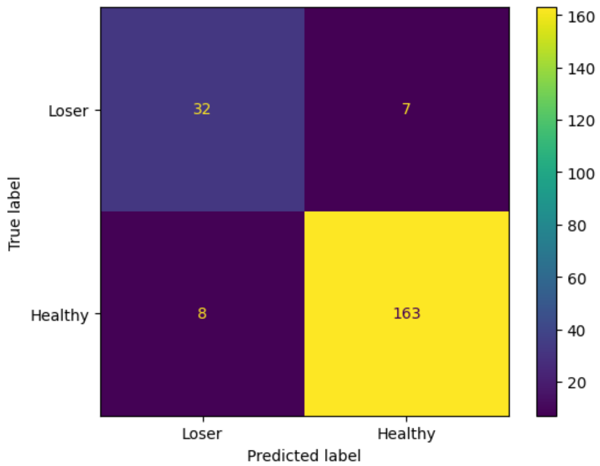


Figure 4.2.22: Confusion matrix for test set.

4.2.6.1 Wrongly Classified Fish: Qualitative Analysis

For the test set, consisting of 21 images and 210 manually detected fish, 15 fish were classified wrong. A selection of some of them are shown in Figures 4.2.23, 4.2.24, 4.2.25, 4.2.26, and 4.2.27. Each ground-truth fish is shown with a bounding box around it as well as its manually assigned label. Each wrongly classified fish is additionally displayed with a white dashed circle around it. In Figure 4.2.23, 1 out of 4 detected fish were misclassified. The fish were given a label of healthy when they should have been classified as loser. In Figure 4.2.24, only 1 fish was labelled as loser, and was the only fish that the model misclassified. The image is quite blurry, which might have affected the model's decision. Figure 4.2.25 contains 4 misclassified fish, whereas one of them is loser and 3 are healthy. In this case, the fish are swimming in groups, such that some of the bounding boxes contain parts of other fish as well as the targeted fish. This might have affected the classification model. Figure 4.2.26 contains 2 misclassified fish, which are both healthy. One of them is displayed from the front, which might make the class distinction decision harder. In Figure 4.2.27, one loser fish was misclassified. Overall, the model seems to be promising at classifying even for challenging scenarios: noisy image quality, fish swimming in groups, fish appearance from different angles, and fish at different distances.

4.3 Fish Counting

This section contains the results and discussion for the third part of the framework, fish counting.

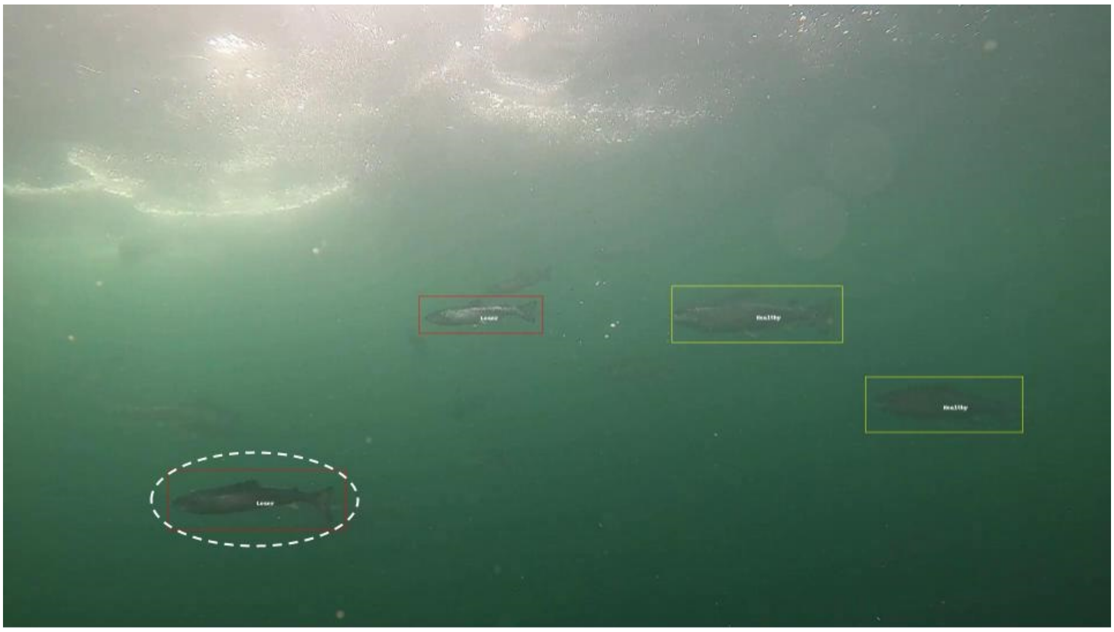


Figure 4.2.23: One wrongly classified fish.

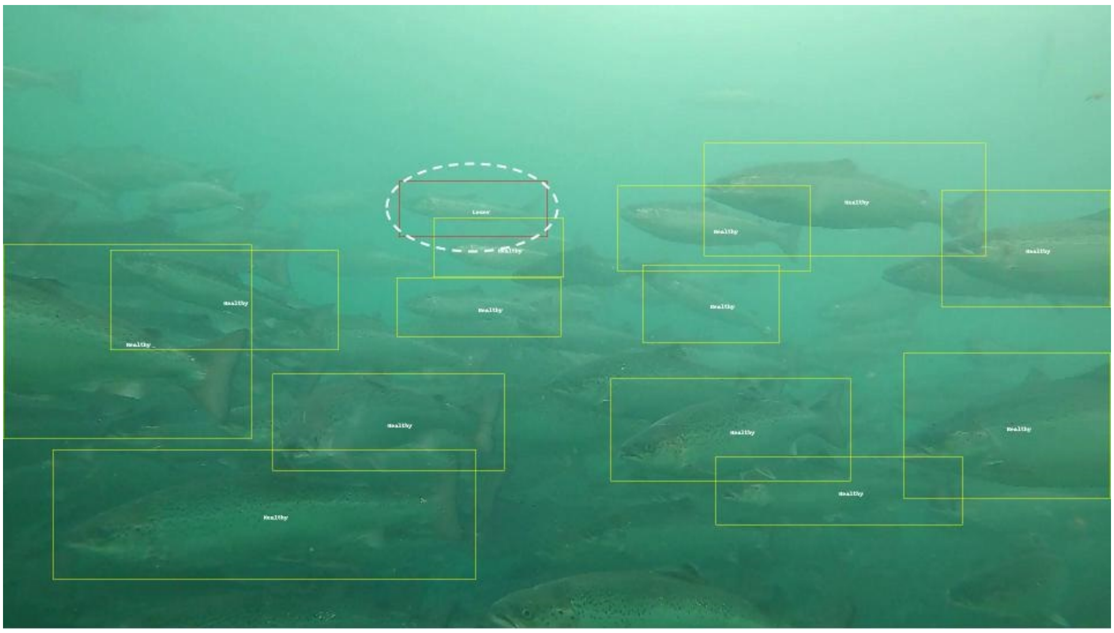


Figure 4.2.24: One wrongly classified fish.

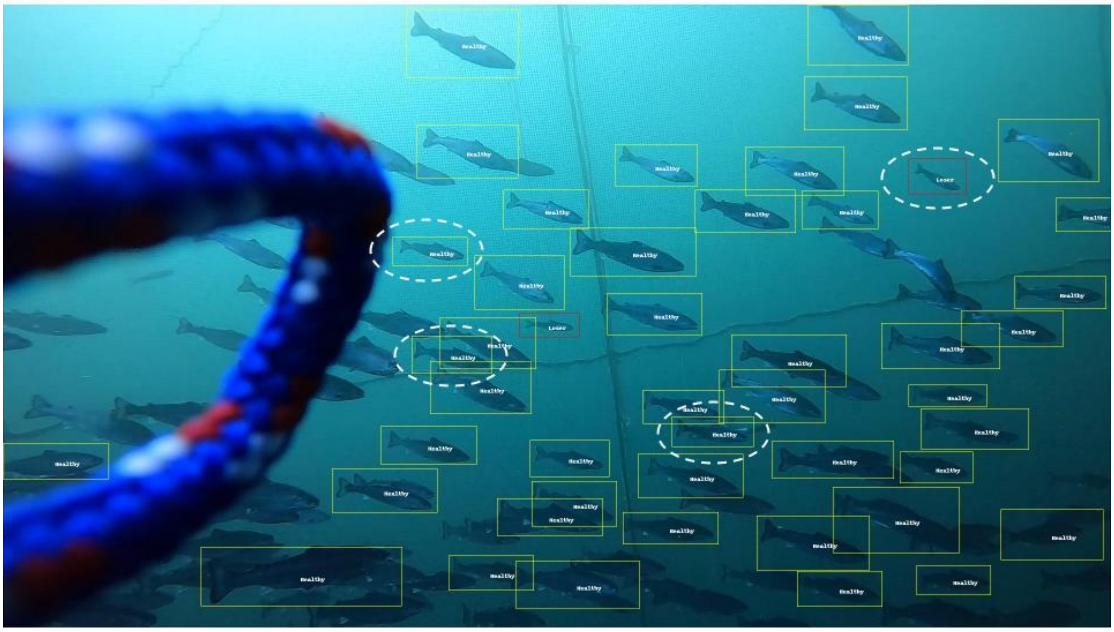


Figure 4.2.25: Four wrongly classified fish.

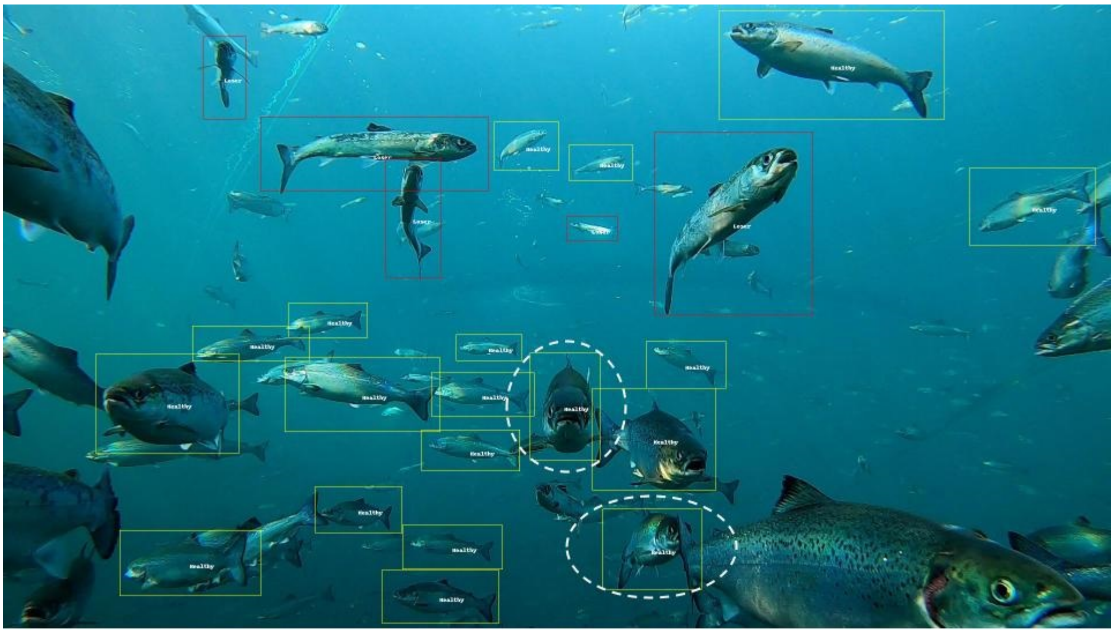


Figure 4.2.26: Two wrongly classified fish.

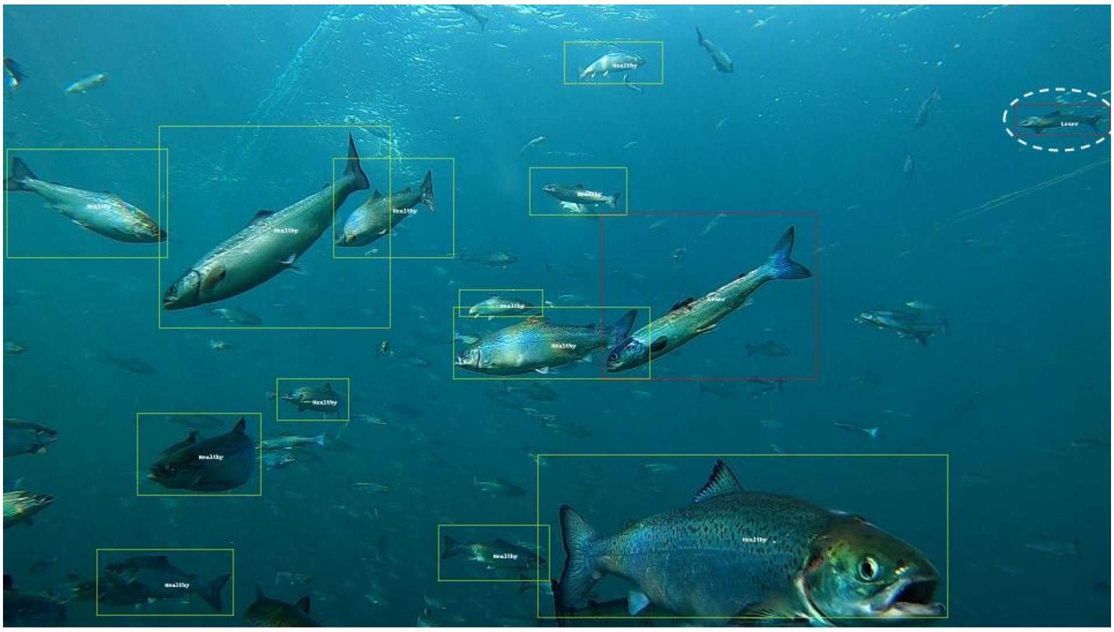


Figure 4.2.27: One wrongly classified fish.

4.3.1 Fish Counting: Quantitative Analysis

The assessment of the counting system relied on the ground-truth defined for the test set. Using the fish detection pipeline, fish were automatically detected from the test set. The automatically detected fish were classified using the fish classification pipeline.

The amount of fish predicted for each class, compared to the amount of fish manually assigned to each class, can be seen in Table 4.3.1. The automatic fish detection model found 196 fish in the test set images, while 210 fish were annotated in the ground-truth. Out of these, 73 fish were classified as loser fish, using the classification model, and 39 were labelled as loser fish in the ground-truth.

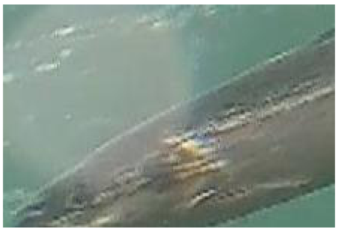
Even though 73 and 39 loser fish are quite a big difference, it does not necessarily mean that the counting pipeline is not performing well, i.e., counting an incorrect number of loser fish. First, manual classification is challenging, and some of the manual annotations might therefore be incorrect. Secondly, from the detection model, it was discovered that the automatic and manual detection only detected 146 of the same fish. This means that they have detected only 74.5% of the same fish. That means that the automatic detection contains 50 fish that was not included in the ground-truth, and that the ground-truth contains 64 fish that the automatic detection did not find. The correct class for these extra automatically detected fish is not known, and therefore the correct number of loser fish is unknown. Since the classification model yielded an accuracy of 92.9% for the test set, it is believed that about the same accuracy is obtained when classifying in the counting system.

4.3.2 Automatically Detected Fish Classified as Loser

Even though the correct class for each automatically detected fish is hard to verify, the correct detection of fish can more easily be validated. Figure 4.3.1 shows some

Table 4.3.1: Counting of loser fish using the test set.

Fish Counting	# of loser fish	# of fish	% of loser fish
Automatic	73	196	37.2
Manual	39	210	18.6

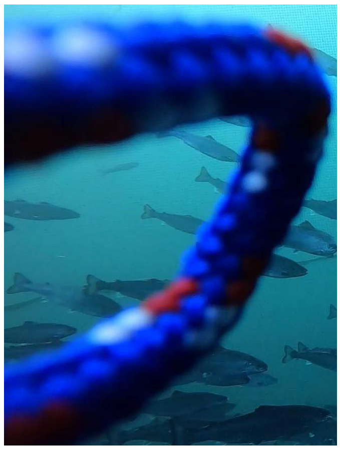


(a) Classified as loser.

(b) Classified as loser.

(c) Classified as loser.

(d) Classified as loser.



(e) Classified as loser.

Figure 4.3.1: Automatically detected fish classified as loser.

of the objects predicted as loser fish. While most of them are in fact fish, only some parts are detected or the quality of the image is poor, making the classification hard. This is however only the case for a few objects, which could have been improved if the dataset consisted of enough images to train the detection model on similar images of salmon in sea cages.

CHAPTER

FIVE

CONCLUSIONS

This chapter summarises the main findings of this work and points out directions for future work.

5.1 Summary of Findings

The objective of this study was to design, implement, and validate an automated system for monitoring the amount of loser fish in sea cages. The occurrence of loser fish can be an important indicator for establishing and improving fish welfare, discovering diseases, or identifying improper conditions related to the environment. With the use of underwater cameras mounted in sea cages, images of fish can be collected, and single fish can be detected using object detection. Later, detected fish can be classified using machine learning, and fish classified as loser can be counted.

In the following, we revisit the raised research questions, indicating associated key findings:

- RQ_1 – What is the performance of a state-of-the-art detector for the fish detection problem?

  This question relates to the fish detection step. Using the test set and a pre-trained fish detection model, we tested the model by Aoi [6]. The goal was to investigate the potential of using pre-trained models for the problem, a common scenario when the amount of labelled data available for training is not high. The method was able to detect 74.5% of the same fish as the ground-truth (see the results reported in Section 4.1). That is a promising result as it suggests it can be worthwhile to replace manual identification with an automatic approach to ensure an effective and reliable method for detecting fish in sea cages.

- RQ_2 – Which state-of-the-art classification algorithms has the highest effectiveness performance for the fish classification problem?

  This question relates to the fish classification step. Here, the performance of five different classification algorithms was tested using a grid search procedure. The Multi-Layer Perceptron (MLP) classifier yielded the best performance with an accuracy of 91.9% (see the results reported in Section 4.2.3).

These results indicate that the MLP classifier is highly accurate at classifying fish as *healthy* and *loser*. By combining it with the detection method, the MLP classifier becomes a sensible method for finding loser fish in sea cages.

- RQ_3 – How much diversity is there between the classification algorithms used in the fish classification problem?

  This question also relates to the fish classification step. The goal of this investigation was to identify the most promising classifiers to be combined. The perceptron, SVC, and MLP classifiers yielded the highest F1 scores for classification, reaching 79%, 81.5%, and 82.7%, respectively. These classifiers also led to the highest F1 scores when combined with each other. The perceptron and MLP classifiers were, however, highly correlated with each other, making their predictions overlap (see the results reported in Section 4.2.4). This made perceptron, SVC, and MLP promising classifiers for combining, with SVC being the most relevant of them.

- RQ_4 – Which state-of-the-art ensemble method has the highest performance for the fish classification problem?

  This question is also related to the fish classification step. Here, the classification algorithms are combined using different ensemble methods. The best performance was observed for a voting-ensemble using hard voting. This method led to an accuracy of 92.8% using two votes from the SVC classifier and one vote each from the perceptron and the MLP classifiers (see the results reported in Section 4.2.5). This result shows that by combining the classifiers, the classification accuracy was indeed improved from using only the MLP classifier. The hard-voting ensemble method is therefore a promising method for classifying fish accurately.

  The classification model with the highest F1 score consisted of feature extraction using CNN, dimensional reduction using PCA, and a hard voting-based ensemble for combining the classification results of different classification algorithms. This classification model received an accuracy of 92.9% for the test set (see the results reported in Section 4.2.6). These results are even better than during the training of the classification method, indicating that the model is well-suited for classifying new unseen data.

- RQ_5 – What is the performance of the proposed loser fish counting system?

  This question relates to the fish counting part. Here, the object detection model and the best-performing classification model were used. Detected fish were classified as either *healthy* or *loser*. The amount of fish classified as loser was then counted (see the results reported in Section 4.3). Even though the exact performance of the counting method is unknown, the automatic counting method provides a more reliable method for counting loser fish with higher effectiveness and accuracy than manual counting. The proposed framework is therefore an effective tool for monitoring the occurrence of loser fish and therefore for contributing to improving fish welfare in sea cages.

This project has been constrained by the time available for conducting the research. Since this is a master's thesis, the work has to be conducted in a specific amount of time, making it difficult to investigate multiple alternatives for all parts of the envisioned system. A limited number of classification algorithms and ensemble methods have been chosen and investigated. The project was also limited in regard to the amount of available data for training the detection and classification models. All these limitations made it challenging to select few but promising methods for investigation as well as choose appropriate values for parameters and reasonable approaches. Another challenging aspect of the project has been to optimise the machine learning processes for reducing the time consumed for training, without using a too-powerful computer.

5.2 Future Work

Some possible improvements to the already promising counting system can be investigated in future work. If a larger dataset is available, the system can be significantly improved. With more images, the object detection model can be trained on detecting salmon fish in sea cages, instead of using a pre-trained fish detection model, specialised in detecting different kinds of fish species. With an end-to-end detection and classification model, a detection model specifically trained for detecting salmon can be able to detect a larger amount of fish in the images. The classification model itself can also be improved with a larger dataset, as the model would have more images of healthy and loser fish to train on.

The developed system can also be installed in a sea cage to further validate the results achieved in this study in real-world monitoring settings. Complementary analyses based on videos can also be performed. For example, videos from sea cages can be used for assessing fish behaviour similar to the work of Li et al. [22], Wu et al. [23], Zeng et al. [24], and Spampinato et al. [26]. Since loser fish swim differently than healthy fish, the classification of healthy and loser fish could probably have been further improved by this assessment. With the implementation of fish behaviour assessment in videos, fish tracking could also be implemented, similarly to the methods presented in the studies of Spampinato et al. [26], Morais et al. [32], Kandimalla et al. [33]. The goal would be to keep track of which fish have already been classified and counted. The system can also be tested with additional classification algorithms (e.g., Decision Trees[1] or Naïve Bayes[2] classifiers), and other ensemble methods [36]. Another research venue refers to the use of other consolidated detection methods, such as those based on the YOLO family [37].

[1]Scikit-learn, sklearn.tree.DecisionTreeClassifier. `https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html` (As of June 2023).

[2]Scikit-learn, Naive Bayes. `https://scikit-learn.org/stable/modules/naive_bayes.html` (As of June 2023).

REFERENCES

[1] Kana Banno et al. "Automatic Detection of Growth-Stunted Phenotype in Farmed Atlantic Salmon: A New Insight into Quantify their Distribution and Bahaviour based on a Machine Learning Approach". In: *Aquaculture Europe 2022*. 2022.

[2] Marco A Vindas et al. "Brain serotonergic activation in growth-stunted farmed salmon: adaption versus pathology". In: *Royal Society open science* 3.5 (2016), p. 160030.

[3] Christian Medaas et al. "Minding the Gaps in Fish Welfare: The Untapped Potential of Fish Farm Workers". In: (2021). URL: https://link.springer.com/article/10.1007/s10806-021-09869-w.

[4] Salmon Group. "Fish welfare in fish farming – What is it?" In: (2020). URL: https://salmongroup.no/wp-content/uploads/2020/09/SG_Fiskevelferd_ENG_Digital.pdf.

[5] INSTITUTE OF MARINE RESEARCH. "Topic: Fish welfare". In: (2019). URL: https://www.hi.no/en/hi/temasider/aquaculture/fish-welfare.

[6] AI Aoi. "*fish_ detection*". In: (2018). URL: https://github.com/kwea123/fish_detection.

[7] Cambridge Dictionary. In: (). URL: https://dictionary.cambridge.org/.

[8] Paul J. Ashley. "Fish welfare: Current issues in aquaculture". In: *Applied Animal Behaviour Science* 104.3 (2007). Fish Behaviour and Welfare, pp. 199–235. ISSN: 0168-1591. DOI: https://doi.org/10.1016/j.applanim.2006.09.001. URL: https://www.sciencedirect.com/science/article/pii/S0168159106002954.

[9] Brian Key. "Why fish do not feel pain". In: *Animal Sentience* 1.3 (2016), p. 1. URL: https://www.wellbeingintlstudiesrepository.org/animsent/vol1/iss3/1/.

[10] James D. Rose. "The Neurobehavioral Nature of Fishes and the Question of Awareness and Pain". In: *Reviews in Fisheries Science* 10.1 (2002), pp. 1–38. DOI: 10.1080/20026491051668. URL: https://doi.org/10.1080/20026491051668.

[11] James D Rose. "Pain in fish: Weighing the evidence". In: *Animal Sentience* 1.3 (2016), p. 25. URL: https://www.wellbeingintlstudiesrepository.org/animsent/vol1/iss3/25/.

[12] K.P Chandroo, I.J.H Duncan, and R.D Moccia. “Can fish suffer?: perspectives on sentience, pain, fear and stress”. In: *Applied Animal Behaviour Science* 86.3 (2004). International Society for Applied Ethology Special Issue: A selection of papers from the 36th ISAE International Congress., pp. 225–250. ISSN: 0168-1591. DOI: https://doi.org/10.1016/j.applanim.2004.02.004. URL: https://www.sciencedirect.com/science/article/pii/S0168159104000498.

[13] Rebecca Dunlop and Peter Laming. “Mechanoreceptive and Nociceptive Responses in the Central Nervous System of Goldfish (Carassius auratus) and Trout (Oncorhynchus mykiss)”. In: *The Journal of Pain* 6.9 (2005), pp. 561–568. ISSN: 1526-5900. DOI: https://doi.org/10.1016/j.jpain.2005.02.010. URL: https://www.sciencedirect.com/science/article/pii/S1526590005005250.

[14] C. Noble et al. “Welfare indicators for farmed atlantic salmon: tools for assessing fish welfare”. In: (2018).

[15] RSPCA. “RSPCA welfare standards for farmed Atlantic salmon”. In: (2021). URL: https://science.rspca.org.uk/sciencegroup/farmanimals/standards/salmon.

[16] Randi Nygaard Grøntvedt et al. “Thermal de-licing of salmonid fish–documentation of fish welfare and effect”. In: *Norwegian Veterinary Institutes Report Series* 13 (2015), p. 2015.

[17] Lars H Stien et al. “Salmon Welfare Index Model (SWIM 1.0): a semantic model for overall welfare assessment of caged Atlantic salmon: review of the selected welfare indicators and model presentation”. In: *Reviews in Aquaculture* 5.1 (2013), pp. 33–57.

[18] Marian Stamp Dawkins. “Behaviour as a tool in the assessment of animal welfare1”. In: *Zoology* 106.4 (2003), pp. 383–387. ISSN: 0944-2006. DOI: https://doi.org/10.1078/0944-2006-00122. URL: https://www.sciencedirect.com/science/article/pii/S0944200604701130.

[19] D.M. Broom. “Assessing welfare and suffering”. In: *Behavioural Processes* 25.2 (1991), pp. 117–123. ISSN: 0376-6357. DOI: https://doi.org/10.1016/0376-6357(91)90014-Q. URL: https://www.sciencedirect.com/science/article/pii/037663579190014Q.

[20] Alzayat Saleh et al. “A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis”. In: *Scientific Reports* 10.1 (2020), p. 14671.

[21] Abdullah Al Muksit et al. “YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment”. In: *Ecological Informatics* 72 (2022), p. 101847. ISSN: 1574-9541. DOI: https://doi.org/10.1016/j.ecoinf.2022.101847. URL: https://www.sciencedirect.com/science/article/pii/S1574954122002977.

[22] Xin Li et al. “A novel automatic detection method for abnormal behavior of single fish using image fusion”. In: *Computers and Electronics in Agriculture* 203 (2022), p. 107435. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2022.107435. URL: https://www.sciencedirect.com/science/article/pii/S0168169922007438.

[23] Yao Wu et al. "Locomotor posture and swimming-intensity quantification in starvation-stress behavior detection of individual fish". In: *Computers and Electronics in Agriculture* 202 (2022), p. 107399. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2022.107399. URL: https://www.sciencedirect.com/science/article/pii/S0168169922007074.

[24] Yuhao Zeng et al. "Fish school feeding behavior quantification using acoustic signal and improved Swin Transformer". In: *Computers and Electronics in Agriculture* 204 (2023), p. 107580. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2022.107580. URL: https://www.sciencedirect.com/science/article/pii/S0168169922008882.

[25] Mutasem K. Alsmadi and Ibrahim Almarashdeh. "A survey on fish classification techniques". In: *Journal of King Saud University - Computer and Information Sciences* 34.5 (2022), pp. 1625–1638. ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2020.07.005. URL: https://www.sciencedirect.com/science/article/pii/S1319157820304195.

[26] Concetto Spampinato et al. "Automatic Fish Classification for Underwater Species Behavior Understanding". In: *Proceedings of the First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*. ARTEMIS '10. Firenze, Italy: Association for Computing Machinery, 2010, pp. 45–50. ISBN: 9781450301633. DOI: 10.1145/1877868.1877881. URL: https://doi.org/10.1145/1877868.1877881.

[27] SO Ogunlana et al. "Fish classification using support vector machine". In: *African Journal of Computing & ICT* 8.2 (2015), pp. 75–82.

[28] Daniel Marrable et al. "Accelerating species recognition and labelling of fish from underwater video with machine-assisted deep learning". In: *Frontiers in Marine Science* 9 (2022).

[29] Yuanyang Zhao et al. "LFCNet: A lightweight fish counting model based on density map regression". In: *Computers and Electronics in Agriculture* 203 (2022), p. 107496. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2022.107496. URL: https://www.sciencedirect.com/science/article/pii/S0168169922008043.

[30] Lu Zhang et al. "Automatic fish counting method using image density grading and local regression". In: *Computers and Electronics in Agriculture* 179 (2020), p. 105844. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2020.105844. URL: https://www.sciencedirect.com/science/article/pii/S0168169920320172.

[31] Song Zhang et al. "Automatic fish population counting by machine vision and a hybrid deep neural network model". In: *Animals* 10.2 (2020), p. 364.

[32] Erikson F Morais et al. "Particle filter-based predictive tracking for robust fish counting". In: *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)*. IEEE. 2005, pp. 367–374.

[33] Kandimalla V et al. "Automated Detection, Classification and Counting of Fish in Fish Passages With Deep Learning". In: (2022). URL: https://www.frontiersin.org/articles/10.3389/fmars.2021.823173/full.

[34] Hongkun Yu et al. "*TensorFlow Model Garden*". In: (2020). URL: `https://github.com/tensorflow/models`.

[35] Jefersson A dos Santos et al. "Descriptor correlation analysis for remote sensing image multi-scale classification". In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE. 2012, pp. 3078–3081.

[36] Fabio A. Faria et al. "A framework for selection and fusion of pattern classifiers in multimedia recognition". In: *Pattern Recognition Letters* 39.0 (Apr. 2014). Advances in Pattern Recognition and Computer Vision, pp. 52–64. ISSN: 0167-8655. DOI: `10.1016/j.patrec.2013.07.014`. URL: `http://www.sciencedirect.com/science/article/pii/S0167865513002870`.

[37] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: *arXiv* (2018).

APPENDICES

A Specialisation Project Report

The specialisation project report which is conducted for the same topic.

Machine Learning for Automatically Detecting Loser Fish in Sea Cages

1st Linda Helen Sperre
Department of ICT and Natural Sciences
NTNU
Ålesund, Norway
lindahsp@stud.ntnu.no

***Abstract*—Sea cages often contain a large amount of loser fish. The presence of loser fish may cause welfare problems for both healthy and unhealthy fish. By detecting and counting the occurrence of loser fish, welfare problems can be identified and addressed. For example, the loser fish can be moved or euthanized. The objective of this study is to develop and validate a machine learning solution for detecting and counting the number of loser fish in a sea cage. The fish are detected using an object detection algorithm, with a pretrained fish detection model. The detected fish are classified into classes (healthy or loser fish). The machine learning solutions investigated in our study consider the use of a convolutional neural network feature extractor, principal component analysis for dimensionality reduction, and the combination of different classification methods. This study has resulted in an algorithm able to detect multiple fish from images in a sea cage and classify each of the detected fish with an accuracy of 97.3 %.**

***Index Terms*—machine learning, classification, convolutional neural network, fish detector, support vector machine, principal component analysis, voting-based ensemble, loser fish**

I. Introduction

Welfare corresponds to the physical and mental health and happiness of a being, according to Cambridge University [1]. Good welfare therefore includes that animals are treated well, have a life worth living, and experience a good quality of life, in general without suffering or facing cruel conditions [2]. Since fish are not seen as a sentient being, their welfare has been given less concern than the welfare of other farmed animals [3]. Good fish welfare has however been given more attention in the last years with establishing their sense of pain and consciousness [4]. Fish farmers have as a consequence tried to promote fish welfare, and make the fish thrive, grow, and stay healthy. That, together with laws and regulations, such as the Norwegian Animal Welfare Act, the Aquaculture Act and the Food act, have all contributed to promoting the welfare of farmed fish [2] [4].

The fish welfare is affected by a lot of parameters, such as oxygen, feeding strategy, diseases, amount of fish in the cage, the environment, their safety and so on [5]. All these factors make it hard to evaluate fish welfare in a standardised way. As there is no way to measure what fish experience, we have to rely on indirect measuring methods. Fish needs have to be fulfilled in order to have a positive impact on their welfare, and to avoid a negative one. Their welfare is assumed to be closely affected by the coverage of their needs, and is indicated by their growth, health and physiological function, and behaviour [5]. According to Noble et al. [2], their welfare can be measured by a function-based approach, measuring the health, growth and performance of the fish; a nature-based approach, where they have a natural environment and can have innate species-specific behaviour; or a feelings-based approach, emphasising the emotions, such as long lasting negative emotions or the experiencing of pleasure. There is a lot of overlap among these three approaches, which makes it very complex and difficult to determine how to best measure and assess animal welfare [2].

About 60 million fish (15-20%) die before they are big enough to be slaughtered, because of insufficient welfare factors, such as diseases, parasites, or injuries. Among the surviving fish, there are millions of loser fish with welfare problems in the aquaculture industry [5]. These are fish that grow too slowly during their first months in comparison to the rest of the group, probably because of a combination of insufficient welfare factors [3] [4]. They can be recognised by their abnormal behaviour and since they often isolate themselves from the rest of the group close to the surface [2]. Loser fish are seen as unhealthy fish and are unwanted from a production perspective. Even though they may survive for long, they often get diseases and do not have a satisfying life regarding animal welfare [4]. These welfare problems are also resulting in big financial losses and are harming the reputation of the aquaculture industry, leading to the loser fish being sorted away and euthanized if detected by the fish farmers [3] [4] [5]. Figures 1a and 1b show a healthy and loser fish respectively.

Loser fish are normally hard to spot, and only detected manually near the surface [6]. Since fish are often farmed in large populations, it is especially difficult to monitor or treat a single fish. Therefore they are often treated as a population, leading to excess use of medication and handling of animals who do not need it [4]. If treatment was given only to the individuals who need it, the animal welfare would be increased, as well as the environmental impact would be reduced [4]. There are currently no available methods to individually recognise and sort fish, but since a lot of loser fish could remain undetected, especially in deeper water, an automatic method to monitor the fish should be established, in order to improve the productivity and the welfare of farmed fish [6].

The objective of this work is to design, implement, and validate a method for automatically detecting loser fish in a sea cage. This will be achieved by using an object detection algorithm for detecting fish in images from the sea cage, as well as a machine learning algorithm to classify fish as loser or healthy. In this work, I focus on investigating which kind of input gives the best results, find out which pipeline of classification algorithm works the best, and see if the results can be further improved by combining the views provided by different classifiers using a voting-based ensemble.

Section II contains an overview of related work. The section dedicated to materials and methods, Section III, contains a detailed description of what have been done, so that the results can be reproduced. Section IV contains the presentation of results and solutions, and comparisons between them, and presents a discussion regarding the work and proposed solutions based on the defined objective. Section V provides conclusion and outlines possible future work.

(a) A healthy fish.

(b) A loser fish.

Fig. 1: Images of healthy and loser fish.

II. Related Work

Some work has been done toward creating a standardised way to evaluate fish welfare. Stien et al. [7] presented a semantic model, SWIM 1.0, for overall welfare assessment of Atlantic salmon in sea cages. The model was designed to support a formal and standardised assessment of fish welfare using a set of selected welfare indicators. Semantic modelling was used to investigate the known welfare needs of Atlantic salmon with the feasible welfare indicators [7]. These indicators need to be measurable, and able to be divided into levels from good to poor welfare. The indicators were weighted based on literature reviews and based on semantic modelling concepts. The objective was to estimate the indicator impact on welfare. The result of the study was a model designed to calculate welfare indices for salmon in sea cages. The model also identifies how each indicator contributes to the overall index and hence discovers which welfare must be improved.

Folkedal et al. [8] continued the work of Stien at al. by investigating the operational feasibility of the Salmon Welfare Index Model (SWIM 1.0 and 2.0) for welfare assessment of Atlantic salmon in sea cages. Ten salmon farms containing smolts were visited twice to assess the SWIM and to detect the best and the worst welfare status. Multiple welfare indicators were applied, and the ranking of the best and the worst ones were determined. The results of the study concluded that the SWIM model is a promising tool for documenting the animal welfare of farmed salmon, serving as a first step towards standardised monitoring and benchmarking of overall salmon welfare.

Noble et al. [2] introduced a handbook containing tools for assessing fish welfare. They collected information about the welfare of Atlantic salmon in relation to its welfare needs at different life stages, and investigated how each welfare indicator may be linked to specific welfare needs. The handbook also assessed which Operational Welfare Indicators (OWI) and Laboratory-based Welfare Indicators (LABWI) are appropriate and fit for purpose for different production systems and for different husbandry routines and operations. As a conclusion, the handbook suggests a unified scoring system standardised for different welfare indicators to help farmers assess welfare and detect potential welfare problems.

Some have tried to recognise and evaluate fish automatically. Banno et al. [6], for example, created a model for classifying healthy and loser fish. They collected videos from sea cages and labelled images of fish according to the presence of loser and healthy fish. A convolutional neural network (CNN) model was used for extracting features from the images, using a pretrained model which produced vectors with 2048 features. In order to classify the images, a support vector machine (SVM) binary classifier was implemented using grid search to find the best parameters. By doing this, they were able to obtain an accuracy of 97.17% for the test data when classifying the fish images. Their work has not considered the detection problem.

In another study, Al Aoi [9] created a model for fish detection. This model is pretrained on a large dataset of various fish species using Open Images V7[1]. The model creates bounding boxes around detected fish in an image, and can be easily imported into any project. Yu et al. [10], in turn, created an open source framework for detecting a large number of objects in an image, using the TensorFlow library. The framework makes it easy to construct, train, and deploy object detection models.

For my work, I used the images collected from Banno et al. [6], and used part of the CNN model for creating a feature extraction function. I also used the SVM classifier as a starting point for creating a classification algorithm. I used the pretrained fish detection model from Al Aoi [9] for detecting fish in images, but would have made this model myself if I had access to a lot of fish images to train the model upon. The object detection framework from Yu el al. [10] is also used as a starting point for the object detection algorithm, although the framework was not able to detect fish or create new images from bounding boxes.

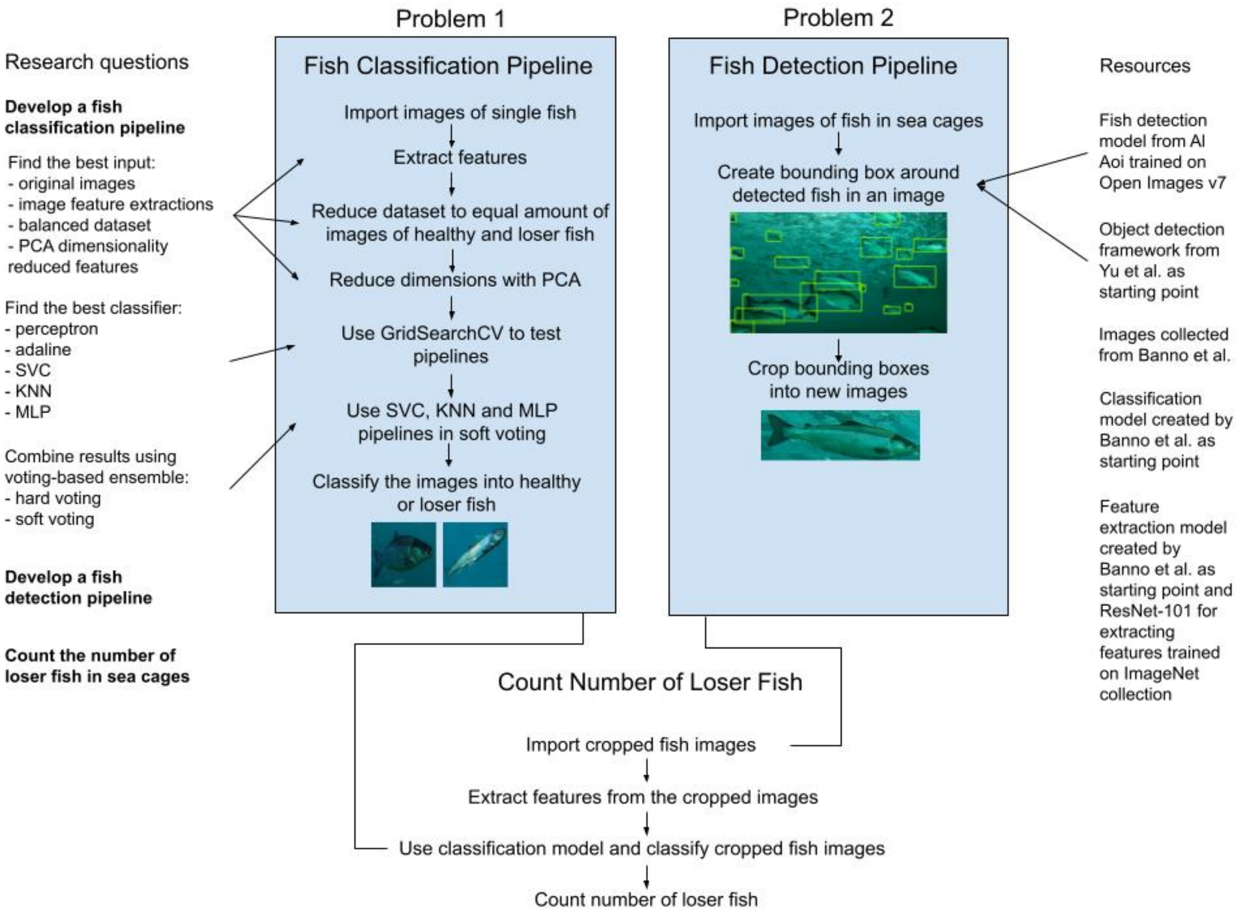


Fig. 2: Schematic view of the methodology adopted in our study.

III. Materials and Methods

A. Schematic of the Solution

Figure 2 shows the research methodology explored in this study. It displays how loser fish in sea cages is intended to be counted. It shows the structure of the two problems investigated in order to count fish: fish detection and fish classification. The research questions investigated are also addressed as well as the resources that have been used.

B. Datasets

The dataset used for Problem 1 was created by Banno et al. [6] and consists of 352 images with a size of 256×256 pixels. In this collection, 227 images are labelled as healthy fish, while 125 images are labelled as loser fish. All the images contain exactly one fish in the middle of the image, and the size of the fish seems to be approximately the same, relative to the size of the images.

For Problem 2, another dataset created by Banno et al. [6] is used. The images in this dataset consists of multiple fish of different sizes and positions in a sea cage, with a size of 1920×1080 pixels. 50 out of the images contain healthy fish, and 50, loser fish.

C. Feature Extraction

A file containing the extracted features for the images, as well as the code for the extraction were also provided by Banno et al. [6]. Using the pretrained model ResNet-101[2], which is a 101-layer deep CNN trained on more than a million images from the ImageNet collection[3]. This model extracts feature vectors of size 2048 from each image.

D. Fish Detection

An object detection method was created to detect multiple fish in an image from the sea cage. By importing the fish detection model from Al Aoi [9], and using the object detection framework created by Yu el al. [10] as a starting point, a fish object detection algorithm was created. By inserting a number of fish images received from Banno et al. [6], the algorithm

[1] Open Images Dataset V7. https://storage.googleapis.com/openimages/web/factsfigures_v7.html#class-definitions. (As of 18. Nov. 2022).

[2] Kaggle, ResNet-10. https://www.kaggle.com/datasets/pytorch/resnet101 (As of 24. Nov. 2022).

[3] ImageNet. http://www.image-net.org (As of 22. Nov. 2022).

created a bounding box around any object that is predicted with at least 50 % certainly to be a fish. The algorithm then creates a new image for every fish in a bounding box with a certain certainty percentage. These new cropped images are stored into a folder, and later imported into the classification algorithm.

E. Fish Classification

All the images are labelled into a value of 0 for healthy fish and 1 for loser fish. The available data was divided into training and testing sets, corresponding to 70 % and 30 % of the data, respectively. The model uses the exhaustive search method GridSearchCV[4], from scikit-learn, which implements a function to fit the model to training data, evaluate the score of the model, and predict the label of new samples.

This search method uses the supervised machine learning algorithm C-support vector classification (SVC), which requires a regularisation parameter (C), kernel coefficient (gamma), and kernel type (kernel). In order to find the best values for C, gamma, and kernel, the refit=True parameter is used in the exhaustive search considering 10 different c-values, 10 different gamma-values, and 4 types of kernels. The model is, in total, fitting 5 folds for each 400 candidates, in total performing 2000 fits. The best fit values for the parameters are then obtained as well as the accuracy, precision, recall, and confusion matrix of the model.

The confusion matrix encodes the results of the model by showing the actual and predicted classes for the dataset. The accuracy, the precision, and the recall are used for evaluating the results and for each, the model is given a percentage score. The accuracy tells how good the model is at classifying the data, the precision tells how many of the positive identifications actually were correct, and the recall encodes how many of actual positives were correctly identified. An F1 score, combining the precision and recall, is also used for evaluating the results. A python program written by Banno et al. [6] is used as a starting point for developing the fish classification model. The classification model is developed using a standard Google Colab session with an Intel(R) Xeon(R) CPU @ 2.20GHz, and around 13 GB of RAM.

F. Find out which Input gives the Best Results

1) Find Input Features: At first, the gridSearchCV method was tested using the full images as input. The method was also tested using a smaller input size, by means of a CNN feature extraction procedure applied on full images. The objective is to improve the results, and classify the images faster. Here a CNN function is created, based on the provided CNN feature extraction code, using the pretrained model ResNet-101. Both the full images and the feature reduced images are imported, before being run through the gridSearchCV method using SVC.

2) Make the Dataset Balanced: In order to make the dataset balanced, an equal amount of images were used for training and testing. Since there were 125 images of loser fish, 102 randomly selected images of healthy fish were discarded. Therefore, we ended up with 125 images left of healthy fish as well. The model is also made balanced by splitting the training and test subsets such that they have the same proportions of class labels as the array containing the labelled images.

3) Parameters for Splitting the Dataset: When splitting the dataset into training and testing, some parameters have to be specified such as random state and the size of the test set. The random state is set to 42, and the test size is 30% of the dataset. The GridSearchCV algorithm also needs some specifications, whereas the number of folds for performing cross validation is set to 4 folds.

4) Check for Overfitting: In order to check if a model is overfitting, e.g., fit the training set too well, so that it does not generalise for the test set, the results for the training set and testing set were also calculated.

5) Dimensionality Reduction with Scaling and PCA: Even though we used vectors with only 2048 features, there could still be many features that are redundant or unimportant for the classification decision, since ResNet-101 is a general model not specific to this problem. By reducing the dimension of the features, the model is expected to be better suited for preventing overfitting and can make the classification process even faster. The number of input features should be less that the number of images, and is reduced to the 150 most significant components. The feature vectors for the samples is also scaled using the standardScaler function, which centralises the feature vectors around 0 with a standard deviation of 1.

G. Adding Multiple Classification Algorithms as Pipelines

In order to improve the classifier, 5 pipelines were evaluated. Those pipelines enable cross validation for multiple steps, enable tuning of different parameters, and improve the structure of the code[5]. The pipelines use minMaxScaler, standardScaler, and principal component analysis (PCA), and is implemented into a gridSearchCV function. Different values for the number of principal components and number of folds for the cross validation are tested to find the most optimal values for each pipeline, and these numbers are saved and stored for later use.

The first pipeline uses a linear perceptron, which divides the classes using a single straight line, and is sufficient if the classes are linearly separable[6]. The second pipeline adaline[7] is similar to the perceptron, except for the fact that it adjusts the weights for the learning phase based on the inputs instead of the functions output. An SVC pipeline with linear kernel uses a one-versus-one approach for multi-class classification where it tries to maximise the margin and incur a penalty when

[4]Scikit-learn, sklearn.model_selection.GridSearchCV. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. (As of 02. Dec. 2022).

[5]Scikit-learn, sklearn.pipeline.Pipeline. https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html (As of 18. Nov. 2022).

[6]Wikipedia, Perceptron, 2022 https://en.wikipedia.org/wiki/Perceptron. (As of 18. Nov. 2022).

[7]Wikipedia, ADALINE, 2022. https://en.wikipedia.org/wiki/ADALINE (As of 18. Nov. 2022).

a sample is misclassified[8]. The K-nearest neighbors (KNN) pipeline votes for the corresponding class based on which class has the most close samples to the new sample[9]. At last, the multi-layer perceptron (MLP) pipeline uses multiple layers and non-linear activation functions, enabling it to distinguish data that is not linearly separable[10].

H. *Combining Results using Voting-based Ensemble*

A voting strategy was investigated in order to improve the classification results of the model. By using the 5 classification algorithms (perceptron, adaline, SVC, KNN, and MLP), the image would be labelled as belonging to the class that most algorithms agreed upon.

By using hard voting, all 5 algorithms will decide which class to classify the given image, and the image will be classified as the class receiving the most votes. Since some of the algorithms are more accurate than the others, different weights are assigned to them based on their accuracy.

Soft voting is also included, whereas only SVC, KNN, and MLP are used. Here the algorithms calculated the probability for how likely the image belongs to each of the classes, and the image is classified as belonging to the class receiving the total highest probability score combined.

I. *Count Number of Loser Fish*

Now that the model is trained to classify healthy and loser fish, it can be used to count the number of loser fish from a set of images. A function called *predict_loser_count* was created for this purpose. This function takes in an array of the cropped fish images from the fish detection algorithm, runs the CNN feature extraction for each of the images, and calls the predict method using the pretrained classification model. It counts how many of the images contains a loser fish, by summing all the images classified with a value of 1. To check the classification, some fish images are printed with their given class.

IV. Results and Discussion

A. *Object Detection*

The object detection algorithm looks for fish in an image, and creates bounding boxes around each detected fish. Figure 3 provides an example of result, in which 18 fish are detected. Each bounding box is displayed with a percentage, showing how certain the algorithm is that the bounding box is surrounding a fish. Only the bounding boxes with a certainty of 50 % or more are displayed in the figure. The algorithm used the dataset for Problem 2 (with 50 images containing healthy fish, and 50 images containing loser fish).

Figure 4 displays a new image, created by cropping the bounding box out from the original image. A new image is created for every bounding box with a certainty of 60 % or more. 1094 new images are created using this cropping procedure.

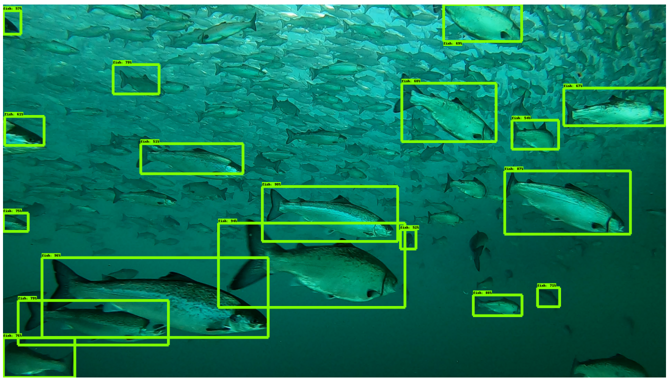


Fig. 3: Bounding boxes around detected fish in an image, with a certainty of 50 % or more.

Fig. 4: Image of a cropped bounding box.

B. *Test Different Inputs for Fish Classification*

The dataset for Problem 1 is extremely small, only consisting of 352 images. This is normally not enough to archive effective results in typical machine vision solutions used in similar problems. Ideally, there should be many more images for the model to train and test upon, but this is not available, so this amount has to be adequately for this model for now.

1) Original Images: This solution considers the full image as feature vector. It can be considered as a baseline. When using the SVM classification code employed by Banno et al. [6] and the original 352 images, the best-performing model had C=0.000001, gamma = 1000 and kernel = linear. The C-value tells how precise the model should be, and hence how much it should avoid misclassifying the samples in the training set. With higher values of C, the optimisation will choose a smaller-margin hyperplane for separating the two classes, if that is better at classifying the training samples correctly[11]. The gamma value defines how far the influence of each training sample reaches, where high values means that the distance is close. The behaviour of the model is very sensitive to the gamma value, in which high values of gamma means that the influence radius of the support vectors only include the support vector itself and no amount of regularisation with C

[8] Scikit-learn, 1.4. Support Vector Machines. https://scikit-learn.org/stable/modules/svm.html (As of 18. Nov. 2022).

[9] Scikit-learn, sklearn.neighbors.KNeighborsClassifier. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (As of 18. Nov. 2022).

[10] Wikipedia, Multilayer perceptron, 2022. https://en.wikipedia.org/wiki/Multilayer_perceptron (As of 18. Nov. 2022).

[11] Scikit-learn, RBF SVM parameters. https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html (As of 18. Nov. 2022).

will be able to prevent overfitting[12]. When the kernel is linear however, the gamma value is not used, and is irrelevant to the model[13]. The kernel being linear means that the hyperplane separating the data into classes is linear, simply consisting of straight lines[14].

The model is able to predict the correct class for 68.9% of the dataset, which is the accuracy of the model. The amount of correctly positive identifications was 45%, which is the model's precision. The recall received a score of 29%, which indicates how many actual positives were correctly identified.

A confusion matrix of the classification model is shown in Figure 5. This shows how many images in the test set were correctly classified. 64 images were correctly classified as healthy fish and 9 were correctly classified as loser fish. 11 images of healthy fish were falsely classified as loser, and 22 loser fish were classified as healthy.

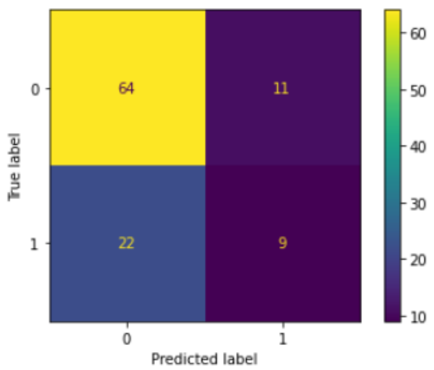


Fig. 5: Confusion matrix of the model using full images.

2) CNN Feature Extraction: By implementing a CNN feature extraction algorithm, the input was reduced to 2048 features. When using these extracted features instead of the full images, the best-performing model had C=10 instead of 0.000001, making the model use a smaller-margin hyperplane, which means that the distance between the hyperplane and the closest point from each class is smaller. The support vector machine (SVM) classification then archives a higher accuracy using a radial basis function (RBF) kernel, which is the default kernel for SVM classification, instead of the linear kernel. Thus the gamma value of 0.01 has become relevant, which regulates the influence distance of each training sample.

The model reaches an accuracy of 94.3%, a precision of 90.3%, and a recall of 90.3%. The confusion matrix has improved significantly, as can be seen in Figure 6. The speed of the model has also improved largely, from using about 5.5 hours for the original images to using about 2 minutes for the extracted images.

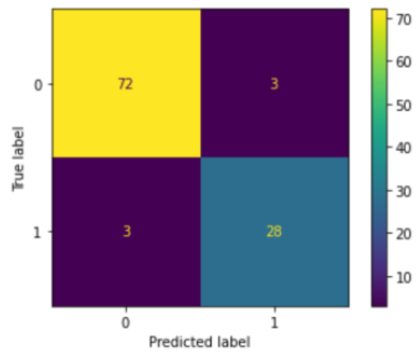


Fig. 6: Confusion matrix of the model using CNN feature extraction.

The distribution of the feature extracted dataset, using 2048 input features, is shown in Figure 7. The x-coordinate represents the first of the 2048 features, while the y-coordinate represents the second of the 2048 features. Here the yellow data points are healthy fish, while the purple ones represent loser fish. The data points are overlapping a lot, making the two classes hard to separate.

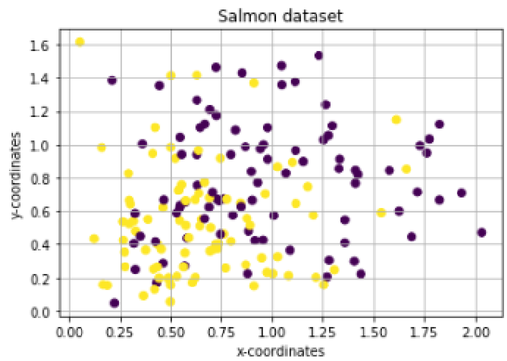


Fig. 7: Plotting of dataset using 2048 features.

3) Balanced Training Set: When using the balanced training set, the highest accuracy was achieved with the use of C=0.1 and with the kernel back to being linear, such that gamma became irrelevant. The dataset was now a lot smaller, but the test set was more equally divided. The accuracy, precision, and recall has all increased to 94.7%.

The confusion matrix, as seen in Figure 8, shows that fewer images are incorrectly classified. It now leads to two false positives and two false negatives, making the difference in amount of misclassified images of false positives and false negatives smaller, indicating that the results of the model are less biased. The results are now better and the model is fairer, making it a better suited classification model.

When using a different amount of images of healthy and loser fish, the model is taught that healthy fish are more

[12]StackExchange, What is the influence of C in SVMs with linear kernel?, 2020. https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel (As of 18. Nov. 2022).

[13]Scikit-learn, sklearn.svm.SVC. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (As of 18. Nov. 2022).

[14]Medium, In Depth: Parameter tuning for SVC, 2018. https://medium.com/all-things-ai/in-depth-parameter-tuning-for-svc-758215394769 (As of 18. Nov. 2022).

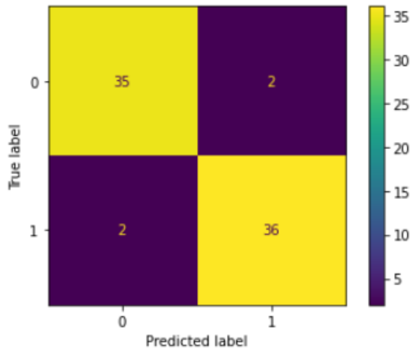


Fig. 8: Confusion matrix of the model when a balanced training set is used.

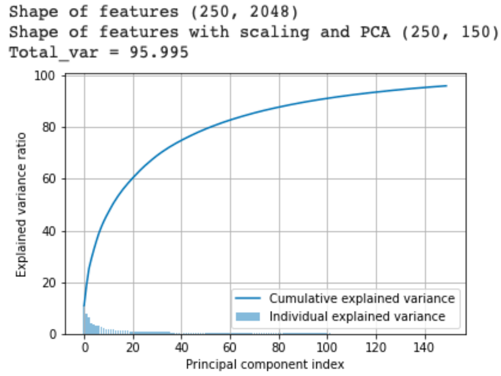


Fig. 9: Explained variance with scaling and PCA of 150 components.

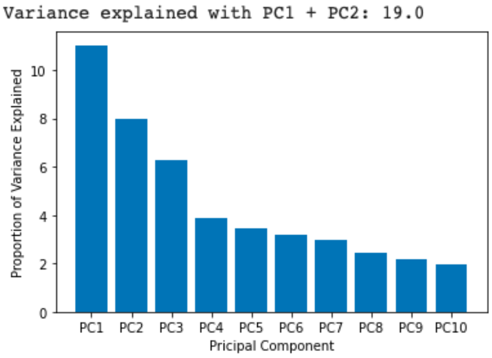


Fig. 10: Explained variance with scaling and 10 most significant PCAs.

common than loser fish, and will favour the healthy images when classifying. Using 125 images of each will make it less biased and the model will classify the healthy and loser fish more evenly, instead of perfecting the classification of one of them. Changing from an unbalanced dataset with an accuracy of 94.3% to a balanced dataset with an accuracy of 94.7% gives a significant improvement.

4) Check for Overfitting: At first, it was believed that the model was overfitting, since it was using a very high value of gamma. This would have made the model unable to generalise for the test set. It was however discovered that the gamma value is not used as long as the model uses linear kernel. Overfitting was also checked by calculating the score of the model for the training set and testing set. If the score for the training set had been high, while the score for the testing set was low, the model would have overfitted. Since they have a score of 100% and 96%, respectively, the difference is quite small, making the model fit well without overfitting.

5) Dimensionality Reduction with PCA: Figure 9 shows the importance of different principal components. By reducing the number of input features from 2048 to 150, about 96% of the information is preserved. This makes the classification model almost as good as before, and much faster. This is therefore a lot better to use if the classification has to happen fast, such as when using live-feed images from sea cages as dataset.

Figure 10 shows the importance of the 10 most significant principal components. The two most significant ones explains 19% of the variance in the dataset.

C. Test Different Pipelines

Table I shows the accuracy, precision, and recall of all the 5 classification methods using gridSearch with PCA. All the methods consider the use of either 50 or 75 features determined by PCA. The green colour in the table shows which of the algorithms is performing best at accuracy, precision, and recall, the red colour is visualising the worst performance, and the yellow colour is for medium performance. All numbers are given in percentages. MLP has the highest accuracy, SVC and KNN have the highest precision, and MLP has the highest recall.

In total, MLP is performing the best, with a F1 score of 95.9%, and is also the only method which is not worst at any of the measurements. With the use of MLP the accuracy of the model has increased from 94.7% to 96%. Precision has also increased, while recall has decreased slightly. The number of components for the MLP is chosen to be 50, which corresponds to about 79% of the total data information. Since the model is only using 50 input features instead of 2048, the speed is increased significantly. This is therefore a much better solution for the purpose of classifying multiple fish at the same time from images in a live-feed. The confusion matrix of the MLP, as seen in Figure 11, is about the same as earlier from figure 8. The only difference is that it classifies one less healthy fish as loser fish.

D. Combine Results using Voting-based Ensemble

There are no large differences between the results of the 5 algorithms, but since different algorithms are best in different areas, it could be worth to assess if the combination of their results would lead to an improved classification. By using hard voting and all the 5 algorithms to classify, the weights for the voting are based on the total performance of the algorithms. The MLP pipeline performed the best and is given a weight of

TABLE I: Accuracy, precision, and recall for the five classifiers.

Method	Accuracy	Precision	Recall	Average	F1 score
Perceptron	94.7	97.1	91.9	94.6	94.4
Adaline	94.7	97.1	91.9	94.6	94.4
SVC	94.7	100	89.2	94.6	94.3
KNN	94.7	100	89.2	94.6	94.3
MLP	96.0	97.2	94.6	96.0	95.9

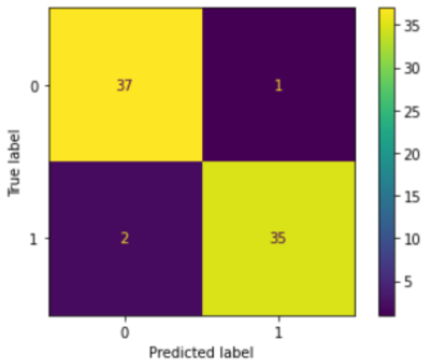


Fig. 11: Confusion matrix of the MLP model when using PCA with 50 features.

3 and the rest a weight of 1. For this voting ensemble all the results have increased, to an accuracy of 97.3%, a precision of 100% and a recall of 94.6%, and is now only classifying 2 images wrongly.

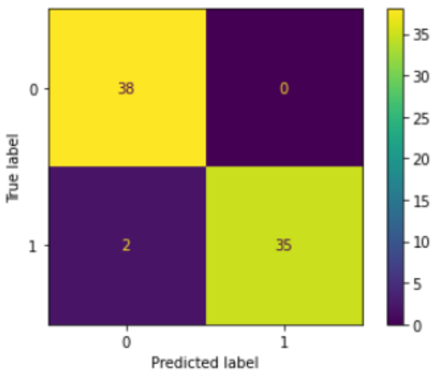


Fig. 12: Confusion matrix using voting-based ensemble.

The soft voting has no weights and uses probability from the SVC, KNN and MLP pipelines for classifying. The perceptron and adaline pipelines could not be used for soft voting since they are not able to give a probability score. The results are exactly the same as for the hard voting, making the model perform as well using hard voting as soft voting.

E. Comparison of Methods

An overview over the classification results using different methods can be seen in Table II. Here the different methods can be compared. All the methods have resulted in high accuracy, precision and recall, even though some of the methods have higher impact on the results than others. The results seem to have increased for each method, whereas the model also has become fairer and faster. The implementation of a CNN feature extraction seems to have improved the results the most, but using balanced datasets using different pipelines and dimensionality reduction (PCA), and exploring voting-based ensemble have seemingly also been important changes for improving the overall effectiveness performance.

TABLE II: Results for the classification problem according to different evaluated solutions.

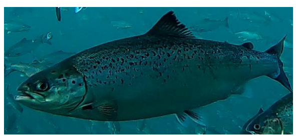
Method	Accuracy (%)	Precision (%)	Recall (%)
Original images	68.9	45.0	29.0
CNN feature extraction	94.3	90.3	90.3
Balanced dataset	94.7	94.7	94.7
MLP pipeline with PCA	96.0	97.2	94.6
Hard voting ensemble	97.3	100	94.6
Soft voting ensemble	97.3	100	94.6

F. Count Number of Loser Fish

In order to use the model to count number of loser fish in a sea cage, the model has to be able to take in a set of images, and classify each of the instances found as healthy or loser fish. By using the classification model implemented above, the counting function will be able to take in any image of a fish instance, perform a CNN feature extraction of the image and classify the fish as either healthy or loser with a high level of certainty.

By summing the amount of images classified as loser fish, the predicted number of loser fish could be counted. This method for counting loser fish is much more effective than counting them manually, which could lead to more loser fish being discovered, such that the welfare of fish could be improved.

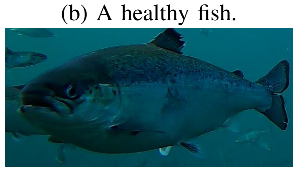
The classification algorithm classified 354 of the 1094 cropped images as loser fish. By performing a spot check of the classified images, a lot of them seems to be correctly classified, such as the ones in Figure 13, which were classified as healthy, and the ones in Figure 14, classified as loser.



(a) A healthy fish.

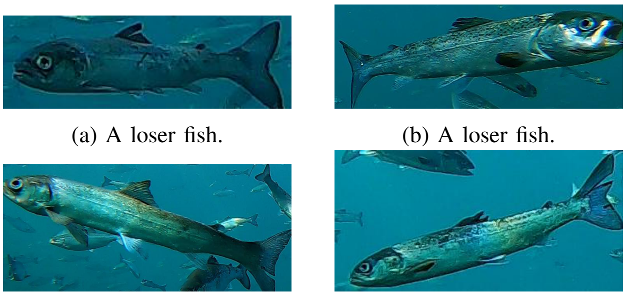
(b) A healthy fish.

(c) A healthy fish.



(d) A healthy fish.

Fig. 13: Images correctly classified as healthy fish.

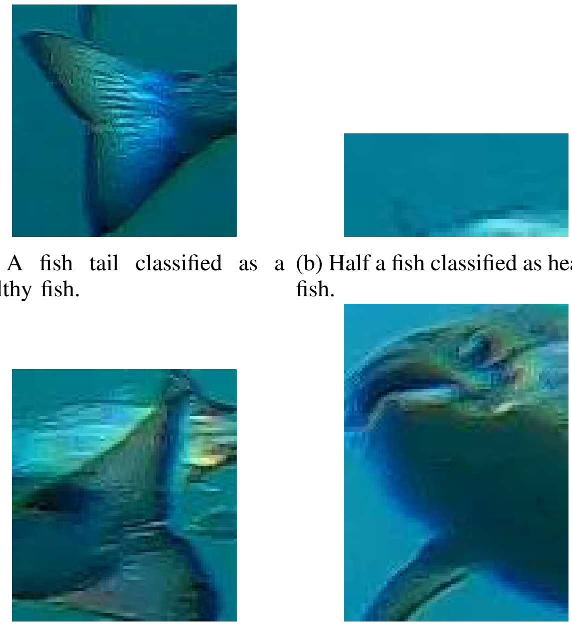


(a) A loser fish. (b) A loser fish.

(c) A loser fish. (d) A loser fish.

Fig. 14: Images correctly classified as loser fish.

There are however some images that are hard to classify, such as the ones in Figure 15, showing only the tail or head of the fish. These are all classified as healthy fish, which might be correct, but this is hard to know. These images could have been removed manually from the collection before classifying the fish or could have been removed by setting the threshold for creating images of the bounding boxes to a higher certainty percentage.



(a) A fish tail classified as a healthy fish. (b) Half a fish classified as healthy fish.

(c) A fish tail classified as a healthy fish. (d) A fish head classified as a healthy fish.

Fig. 15: Images of part of fish classified as a healthy fish.

V. Conclusion

This study has investigated the possibility to use machine learning to automatically detect and count loser fish in sea cages. By discovering the amount of loser fish, the fish welfare factors can be established and improved. By using images collected from cameras installed in sea cages, single fish can be recognised from the images. Each of the discovered fish can then be classified as either healthy or loser fish, with an accuracy of 97.3 %, based on the best classification model identified in this study. The classification model was discovered to be most effective when the features from the images are extracted using CNN, the dataset is balanced, the dimensions are reduced, and when using voting-based ensemble for combining the classification results of different classifiers. This counting of classified fish is then a lot more accurate and effective then manual counting, and could be a sensible tool for improving the welfare in sea cages.

Even though the fish detection and classification systems have led to promising results, some possible improvements can be carried out in future work. One improvement could be to use a larger dataset, for counting a larger number of fish from sea cages, and most importantly for training the fish classification model more accurately. A larger dataset can be achieved if more images are taken of fish in sea cages, use an AI image generator to create synthetic images of fish, create 3D fish models or perform dataset augmentation, such as rotating, flipping, zooming or translating operations. Another improvement that could be possible with a larger dataset is to create other fish detection models. With those large collections, detection algorithms could be trained to recognise only salmon fish, instead of all fish types, and to only create bounding boxes around large sections of fish, instead of fish tails or heads. Such detections of fish tail or head, which are not possible to be classified correctly could also have been removed from the dataset manually, but this would have been a very laborious and time consuming task. The use of semi automatic annotation tools could be explored.

One last improvement could be to install the fish detection and classification solution in a sea cage to test and evaluate the performance of the solution in a real world scenario. In order to classify the fish more correctly, fish behaviour could have been assessed as well, since loser fish swim differently from healthy fish. For detecting fish behaviour, videos from sea cages would have to be analysed and implemented in a classification model. If videos are used, the system needs to keep track of which fish are already classified, so that the same loser fish is not counted multiple times.

References

[1] Cambridge Dictionary. Cambridge University Press 2022. https://dictionary.cambridge.org/

[2] C. Noble, K. Gismervik, M.H. Iversen, J. Kolarevic, J. Nilsson, L.H. Stien, and J.F. Turnbull (2018). Welfare indicators for farmed atlantic salmon: tools for assessing fish welfare.

[3] C. Medaas, M. E. Lien, K. Gismervik, T. S. Kristiansen, T. Osmundsen, K. Vedal Størkersen, B. Tørud, and L. H. Stien (2021). Minding the gaps in fish welfare: the untapped potential of fish farm workers, Journal of Agricultural and Environmental Ethics, 34(5):29.

[4] Salmon Group (2020). Fish welfare in fish farming - what is it?.

[5] Institute of Marine Research (2019). Topic: fish welfare.

[6] K. Banno, F. M. F. Gonçalves , M. Anichini, L. C. Gansel and R. Torres (2022). Automatic Detection of Growth-Stunted Phenotype in Farmed Atlantic Salmon: A New Insight into Quantify their Distribution and Behaviour based on a Machine Learning Approach. Aquaculture Europe 2022.

[7] L.H. Stien, M.B.M. Bracke, O. Folkedal, J. Nilsson, F. Oppedal, T. Torgersen, S. Kittilsen, P.J. Midtlyng, M.A. Vindas, Ø. Øverli and T.S. Kristiansen (2012). Salmon Welfare Index Model (SWIM 1.0): a semantic model for overall welfare assessment of caged Atlantic

salmon: review of the selected welfare indicators and model presentation. Reviews in Aquaculture, 5, 33-57.
[8] O. Folkedal, J.M. Pettersen, M.B.M. Bracke, L.H. Stien, J. Nilsson, C. Martins, O. Breck, P.J. Midtlyng and T. Kristiansen (2016). On-farm evaluation of the salmon welfare index model (SWIM 1.0): theoretical and practical considerations. Animal Welfare, 25(1), 135-149. @InProceedingsBanno2022Aquaculture, author = Kana Banno and Filipe M. F. Gonçalves and Marianna Anichini and Lars C. Gansel and Ricardo da Silva Torres, booktitle = Aquaculture Europe 2022, title = Automatic Detection of Growth-Stunted Phenotype in Farmed Atlantic Salmon: A New Insight into Quantify their Distribution and Bahaviour based on a Machine Learning Approach, year = 2022,
[9] AI Aoi, fish_detection, 2018. https://github.com/kwea123/fish_detection
[10] Hongkun Yu, Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Le Hou, Pengchong Jin, Fan Yang, Frederick Liu, Jaeyoun Kim, and Jing Li, TensorFlow Model Garden, 2020. https://github.com/tensorflow/models

B Confusion Matrices of Classifiers

The confusion matrix of perceptron can be seen in Figure B.1, of Adaline in Figure B.2, of SVC in Figure B.3 and of KNN in Figure B.4.

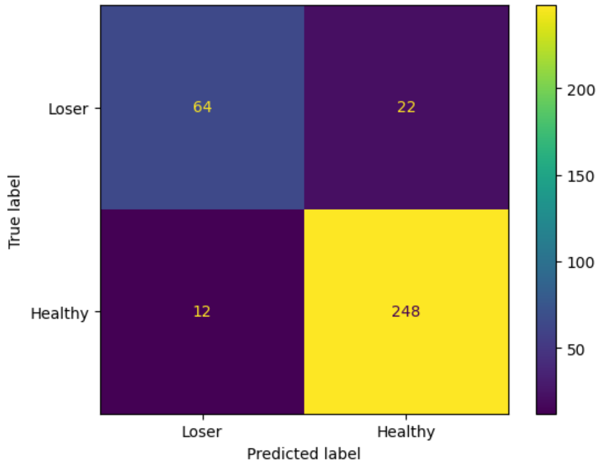


Figure B.1: Confusion matrix results for the perceptron classifier.

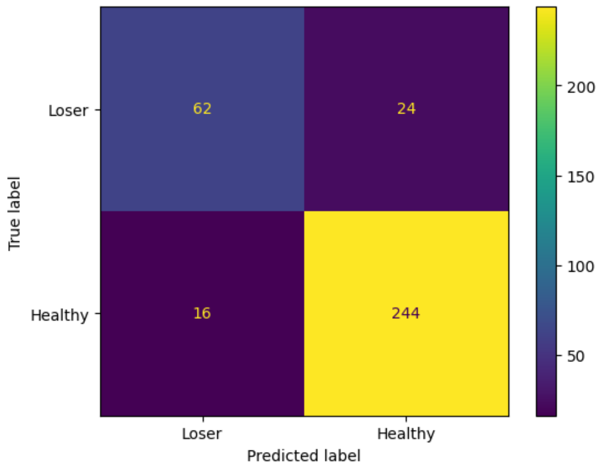


Figure B.2: Confusion matrix results for the adaline classifier.

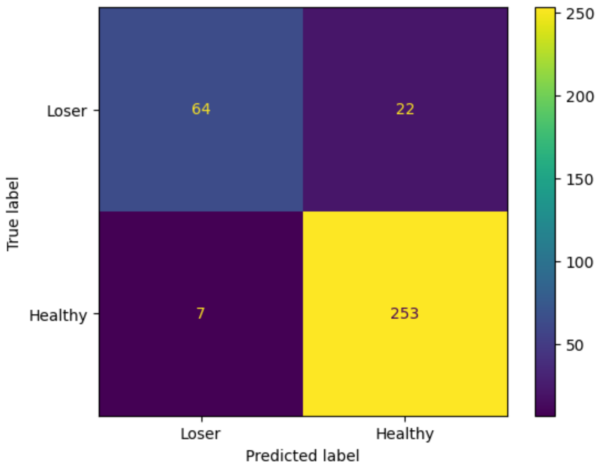


Figure B.3: Confusion matrix results for the SVC classifier.

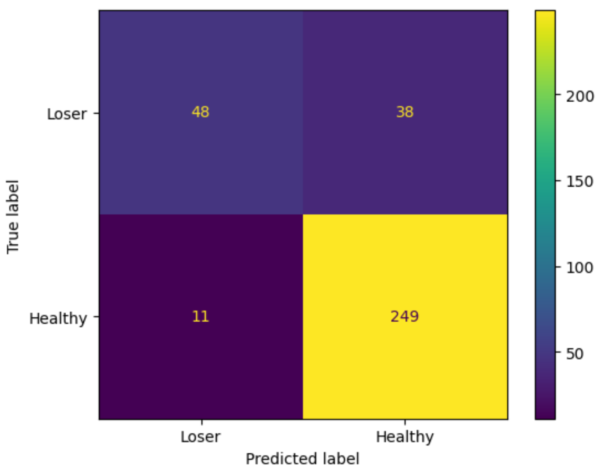


Figure B.4: Confusion matrix results for the KNN classifier.

NTNU

Norwegian University of
Science and Technology