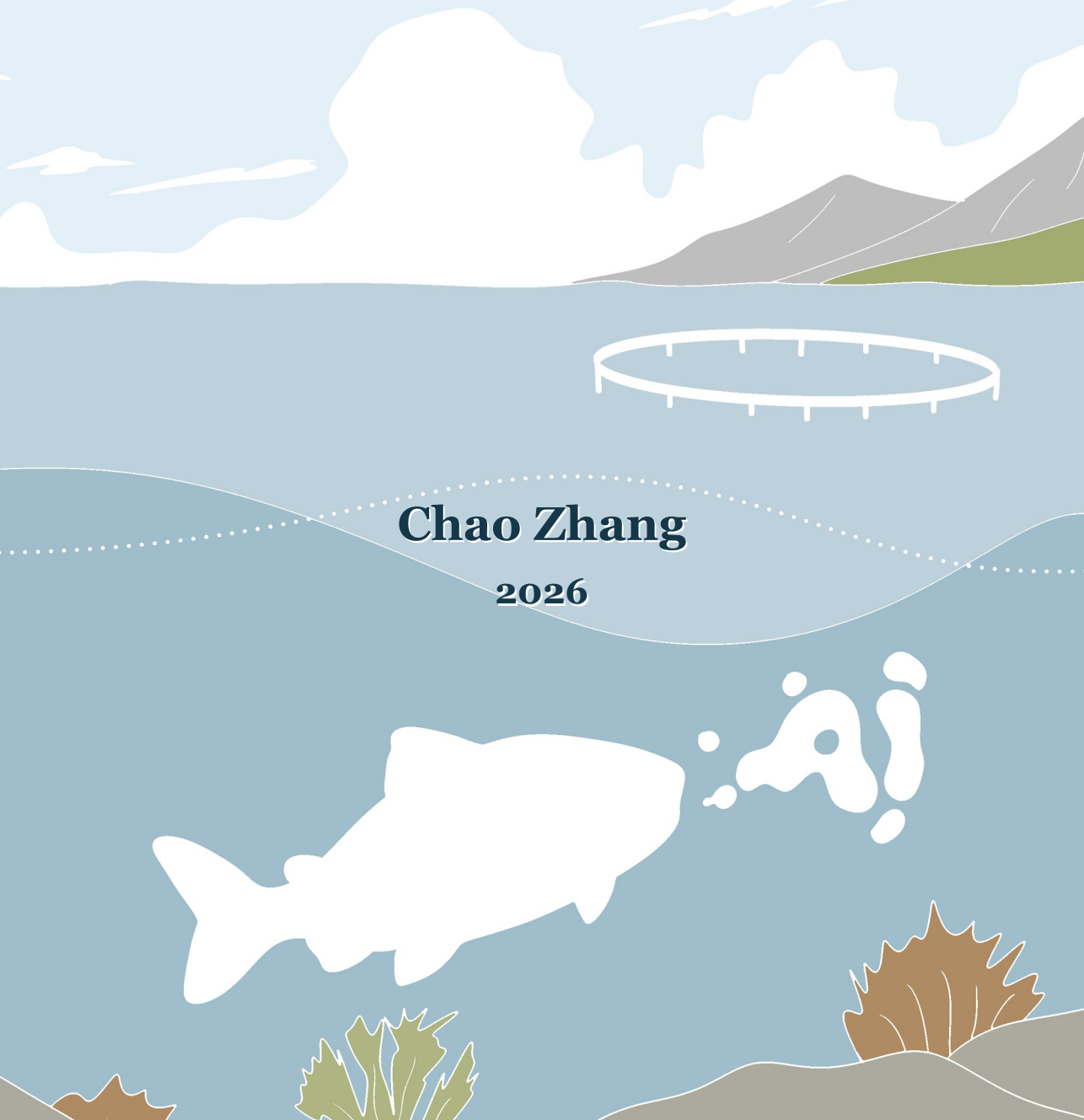
Artificial Intelligence for Salmon Farming: Perception, Welfare Modelling, and Decision Support



Chao Zhang

2026

Propositions

1. Synthetic data matter most in complex domains with scarce data.
   (this thesis)
2. Information retrieval is essential for the explainability and reliability of AI agent systems.
   (this thesis)
3. Data quality and quantity matter more than model complexity.
4. Early-stage publication motivates PhD researchers more than a single large study saved for the end.
5. Convenience is paid for by giving up privacy.
6. Salmon lice have a right to welfare too.

Propositions belonging to the thesis, entitled

Artificial Intelligence for Salmon Farming: Perception, Welfare Modelling, and Decision Support

Chao Zhang
Wageningen, 02 December 2026

Artificial Intelligence for Salmon Farming: Perception, Welfare Modelling, and Decision Support

Chao Zhang

Thesis committee

Promotor
Prof. Dr R. da Silva Torres
Professor in Artificial Intelligence
Wageningen University & Research

Co-promotors
Dr M.B.M. Bracke
Researcher in Animal Health & Welfare
Wageningen University & Research

Dr L.C. Gansel
Associate Professor in Department of Biological Sciences
Norwegian University of Science and Technology, Ålesund, Norway

Other members
Prof. Dr *ir.* B. Tekinerdogan, Wageningen University & Research
Prof. Dr *ir.* G. F. Wiegertjes, Wageningen University & Research
Dr Ø. Patursson, University of the Faroe Islands, Tórshavn, Faroe Islands
Dr P. Han, Norwegian University of Science and Technology, Ålesund, Norway

This research was conducted under the auspices of the C.T. de Wit Graduate School of Production Ecology & Resource Conservation (PE&RC)

Artificial Intelligence for Salmon Farming: Perception, Welfare Modelling, and Decision Support

Chao Zhang

Thesis
submitted in fulfilment of the requirements for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus
Prof. Dr C. Kroeze,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on Wednesday 2 December 2026
at 10:30 a.m. in the Omnia Auditorium.

Chao Zhang
Artificial Intelligence for Salmon Farming:
Perception, Welfare Modelling, and Decision Support
233 pages.

PhD thesis, Wageningen University, Wageningen, NL (2026)
With references, with summary in English

DOI: https://doi.org/10.18174/682161

Summary

Aquaculture plays an important role in global food security and human nutrition, but its continued development requires more responsible and sustainable management. This is particularly challenging in open sea-cage salmon farming, where fish are raised in large and dynamic marine environments, and information about environmental conditions, behaviour, health, welfare, and regulatory risks is often incomplete, delayed, or uncertain. Effective management therefore depends on improving how farm data, scientific knowledge, and regulatory information are acquired, interpreted, and used.

The overall aim of this thesis is to investigate how artificial intelligence (AI)-based methods can support farm-state perception, welfare assessment, and decision support in salmon farming, thereby contributing to more transparent and evidence-based aquaculture management. Atlantic salmon farming is selected as the case system for its combination of open-environment production, welfare and biosecurity risks, and regulatory requirements.

The thesis addresses three interconnected management layers, each representing a key bottleneck between data acquisition, knowledge use, and management decisions. Salmon lice serve as the representative case at the perception layer, where this thesis investigates AI-based detection of early planktonic-stage lice in seawater. At the welfare layer, it examines AI-assisted extraction and classification of welfare-related scientific statements to support semantic welfare modelling. At the decision-support layer, it explores a modular AI-agent-based framework for integrating regulatory and management-related information into evidence-grounded decision support. Specifically, the thesis is structured as follows:

Chapter 2 explores the detection of planktonic-stage salmon lice larvae in seawater. In this chapter, an online image acquisition system is introduced to continuously collect images from seawater samples. Using this system, a dataset of 8316 images was constructed for salmon lice detection. Sixteen object detection models, includ-

ing one-stage and two-stage CNN-based models and Transformer-based models, were evaluated for salmon lice detection. RT-DETR achieved the best overall performance with a precision of 97.2% and a recall of 97.8%, while most YOLO-family models maintained high precision and recall with fast inference speeds. To our knowledge, this is the first study to apply machine learning to the rapid detection of different developmental stages of salmon louse larvae in seawater samples. The results demonstrate the feasibility of machine learning-based planktonic lice detection and establish a baseline for subsequent studies on rapid, easy-to-use, and cost-effective salmon lice monitoring in seawater.

Chapter 3 extends salmon lice detection from relatively clean and controlled seawater samples to more complex real-world seawater backgrounds. In such environments, both data acquisition and detection become more difficult because salmon louse larvae are small and sparse, while seawater samples contain diverse non-target organisms, particles, and impurities. To address the lack of annotated data under these conditions, this chapter presents an automated synthetic data generation method based on the Segment Anything Model (SAM), designed to improve the detection of multi-stage salmon louse larvae in complex seawater environments. By extracting annotated lice instances and combining them with real target-free background images, the method makes effective use of large numbers of unlabeled background images and generates diverse, realistic synthetic training data. Orthogonal experiments show that the diversity of non-target regions in synthetic images was important for data quality and model performance. Across different detection models, the use of synthetic data consistently improved detection performance. In real-world seawater tests, synthetic data increased model recall by 11.0% to 87.0%, further demonstrating its value for improving detection under complex seawater conditions.

Chapter 4 further explores online synthetic data generation for salmon lice detection and proposes Salmon Lice Detection based on Online Synthetic Training (SDOST). Unlike offline synthesis, where synthetic images are fixed before training, SDOST generates new image compositions during training, allowing the model input to remain dynamic and reducing overfitting. A feature-proximal replay mechanism was used to reuse synthetic samples with features close to the real-data distribution, while a fast gradient sign method (FGSM)-based adversarial perturbation was introduced to improve robustness. Compared with models trained only on fixed offline synthetic data or the original limited real data, SDOST improved the F1-score by 4.3% to 17.7% and 13.8% to 166.4%, respectively. These improvements were especially evident under few-shot conditions, demonstrating that online synthetic training can support the detection of small, sparse, and visually challenging salmon lice larvae and may also be useful for other aquaculture perception tasks where annotated target data are

scarce.

Chapter 5 explores AI-assisted knowledge extraction for modelling of animal welfare. Semantic models for welfare assessment depend on scientific statements that describe environment-based welfare attributes, animal-based welfare measures grouped into weighting categories, and their interrelationships. However, these statements are scattered across large amounts of literature and are normally identified manually by domain experts. In this chapter, a natural language processing (NLP)-based framework is developed to automatically screen scientific documents, split them into sentences, and identify welfare-related statements. Different classification strategies were evaluated, including large language model (LLM)-based few-shot learning, two-stage classification, and end-to-end fine-tuning of pretrained Transformer models. The results show that end-to-end fine-tuning achieved the strongest overall performance and could identify welfare-related statements with a precision of 96.0% and a recall of 92.4% on the main test set. An interactive system was also developed so that users can upload documents, inspect classified sentences, and use the extracted statements as traceable evidence for the construction and updating of the welfare model. This chapter demonstrated that AI-based text analysis can reduce manual literature screening and support the systematic use of scientific knowledge in animal welfare assessment.

Chapter 6 explores AI-agent-based decision support for Atlantic salmon farming and introduces the Salmon Farm Multi-Agent Decision Support System (S-MADSS), an evidence-grounded, multi-agent framework for supporting farm-management decisions. The framework combines a traceable evidence-construction pipeline with a structured multi-agent workflow for evidence-grounded question answering and compliance-oriented decision support. Retrieved evidence is provided to specialized agents that generate answers, recommend actions, and candidate decisions from regulatory, welfare, risk, and feasibility perspectives. The results show that hybrid retrieval with reranking improves evidence quality, while local context expansion improves knowledge recall by incorporating adjacent text around retrieved knowledge fragments. In risk identification, the evidence-grounded auditing workflow achieves a maximum recall of 95.0%, compared with 63.7% for retrieval-free LLM-based identification. Scenario-based decision tasks and a real-farm case study on salmon lice treatment further demonstrate that S-MADSS can coordinate treatment feasibility, welfare concerns, regulatory compliance, and farm-specific information within a unified workflow. This chapter demonstrated that modular AI-agent systems can provide a feasible foundation for structured, traceable, and evidence-based decision support in salmon farming. The modular design also makes S-MADSS flexible and extensible. In future work, perception and analysis modules, such as broader salmon lice detection,

welfare assessment, environmental monitoring, and farm-operation analysis, could be integrated into the framework so that decision support is grounded in a wider range of farm-specific evidence.

Chapter 7 synthesizes the findings from the preceding chapters and discusses how the three AI-based components contribute to salmon farming management. It reflects on the addressed knowledge gaps, the broader implications for AI-supported salmon farming and sustainable aquaculture, and the main limitations of the current studies. It also outlines future research directions, including broader commercial validation and integration of perception, welfare, environmental, and decision-support modules, and discusses wider societal implications related to animal welfare, regulatory transparency, environmental responsibility, and human judgement in AI-supported farm management.

The thesis contributes methodological foundations for more evidence-based, welfare-aware, and transparent salmon farming management, thereby supporting a broader transition towards more responsible and sustainable aquaculture. Rather than replacing farmer judgement, these methods are intended to support human decision-making under uncertainty by making relevant data, knowledge, assumptions, risks, and regulatory considerations more explicit. Although Atlantic salmon farming is used as the case system, the workflows developed in this thesis may also be relevant to other aquaculture and animal production systems facing similar challenges in perception, welfare assessment, and decision-making.

Contents

Page

Propositions ii

Summary vii

Contents xi

Chapter 1 General introduction 1

Chapter 2 Rapid detection of salmon louse larvae in seawater based on machine learning 17

Chapter 3 An image synthesis framework for enhanced salmon louse larvae detection in complex seawater conditions 45

Chapter 4 Online synthetic training for annotation-scarce salmon lice detection 79

Chapter 5 Automated extraction of scientific statements for integrated assessment of animal welfare using language models 91

Chapter 6 S-MADSS: An agentic framework for evidence-Grounded, compliance-oriented decision support in Atlantic salmon farming 125

Chapter 7 General discussion 161

Appendices 179

Appendix A 180

Appendix B 198

Appendix C 201

References 208

Acknowledgements 225

About the author 229

Authorship statement 232

Chapter 1

General introduction

1.1 Aquaculture, salmon farming, and the need for responsible and sustainable management

Aquaculture has a long history and has gradually evolved from traditional, small-scale practices into a modern and increasingly technology-intensive food production sector. Early aquaculture can be traced back to China between 2000 and 1000 BCE, where carp (*Cyprinus carpio*) were reared under controlled conditions (Rabanal & Fishery and Aquaculture Economics and Policy Division, 1998; Yue & Shen, 2022). However, this rapid expansion is commonly attributed to the second half of the twentieth century, when production shifted toward more intensive, standardized, and technologically advanced farming systems (Rabanal & Fishery and Aquaculture Economics and Policy Division, 1998; Rossi et al., 2024).

Today, aquaculture is one of the fastest-growing food production sectors globally (Garlock et al., 2022; Naylor et al., 2021). In 2022, aquaculture yielded 94 million tonnes of aquatic animals, exceeding capture fisheries for the first time, accounting for 51% of global aquatic animal production (FAO, 2024). Aquatic animal foods also play an important role in human nutrition, providing about 15% of global animal protein intake in 2021 and supplying essential micronutrients and bioactive compounds, including zinc, calcium, and long-chain n-3 polyunsaturated fatty acids (FAO, 2024; Garlock et al., 2022). Consequently, aquaculture has become important for both global food security and human nutrition.

The role of aquaculture is expected to grow further as the global population approaches 9.7 billion by 2050 (Troell et al., 2023). In response to this demand, the Food and Agriculture Organization (FAO) of the United Nations has proposed the Blue Transformation vision (FAO, 2022; Troell et al., 2023). This vision emphasizes that future growth must balance increased production with environmental, social, and economic sustainability (FAO, 2025). Achieving this transformation requires farm management approaches that reduce resource consumption, mitigate environmental impacts, safeguard animal welfare, strengthen long-term governance capacity, and make more effective use of available data resources and scientific knowledge (FAO, 2026; Naylor et al., 2021; Garlock et al., 2024). However, translating these high-level sustainability principles into daily farm operations remains constrained by an information bottleneck. Marine cage farming systems are dynamic, spatially variable, and visually obscured, leaving information about fish behavior, health, welfare, and environmental changes often incomplete, delayed, or uncertain (Føre et al., 2018; Burke et al., 2025; Georgopoulou et al., 2021).

Atlantic salmon (*Salmo salar*) farming is a representative example because it combines intensive production, open marine environments, strict regulation, and major welfare and biosecurity challenges (Misund et al., 2024; FAO, 2024). Key risks, including salmon lice (*Lepeophtheirus salmonis*) infestations and handling-induced acute stress, often occur in underwater environments that are difficult to observe directly (Hansen & Windsor, 2006; Oppedal et al., 2011). Managing these systems requires balancing regulatory requirements with biological and operational constraints under high uncertainty. This makes salmon farming a representative case for evaluating how advanced information processing can support more responsible and sustainable aquaculture management.

Against this background, this thesis addresses three interconnected information-processing layers in salmon farming, situated within the broader agenda of precision fish farming (PFF): farm-state perception, welfare assessment, and decision support. As illustrated in Figure 1.1, this thesis investigates three representative methodological pillars, each addressing a key information bottleneck within its respective layer:

1. **Perception layer:** the automated detection of early planktonic-stage salmon lice in seawater, addressing the bottleneck of incomplete and delayed farm-state information,

2. **Welfare assessment layer:** the AI-supported extraction and classification of welfare-related scientific statements from literature, addressing the bottleneck of fragmented welfare evidence,

3. **Decision-support layer:** an architectural exploration of modular, multi-agent-based decision support under complex legal and regulatory constraints, addressing the bottleneck of translating heterogeneous information into actionable, regulation-compliant decisions.

Together, these three components follow a simple logic: an aquaculture system must first **see** what is happening on the farm (**perception**), then **know** what these observations mean for animal welfare (**welfare assessment**), and finally **act** on this understanding through informed decisions (**decision support**). Although salmon farming is used as the case system, this structure is not species-specific and may be transferable to other aquaculture or animal production systems where management depends on uncertain observations, evolving scientific knowledge, and complex operational constraints.

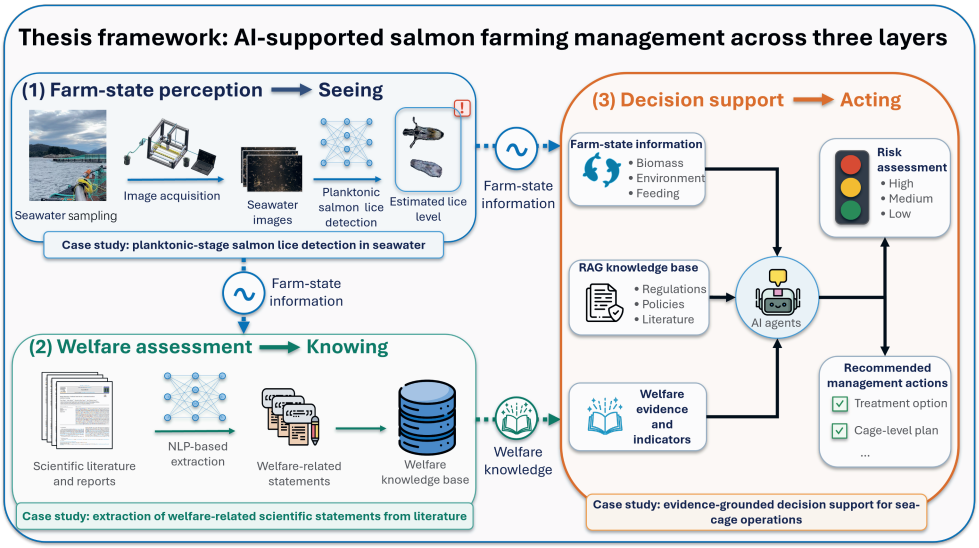


Figure 1.1: Conceptual framework for AI-supported salmon farming management across three information-processing layers. The thesis is structured around farm-state perception, welfare assessment, and decision support. Each layer is represented by one case study: planktonic-stage salmon lice detection in seawater, extraction of welfare-related scientific statements from literature, and modular AI-agent-based decision support for integrating legal and regulatory information. The arrows indicate the conceptual flow envisioned across the three layers, whereby perception outputs and welfare-related evidence could ultimately inform downstream decision support alongside regulatory and farm-specific information. This progression—from seeing, to knowing, to acting—forms the conceptual backbone of this thesis.

1.2 Key challenges in salmon farming

This section examines the information bottleneck at each of the three management layers: perceiving farm states, assessing animal welfare, and coordinating management decisions.

1.2.1 Challenges in perceiving farm states

Perception in aquaculture refers to the process of sensing and identifying information about a farm's environmental and biological state, such as water temperature, parasite density, skin damage, and swimming behaviour (Akram et al., 2026). Timely and reliable perception of critical farm states is essential for responsible salmon farming management (Biazi & Marques, 2023), yet achieving this remains difficult. This perception bottleneck is particularly evident in open marine salmon farming, where

biological processes and environmental thresholds vary across spatial and temporal scales (Føre et al., 2018; Oppedal et al., 2011). Localized risks or early-stage anomalies often emerge rapidly and remain hidden from conventional diagnostic protocols. Although modern commercial farms increasingly use environmental sensors, hydroacoustic sonar, and underwater cameras, these systems still have limited capacity to provide representative, continuous, and biologically grounded data at the farm or cage scale (Zhou et al., 2025; Biazi & Marques, 2023). Vision-based monitoring in particular is constrained by turbidity, changing illumination, suspended particles, and variable fish density (Lu et al., 2017; Yang et al., 2021), and traditional rule-based machine vision systems built on such data often perform poorly under non-standardized commercial conditions (Føre et al., 2018; Konovalov et al., 2019). Manual inspection, meanwhile, is difficult to scale across large cage volumes because it is labour-intensive and time-consuming.

The management of salmon louse (*Lepeophtheirus salmonis*) serves as a clear example of this perception gap. As one of the most persistent threats in salmon aquaculture (Klimesova et al., 2025; Borchel & Nilsen, 2026), salmon lice infestations can cause epithelial damage, chronic physiological stress, and systemic welfare degradation (Igboeli et al., 2014). In open sea cages, infested fish can also act as reproductive reservoirs, dispersing larvae into the surrounding water and migration corridors, increasing the risk of infestation in wild salmonids (Penston & Davies, 2009). Therefore, salmon lice represent an important regulatory and environmental concern, and major producing countries, such as Norway, enforce mandatory monitoring thresholds (Overton et al., 2019).

Despite its urgency, proactively monitoring salmon lice is a major technical challenge (Bui et al., 2021). As shown in Figure 1.2, the salmon louse has eight life stages, including planktonic stages in seawater and parasitic stages attached to fish. During their early planktonic stages (nauplius and copepodite), larvae are small, approximately 0.5–0.7 mm, sparsely distributed, and free-floating in complex seawater samples containing morphologically similar plankton and organic debris (Krolicka et al., 2022) (see Figure 1.3 for an example). Current empirical detection approaches are often indirect, costly, and time-consuming, relying either on manual morphological validation using light microscopy or on molecular detection through quantitative polymerase chain reaction (qPCR) assays of water samples (Bui et al., 2021; Thompson et al., 2022). This inherent data latency eliminates a window for early warning. Current lice monitoring in commercial practice therefore remains concentrated on the parasitic stage, typically relying on sampling and manual visual inspection of a limited number of individuals (Thorvaldsen et al., 2019). Although this parasitic-stage counting approach is widely used in practice and regulation, lice counts may be af-

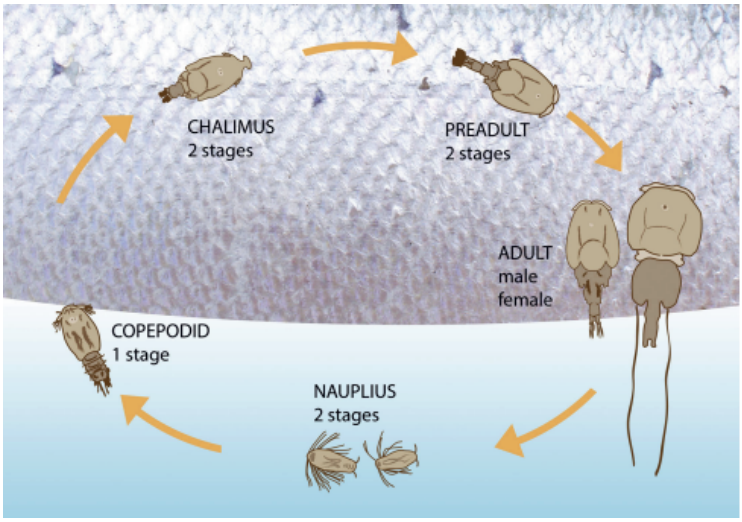


Figure 1.2: The life cycle of salmon louse. Nauplius larvae are approximately 0.5–0.6 mm long, copepodids about 0.7 mm, chalimi 1.1–2.3 mm, preadults 3.4–5.2 mm, and adults about 5–6 mm in males and 8–12 mm in females. Reproduced from (Thorstad et al., 2015) under CC BY. No changes made. Graphic design: Kari Sivertsen, NINA.

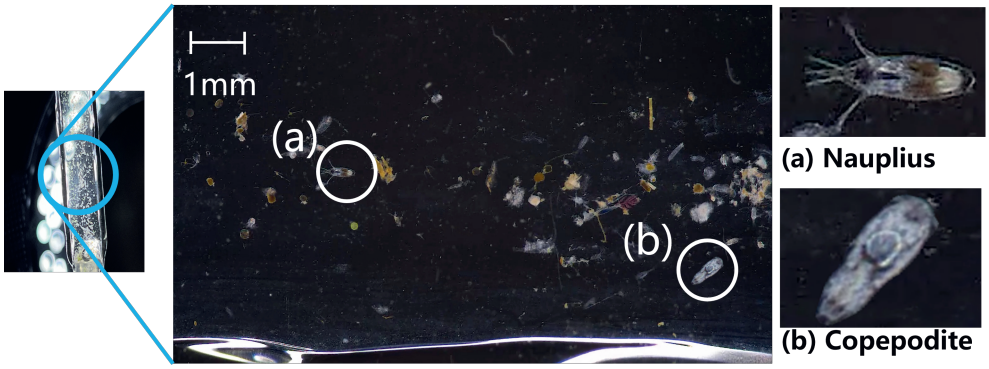


Figure 1.3: Example of salmon louse nauplius and copepodite larvae in a concentrated seawater sample collected from a salmon farm.

fected by sampling design, counting procedures, and limited representativeness at the cage or farm scale (Krolicka et al., 2022).

The case of salmon lice therefore illustrates a broader challenge in salmon farming: some of the most management-relevant states are also among the most difficult to perceive with acceptable accuracy, frequency, and spatial coverage.

1.2.2 Challenges in welfare assessment and system understanding

Animal welfare has been characterized in different ways by different scholars. Fraser et al. (1997) frame it around overlapping concerns about whether animals lead natural lives, feel well, and function well, while Bracke et al. (1999a) describe it more broadly as the quality of life as experienced by the animal itself. These definitions presuppose that animals are capable of subjective experience, and growing evidence indeed suggests that fish, including salmon, are sentient and can experience negative states such as fear and pain (Santurtun et al., 2018). In intensive salmon aquaculture, safeguarding welfare is therefore a multidimensional imperative rooted in ethical responsibility, shifting legal landscapes, and economic performance (Pavlidis et al., 2023). Therefore, reliable welfare assessment is essential for modern veterinary health management and for maintaining the social acceptance of the aquaculture industry (Huntingford et al., 2006). Figure 1.4 showed an example of salmon with poor welfare condition.



Figure 1.4: Example of a visible welfare problem in farmed salmon. The image shows an underdeveloped salmon, illustrating that welfare challenges can appear as concrete physical conditions in individual fish. Adapted from Stien et al. (2013).

Perceiving individual farm states, such as detecting a parasite infestation or an abnormal behaviour, is a first step toward understanding animal welfare. Welfare is not a static physiological metric determined by a single factor, but a multidimensional state shaped by the cumulative interplay of environmental conditions, husbandry practices, and biological responses (Noble et al., 2018). Salmon lice management illustrates why a single perceived state is insufficient on its own: a high lice count calls for treatment, yet delousing itself involves crowding, handling, and transfer that can induce stress and increase reinfection risk, particularly when fish are already in poor health (Føre et al., 2018; Delfosse et al., 2021). Assessing welfare therefore requires

weighing such competing risks rather than detecting factors in isolation. Translating perceived states into a meaningful welfare assessment requires the integration and contextual interpretation of multiple heterogeneous sources of evidence (Stien et al., 2013; Pettersen et al., 2014).

To formalize this reasoning and reduce uncertainty, semantic modelling has been used to develop interpretable assessment frameworks, including SWIM 1.0 (Stien et al., 2013) and SWIM 2.0 (Pettersen et al., 2014). As summarized in the workflow in Figure 1.5, semantic modelling utilizes scientific literature, diagnostic reports, and expert consensus to systematically map qualitative welfare statements into formalized model components, including welfare needs, attributes, weighting categories, and their relationships (Bracke et al., 1999a,b).

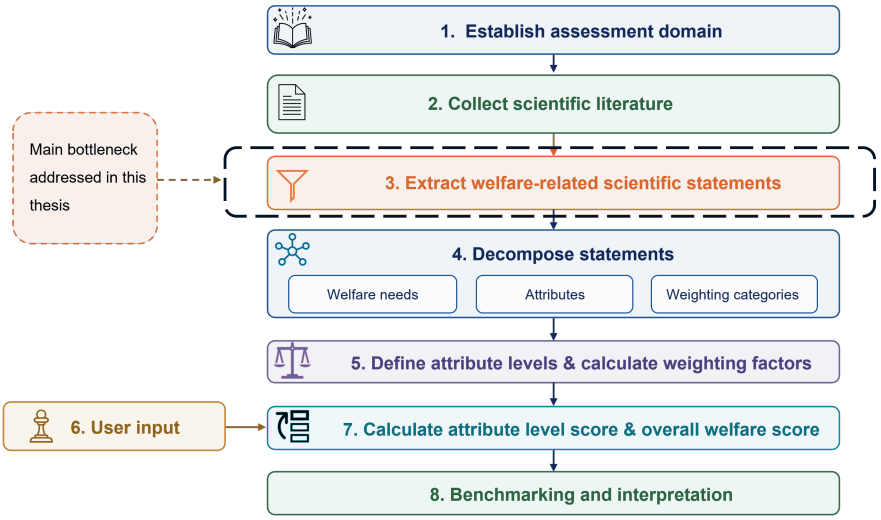


Figure 1.5: Semantic modelling workflow for animal welfare assessment. Scientific literature and expert knowledge are used to identify welfare-related statements, which are then structured into welfare needs, attributes, weighting categories, and relationships. These components support the definition of attribute levels and calculation of weighting factors, allowing user inputs to be translated into attribute level scores, overall welfare scores, and benchmarking results.

Despite the utility of these semantic frameworks, static welfare models are only as good as the knowledge that constructs them. Animal welfare science is an evolving discipline, and new findings, in the form of welfare-related scientific statements in the literature, may change how risk factors and management practices relate to welfare outcomes (Marchant-Forde, 2015). Therefore, welfare models need not only to represent current knowledge, but also to be systematically updated. At present, however, this extraction and classification step still depends largely on manual screen-

ing by domain experts, as shown in Figure 1.5. This manual pipeline is difficult to scale with the growth of scientific literature, creating a methodological gap in how evolving welfare evidence can be efficiently incorporated into interpretable welfare models.

1.2.3 Challenges in decision-making and management coordination

Management decisions in aquaculture span multiple production stages, from hatchery and land-based rearing to sea cage farming and harvesting (Burke et al., 2025; Nilsen et al., 2020). At each stage, decisions must be coordinated across multiple factors, such as growth status, environmental conditions, disease and parasite risks, welfare requirements, economic costs, and policy and regulatory factors (Garlock et al., 2024; ASA, 2025). The challenges in perceiving farm states and assessing animal welfare, discussed above, further compound this complexity (Føre et al., 2018; Mathisen et al., 2016). In practice, many decisions still rely on farmers' experience and direct observation.

Existing optimization and decision-support methods address well-defined problems with clear objectives, inputs, and constraints, such as predicting biomass growth (Zhang et al., 2024c), optimizing feed distribution (Thornburg, 2025), or planning harvest schedules (Domínguez-May et al., 2024). However, practical farm management often requires welfare, regulation, operational feasibility, and multi-source monitoring information to be considered together (Føre et al., 2018), which existing methods are not designed to support.

Salmon lice management illustrates this gap directly: intervention decisions cannot rely on lice levels alone but must also weigh fish condition, welfare risk, treatment feasibility, and regulatory pressure (Føre et al., 2018; Boerlage et al., 2024). These factors may point to conflicting management actions, for instance when regulatory pressure to treat does not align with the stock's biological status.

The main challenge, therefore, is not simply to optimize individual tasks, but to develop a flexible, modular decision-support framework that can integrate diverse information sources and interpret management decisions within the specific biological, operational, and regulatory context of each farm.

1.3 Artificial Intelligence in aquaculture

This section gives a brief overview of artificial intelligence and then reviews how it has been applied to perception, welfare assessment, and decision support in aquaculture.

1.3.1 Artificial Intelligence (AI)

The challenges discussed above suggest that responsible and sustainable salmon farming requires improved capacity in perceiving critical farm states, assessing animal welfare, and supporting management decisions under multiple constraints. Artificial intelligence (AI) provides a broad methodological background for addressing these challenges. In general, AI refers to computational methods that enable machines to perform tasks commonly associated with human intelligence, such as recognizing patterns, making predictions, extracting information, and supporting decisions (Russell & Norvig, 2021; Sarker, 2022). Within this broad field, machine learning refers to methods that improve their performance by learning from data rather than following manually specified rules, and deep learning is a subset of machine learning that uses multi-layer neural networks to automatically learn relevant features from raw data, such as images or text, rather than relying on hand-crafted features (LeCun et al., 2015).

This thesis draws on statistical and deep learning models that learn task-relevant patterns directly from labelled or unlabelled examples, and can process diverse data types, including image, video (Voulodimos et al., 2018), acoustic, time-series (Hamilton, 2020), tabular (Jiang et al., 2026), and textual data (Kowsari et al., 2019).

It also draws on large language models (LLMs) (Yao et al., 2024), a more recent and specialized application of deep learning trained on very large collections of text to model and generate natural language. LLMs are capable of natural language understanding, reasoning, and generation across diverse tasks, including summarization, question answering, and information extraction (Qin et al., 2026). These models are increasingly deployed together with mechanisms that ground their outputs in external, verifiable information, such as retrieval-augmented generation (RAG) (Arslan et al., 2024), and AI agent systems (Feng et al., 2026), which extend this further by coordinating tool use, retrieval, and multi-step reasoning across heterogeneous information sources. These methods are particularly relevant when decisions depend not only on observed data, but also on scientific evidence, expert knowledge, or regulatory constraints expressed in natural language.

1.3.2 AI-based perception

In aquaculture, AI-based perception focuses on converting raw farm data into biologically or operationally meaningful information. Depending on the task, these data may include underwater images, video streams, sonar signals, and environmental sensor records (Føre et al., 2018; Gupta et al., 2022). Compared with rule-based methods that

rely on predefined features or fixed assumptions, data-driven AI methods learns task-relevant patterns directly from examples and applies these learned representations to new farm data (Saleh et al., 2022). This makes AI-based perception particularly useful for state-recognition tasks under variable farming conditions.

In visual perception, deep learning is often used for tasks such as image classification, which assigns an image to a category (Alaba et al., 2022); object detection, which locates and classifies target objects in an image (Banno et al., 2024); segmentation, which separates target regions from the background (Garcia et al., 2020); tracking, which follows objects across frames; and behaviour recognition, which identifies activity patterns over time (Cui et al., 2025). In fish farming specifically, these tasks have been used for disease detection (kumar M et al., 2025; Luo et al., 2025b), fish aggregation and trajectory analysis (Xiao et al., 2026; Georgopoulou et al., 2021), fish size estimation (He et al., 2026), abnormal activity detection (Yang et al., 2026; Liu et al., 2025a), and external parasite identification (Gupta et al., 2022).

AI-based perception is not limited to images. Sonar and hydroacoustic data can be used to analyse fish distribution and movement trajectories in sea cages (Kristmundsson et al., 2023), while water-quality and environmental sensor data can support the detection or prediction of abnormal farming conditions (Pratiwy & Aisyah, 2026; Hridoy et al., 2025). However, despite this broad applicability, automated recognition of planktonic-stage organisms in complex natural samples remains comparatively underexplored.

1.3.3 AI-based welfare assessment and modelling

AI can support animal welfare assessment at two related levels. The first concerns the detection or measurement of welfare-related indicators, such as injuries, deformities, poor body condition, abnormal behaviour, disease symptoms, or environmental stressors (Fitzgerald et al., 2025). Some of these indicators overlap with farm-state perception as they are detected from images, videos, sensors, or other monitoring data. However, they represent only partial evidence of the welfare status rather than a complete assessment (Føre et al., 2018). Recent studies have increasingly applied AI to such indicator-level welfare monitoring, such as cleanliness scoring in cattle (Foris et al., 2025), behaviour tracking in broilers (Kashiha et al., 2013; Peña Fernández et al., 2018), plumage condition in laying hens (Lamping et al., 2022), and pose estimation, locomotion scoring, and body condition scoring in dairy cattle (Michielon et al., 2024). In aquaculture, similar perception methods have been applied for fish injury detection, disease diagnosis, and abnormal behaviour recognition (Yang et al.,

2026; Liu et al., 2025a).

The second level concerns the integration of these indicators into broader welfare assessment models. Overall welfare cannot be inferred from a single indicator alone (Bracke et al., 1999a), but requires the interpretation of animal-based indicators, environmental conditions, management practices, and relationships within a specific context. Therefore, AI-based welfare assessment should also support the organization and interpretation of welfare-related knowledge. As discussed in Section 1.2.2, semantic modelling provides a suitable framework for this broader task, within which local indicators, such as external lesions or poor growth, are interpreted as evidence contributing to a wider assessment. Constructing and updating such models, however, still depends largely on manual extraction of welfare-related knowledge from the literature.

AI-based methods for this second level remain less developed than for indicator-level monitoring. Most existing AI applications focus on detecting or measuring observable indicators, while few studies address how AI can support the acquisition of the underlying scientific evidence needed to construct and update such models. Natural language processing (NLP) (Qin et al., 2026) methods can be used to identify welfare-related statements, extract welfare needs and weighing categories, and their relationships, helping move welfare assessment from indicator detection toward overall welfare modelling. However, this capacity for automated knowledge extraction remains largely unexplored for welfare model construction and updating.

1.3.4 AI-based decision support system

Decision support systems (DSSs) are computer-based tools that use data, models, rules, simulations, or optimization methods to support decision analysis (Mathisen et al., 2016). In aquaculture, DSSs have been developed for tasks such as feeding control, disease warning, environmental management, and harvest planning (Budaev et al., 2025; Azevedo et al., 2025). These systems are useful when the decision problem has relatively clear objectives, inputs, and constraints. However, many practical management decisions require the interpretation of heterogeneous information, such as sensor data, biological indicators, welfare knowledge, operational constraints, expert judgement, and regulatory requirements (Zhou et al., 2025).

AI-based methods can learn complex relationships from historical data and multi-source monitoring information, making them suitable for nonlinear and uncertain farming environments, but integrated AI-driven DSSs remain relatively limited in aquaculture. One example is the use of an extended Siamese neural network to

estimate the likelihood that a specific farming operation will succeed at a given site (Mathisen et al., 2021). More commonly, AI is applied to individual tasks, such as feeding analysis, functioning as local intelligent modules rather than integrated DSSs for multi-objective farm management.

Within this landscape, LLMs (Yao et al., 2024), RAG (Arslan et al., 2024), and AI agents (Feng et al., 2026) offer new possibilities for integrated decision support, since many decision constraints are expressed in natural language rather than as numerical variables or fixed model inputs. RAG can ground recommendations in such sources, while AI agents organize the decision process into specialized steps, such as checking farm-state data, consulting welfare knowledge, evaluating intervention options, and summarizing evidence. Multi-agent systems have already shown progress in other agricultural domains. AgroAskAI, for instance, is a multi-agent reasoning system for climate adaptation decision support in agriculture, integrating real-time weather, historical climate, geospatial data, and external knowledge for vulnerable rural communities (Cantonjos & Biswas, 2025). Similarly, ADMA Copilot coordinates control, input-formatting, and output-formatting agents to automatically interpret user intent, plan data-processing workflows, and complete agricultural data analysis tasks (Pan et al., 2024). This structure suits aquaculture's functional boundaries and context-dependent constraints, yet integrating such AI agent-based architectures with legal and regulatory information in salmon farming remains largely unexplored.

1.4 Knowledge gaps and research questions

Maybe ten years from now, salmon farming management could shift from a largely reactive practice to a proactive one. Rather than responding to problems after they surface, farms could anticipate them. Continuous monitoring across cameras, sensors, and behavioural data could replace much of today's manual inspection. This would turn isolated snapshots into a near real-time picture of animal and environmental states at the cage level. Rising salmon lice pressure or early skin damage could be detected within hours rather than weeks. Welfare assessment could undergo a parallel shift. Today's frameworks depend on periodic manual review by experts. Instead, they could evolve continuously alongside the scientific literature itself, becoming more scientifically grounded and more widely accessible. These gains would not stand alone. Perception, welfare, and management information would be drawn together within a single decision-support platform, weighed against regulations, protocols, and historical farm records. Emerging risks would be flagged to farmers, policymakers, and inspection services before they escalate, with every decision remaining traceable and transparent. However, this trajectory is not yet the reality of aquaculture.

Realizing this trajectory requires progress across all three layers, yet each currently faces a distinct bottleneck. This thesis focuses on one representative bottleneck per layer, selected because each captures a distinct but connected limitation in perception, knowledge extraction, and information integration, respectively.

From the perspective of **farm-state perception**, Section 1.3.2 showed that AI-based visual perception has been applied across a range of aquaculture monitoring tasks. Yet research on salmon lice, despite their major impact on welfare, economics, and ecology, remains comparatively limited. Existing methods mainly targeted parasitic-stage salmon lice attached to fish (Gupta et al., 2022). Automated detection of early, planktonic-stage salmon lice in seawater, which is characterized by low target density, complex visual backgrounds, and scarce labelled data, remains largely unaddressed. This thesis therefore selects early planktonic salmon lice detection as the representative case for investigating AI-based perception under these conditions.

From the perspective of **animal welfare assessment**, Section 1.3.3 showed that existing AI applications for welfare assessment concentrate on detecting or measuring individual indicators. Semantic modelling methods can compute an overall welfare score, but constructing such models requires extracting and classifying welfare-related scientific statements from the literature —a step that still relies heavily on manual screening by domain experts. This thesis therefore selects welfare-related scientific statement extraction as the representative case for investigating how AI can support welfare model construction and updating.

From the perspective of **decision support**, Section 1.3.4 showed that AI remains applied to individual, well-defined tasks, functioning as local modules rather than the integrated, traceable system envisioned above. This thesis therefore selects the integration of legal, regulatory, and management-relevant information as the representative case for decision support. This integration allows such local modules to function as a coherent whole and to make the reasoning behind decisions more structured, traceable, and context-aware.

Based on these knowledge gaps, this thesis addresses the following research questions:

RQ1: How can AI-based methods support the automated detection of planktonic-stage salmon louse larvae in seawater under conditions of low target density, complex visual backgrounds, and limited availability of labelled data?

RQ2: How can AI-based methods support the automatic extraction of welfare-related scientific statements for animal welfare modelling?

RQ3: How can legal, regulatory, and other management-relevant information be in-

tegrated into a flexible and modular AI-based decision-support approach for salmon farming?

1.5 Outline of this thesis

Following the seeing–knowing–acting logic introduced in Section 1.1, Chapters 2–4 focus on seeing (farm-state perception, RQ1), Chapter 5 focuses on knowing (welfare assessment, RQ2), and Chapter 6 focuses on acting (decision support, RQ3).

Chapter 2 focuses on the automated detection of early planktonic salmon lice in seawater. This chapter addresses the feasibility of image-based detection by applying machine learning methods to identify early planktonic salmon lice from seawater images. A high-throughput seawater image acquisition setup is designed, and images of salmon lice are collected from concentrated seawater samples. Different detection models are evaluated to establish a baseline for machine-learning-based detection.

Chapter 3 addresses the data limitations underlying this detection task: planktonic lice are extremely sparse, and annotation is time-consuming even for laboratory-hatched specimens. In addition, image domains may differ across sampling times, locations, and acquisition conditions, creating further challenges for robust detection. Chapter 3 introduces a synthetic image generation framework, combining limited annotated lice images with large amounts of unlabelled background images to increase both the quantity and diversity of training data without additional manual annotation. Several detection models are evaluated to assess the utility of the synthetic data, including its effect on real-world detection performance.

Chapter 4 further extends the synthetic data approach from offline generation to online generation. While offline synthesis improves detection performance, the combinations of targets, impurities, and backgrounds in a pre-generated dataset remain fixed. Chapter 4 proposes Salmon Lice Detection based on Online Synthetic Training (SDOST), an online synthesis method in which instances are sampled and composed with augmented backgrounds in real time at each training step. A feature-based replay mechanism and fast gradient sign method (FGSM)-based adversarial regularization are introduced to stabilize training and reduce reliance on shortcut cues from instance pasting. The chapter evaluates whether this approach further improves detection under extremely data-scarce conditions.

Chapter 5 shifts the focus from farm-state perception to animal welfare modelling. Chapter 5 proposes an NLP-based framework for the automatic extraction and classification of welfare-related scientific statements. To address the scarcity of negative samples, this chapter utilizes a positive-unlabeled learning strategy enhanced by ad-

versarial contrastive sampling. Several classification paradigms are compared, and an interactive visualization interface is introduced to support practical deployment. The framework's performance is evaluated through cross-domain validation.

Chapter 6 focuses on AI-agent-based decision support. It introduces Salmon Farm Multi-Agent Decision Support System (S-MADSS), a conceptual framework for decision support in aquaculture. The framework is structured in layers from the resource layer to the workflow layer, and coordinates multiple components within a flexible decision-support architecture. RAG is used to connect agents with external legal, regulatory, policy, and management-relevant knowledge sources, grounding decisions in traceable information rather than relying only on the internal knowledge of LLMs. This chapter explores how AI agents can support information integration and decision reasoning under legal, policy, and management constraints. In the longer term, such a framework could integrate modules related to salmon lice detection, welfare assessment, regulatory reasoning, and management planning.

Finally, Chapter 7 provides a general discussion of the thesis. It synthesizes the findings from the preceding chapters and reflects on the extent to which the identified knowledge gaps have been addressed. It also discusses the broader implications of the work for AI-supported salmon farming and sustainable aquaculture management, identifies the limitations of the current studies, and outlines directions for future research.

Chapter 2

Rapid detection of salmon louse larvae in seawater based on machine learning

This chapter has been published as:

Zhang, C., Bracke, M., da Silva Torres, R., & Gansel, L. C. (2024). Rapid detection of salmon louse larvae in seawater based on machine learning. *Aquaculture*, 592, 741252. https://doi.org/10.1016/j.aquaculture.2024.741252

Abstract

The proliferation of salmon lice is one of the most challenging problems in salmon farming, seriously affecting the welfare of both farmed and wild salmonids. The detection of salmon lice in seawater, especially in the (pre-) infectious stage, supports the accurate management of salmon lice proliferation. Methods for detecting salmon lice in seawater to date involve traditional light and fluorescence microscopy and Polymerase Chain Reaction (PCR) methods. However, these methods require sample pre-processing and are laborious and time-consuming, rendering them difficult to apply for the task of continuous, rapid salmon lice detection in large water volumes. This study aims to develop a rapid and easy-to-use method for the detection of salmon lice in seawater. A novel approach was introduced for the detection of salmon lice at different larval stages in seawater by leveraging machine learning. A setup for the rapid acquisition of images of salmon lice in seawater was developed, and a dataset comprising 8316 images encompassing salmon lice "Nauplius Live," "Nauplius Shell" and "Copepodite" was created. Then 16 object detection models including one- and two-stage Convolutional Neural Network (CNN-) based models and Transformer-based models were trained, tested, and compared on this comprehensive dataset. RT-DETR showed the best performance among all models tested with a precision of 0.972 and a recall of 0.978. The YOLO series models, except for YOLOv3, showcased remarkable inference speeds (ranging from 152 to 416 frames per second) while maintaining high precision (ranging from 0.968 to 0.977) and recall (ranging from 0.964 to 0.970). To our knowledge, this is the first time machine learning has been applied to the task of rapidly detecting salmon louse larvae at different stages in seawater samples. The results demonstrate that the machine learning approaches can achieve rapid easy-to-use and cost-effective detection of salmon lice in seawater. In the future, this study has the potential for application in measuring salmon louse larvae numbers in aquatic environments, supporting the enhancement of the welfare of both farmed and wild salmon.

2.1 Introduction

Salmon aquaculture started in the 1960s (Aunsmo et al., 2023), and it has since become an important industry in several countries, including Norway, Chile, Canada, Scotland, and the Faroe Islands. The Norwegian salmon industry is producing more than half of the globally farmed salmon today (Iversen et al., 2020) having created about 7,000 jobs and NOK 102 billion in economic value in 2022 (Fiskeridirektoratet, 2023), thereby becoming one of Norway's largest export industries by economic value (Taranger et al., 2015).

The fast growth of intensive salmon farming inevitably comes with biological challenges. One of these originates from the interaction of farmed and wild fish with parasites. The salmon louse (*Lepeophtheirus salmonis*), feeding on the skin and mucus of fish, is one of the most problematic ectoparasites in salmon sea-cage farming in the northern hemisphere (Bui et al., 2021; Olaussen, 2018). In 2018, the costs associated with salmon lice control in the production of 1.2 million tonnes of salmon in Norway were estimated at 520 million EUR (Rosendal & Olesen, 2022). Salmon lice have a simple life cycle with eight stages, which are nauplius 1, nauplius 2, copepodite, chalimus 1, chalimus 2, pre-adult 1, pre-adult 2, and adult (Hamre et al., 2013). After hatching from eggs, they exist as free-living non-infectious nauplii (Figure 2.1(a)) through two nauplius stages (Bui et al., 2021). Salmon louse nauplii have limited swimming ability and are about 0.5 to 0.6 mm in length (Schram, 1993). The following copepodite stage (Figure 2.1(b)) is the infectious stage where the salmon lice try to seek out host salmon and parasitize on them. During this stage, salmon louse copepodites grow up to 0.7 to 0.8 mm long and 0.2 to 0.3 mm wide (Schram, 1993). Upon attaching to the host salmon, they undergo four additional stages of growth before reaching the adult stage. As salmon lice grow, they shed their shells to accommodate their growing bodies (Figure 2.1(c)). Salmon lice cause multifaceted losses in salmon farming, including chronic stress, growth retardation, secondary disease, mortality, and those caused by salmon louse treatment (Mustafa et al., 2001; Jeong et al., 2021). In addition to the effects on farmed salmon, salmon louse recruitment from farmed salmon is recognized as a major hazard for wild salmon (Serra-Llinares et al., 2014). Open net cage farming of salmonids adds a large number of host fish for salmon lice to the ecosystems fish farms are placed in, which consequently can produce large numbers of salmon lice. A large salmon farm may contain over one million fish and has the potential to produce millions of salmon lice daily. Over 500 fish farms are usually active simultaneously along the Norwegian coast.[1] Salmon lice spread with currents and can travel up to 200 km in 10 days, although most are scattered over 20

[1] `https://www.fiskeridir.no/Akvakultur/Tall-og-analyse/Biomassestatistikk/Biomassestatistikk-etter-produksjonsomraade` (As of Feb. 2024).



(a) Nauplius live (b) Copepodite (c) Nauplius shell

Figure 2.1: Example images of salmon louse nauplius live, copepodite, and nauplius shell. Nauplius live (a) includes nauplius 1 and 2. They are very similar in appearance and are not infectious. Copepodite (b) is the infectious stage that will try to parasitize on host fish. Nauplius shell (c) is shed by salmon louse larva (nauplius live) as it grows and is similar in appearance to nauplius live but transparent.

to 30 km (Serra-Llinares et al., 2014). The larvae develop into infectious copepodites at speeds dependent on the sea temperature and can infect migrating wild Atlantic salmon, as well as sea trout (Asplin et al., 2011) within large sea areas around fish farms.

Due to the threat posed by salmon lice, proper management of salmon louse populations is necessary for further sustainable development of the salmon industry. According to Norwegian regulations today, the average number of adult female lice per fish in a facility must be less than 0.5 during most of the year, and 0.2 during the seaward migration of wild salmon smolts (Lerøy Seafood Group, 2022). Norwegian authorities therefore demand that salmon farmers count and report salmon louse infestations weekly when the water temperature exceeds 4 °C. These numbers serve as indicators to assess the infection risk from farm-released salmon louse larvae for wild salmonids. Lakseluskartet[2] is a map of salmon louse distribution estimates, which uses information on lice numbers on farmed fish, biological models for the developmental speed of live larvae at different temperatures together with sea temperature and ocean current models to predict salmon louse larvae numbers in the sea. However, the current data for verification, validation, and model input is very limited, primarily relying on reported numbers of salmon lice on both farmed and some migrating wild salmon smolts (Gupta et al., 2022; Jeong & Revie, 2020), along with few sentinel fish in small cages at discrete locations. This remains a secondary measure, and there is a compelling need for direct measurements of (pre-) infectious larvae in the sea as a more appropriate primary measure. Accurate knowledge of salmon louse larvae

[2] `https://www.hi.no/forskning/marine-data-forskningsdata/lakseluskart/html/lakseluskart.html` (As of Oct. 2023).

distribution, transport mechanisms, and paths in the sea is crucial for area planning and determining optimal fish farm locations to minimize infection risks of farmed and wild fish. It may also play a vital role in production planning, deciding when and where to release fish and how to sequence operations like de-licing. Additionally, this information is essential for implementing operational measures to counteract contact between infectious copepodites and fish, as well as developing effective solutions to remove salmon louse larvae from the sea. Monitoring larvae at different depths would alleviate potential sampling biases (Sauphar et al., 2023). Therefore, to obtain information on the distribution of salmon lice at different locations and depths in the seawater, it is highly desirable to extract seawater from different locations and depths for salmon louse larvae detection.

Therefore, methods that can detect the numbers of salmon louse larvae in seawater by direct measurements are direly needed. Several methods (Figure 2.2) have been used to estimate the population or density of salmon lice (á Norði et al., 2015; Costelloe et al., 1998; Nelson et al., 2018; Skarðhamar et al., 2019). Traditional plankton analysis by light microscopy is one option, but it is highly time-consuming. Fluorescence microscopy is a vision-based method for detecting nauplii or copepodites, which involves illuminating the sample using UV light. A stereomicroscope with an emission filter is used for observation. This method allows copepodites to appear slightly brighter and with a different hue than other particles in the sample (Bui et al., 2021). Another vision-based method is automated fluid imaging. This involves drawing the fluid through a glass flow cell and capturing images using a flow cytometer and microscope (Bui et al., 2021). The method uses saline ethanol to dilute the liquid containing salmon lice. Although post-processing software helps sort the images to remove blank images, manual observation of the number and location of salmon lice in a large number of images is still required and demanding (Bui et al., 2021). Another method that can directly detect salmon lice is the molecular method (Rocha et al., 2019), such as droplet digital Polymerase Chain Reaction (PCR), quantitative fraction PCR, and quantitative PCR (Bui et al., 2021). The molecular method reduces the reliance on human operators compared to vision-based detection methods and can achieve higher accuracy. However, molecular methods require more complex processes to handle and prepare the samples, which is usually done in the laboratory. Furthermore, PCR methods cannot yet be used to distinguish between developmental stages of salmon lice (Bui et al., 2021). They are therefore less suitable for detailed modeling of the distribution and development of larvae. To date, both vision-based methods and molecular-based methods are labor-intensive and require specialized operators for processing samples or observation. This poses significant limitations in accurately and swiftly recording the distribution of salmon louse larvae in seawater at various locations, resulting in risk analyses for transmission of (pre-) infectious stages

at given locations.

In a two-year study, Penson et al (Penston et al., 2008) used traditional intensive sampling methods to build statistical models of salmon lice that include explanatory variables, such as time, depth, and station. The results of the study showed differences in the temporal and spatial distribution of salmon lice in seawater. Salmon farms are the main source of salmon lice larvae during the production cycle and spread more widely after they grow into copepodite. In seawater, the density of salmon lice in seawater may be relatively low compared to other plankton. However, even at a density of 1 louse per cubic meter, the impact on small salmonid smolts can be significant (Moriarty et al., 2023). Though the prevalence of salmon lice in seawater is lower than that of other parasites, it still has a very serious impact on farmed or wild salmon. Therefore, the detection of salmon lice in seawater requires analyzing a large amount of seawater and at the same time needs to ensure high accuracy and recall. This is a crucial reason why automated rapid detection of salmon lice in seawater is challenging. Due to the patchy distribution of plankton and the generally low concentration of salmon louse larvae in the sea, even on salmon farms, a fast and accurate method to detect salmon lice at different stages in various locations in the sea is urgently needed.

Rapidly evolving artificial intelligence offers the possibility of achieving this. Machine learning already has a wide range of applications in aquatic research, such as fish wound detection and detection of parasites attached to fish (Gupta et al., 2022; Pettersen et al., 2019). However, to the best of our knowledge, there are no published studies using machine learning to detect and distinguish larval stages of salmon lice in seawater.

In this study, we used machine learning-based approaches for the real-time detection of salmon lice in video images captured from concentrated seawater that also contained some plankton (Figure 2.2). The primary objective of this method is to enable thorough yet feasible analyses of different salmon louse stages in seawater. Therefore, a dataset comprising salmon louse nauplius and copepodite stages was created by using artificially 'infected' fairly unpolluted seawater with laboratory-hatched salmon lice. Multiple machine learning-based object detection algorithms were systematically implemented, compared, and analyzed to identify suitable approaches for the salmon lice detection task. To the best of our knowledge, this is the first time detection of salmon lice at different stages in seawater based on machine learning approaches has been investigated. Its successful implementation will act as a prerequisite for systems dedicated to monitoring salmon louse larva concentrations in the sea in real-time. The resulting data can be invaluable for both area and production planning, as well as for immediate operational responses to elevated infection risks. Furthermore, this

model holds the potential to contribute to the development of novel solutions aimed at mitigating contact between fish and infectious copepodite stages, along with enabling the selective removal of salmon louse larvae.

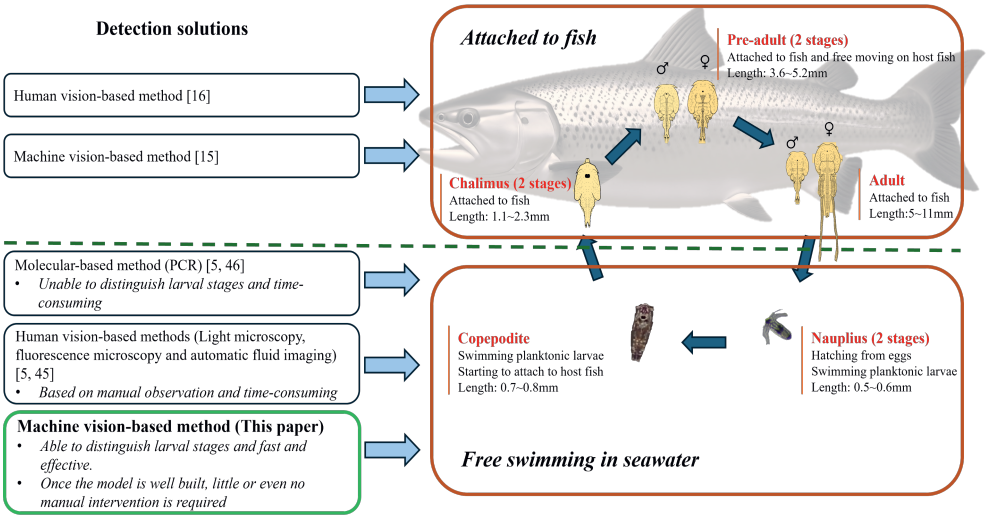


Figure 2.2: Detection methods for salmon lice at different growth stages (in seawater and attached to fish).

2.2 Materials and methods

2.2.1 Data collection

2.2.1.1 Data acquisition platform

The small size of salmon louse larvae (nauplius and copepodite), typically 0.5 to 0.7 mm in length (Schram, 1993), poses a significant challenge for direct observation by the naked eye or normal photographic equipment. To collect data at the relevant resolution of the objects of interest, a salmon louse data acquisition platform was designed, which allows us to collect both still images and video recordings while resolving details in the µm range. Seawater samples containing some plankton and salmon louse larvae were pumped from the sample container (a) through the glass channel (f) with an image acquisition volume of 60 mm^3 ($10 \times 6 \times 1$ mm ($length \times width \times height$)). Images or videos were captured using a Canon 80D digital SLR camera with a LAOWA 60mm f2.8 2:1 Macro lens (d), illuminated by the light source (e) (Figure 2.3). Illumination was from the sides and below the image acquisition section to result in a sufficient contrast between the black background and the organisms of interest. The side illumination aims to enhance seeing both external and internal morphological and anatomical details. The resolution of the image is 1920×1080 pixels, resulting

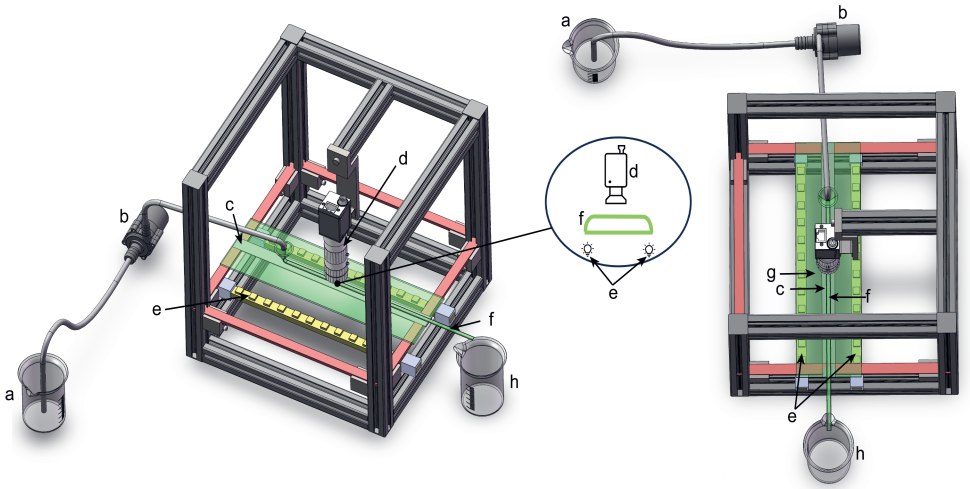


Figure 2.3: Schematic diagram of the data acquisition platform. A small pump (b) delivers the liquid, which includes salmon lice, from a sample container (a) into the glass channel (f). A camera with a macro lens (d) above the glass channel (f) captures video or images of the liquid flowing through the glass channel. Finally, the liquid flows into a second sample container (h). Light sources (e) are located on either side below the glass channel and provide illumination of the liquid through the glass panel (g). A shading strip (c) below the glass channel prevents the light sources from directly illuminating the liquid through the bottom of the glass channel.

in a pixel resolution of about 5 µm. During operation, seawater containing salmon lice passes through the channel at a steady speed and the capacity of the current system is 250 ml of up-concentrated water sample per hour. We found up to over 200 organisms (e.g., salmon louse larvae and various species of plankton) in one image or video frame being acceptable with respect to the potential overlap of organisms, giving an analytical potential of about 1 million organisms per hour. Our samples were pre-filtered to remove organisms much smaller (< 140 µm) and larger (> 2000 µm) than salmon lice. The volume of seawater available for salmon lice analysis within a given time period depends on the concentration of organisms within this size range in the sea. This concentration will vary with time, as primary production and plankton dynamics change at different temporal scales, the most important for our application being at scales from hours to months. To obtain clear images and reduce the trailing shadow, the exposure time of the camera needs to satisfy the condition of Equation 2.1. In this particular case, 1/3000 second was employed as the exposure time.

$$T \leq \frac{KR}{V} \tag{2.1}$$

where T denotes the camera exposure time (s), K denotes the permissible level of image blurring (pixel), R denotes the camera resolution (mm/pixel) and V denotes the flow speed of the water (mm/s).

2.2.1.2 Data Acquisition Process

Training a machine learning model for object detection often requires a substantial amount of data. However, finding and annotating salmon lice in proper seawater samples is very time-consuming and laborious due to the low concentration (á Norði et al., 2015). Therefore, wild adult female salmon lice were collected at a salmon farm site to form the first training dataset. Their eggs were hatched in the laboratory and placed in seawater before they were transferred to the image acquisition platform, where videos and images were captured. The specific method is described below:

a. Sample collection: Adult female salmon lice were sampled from farmed salmon at a commercial farm site for Atlantic salmon (Gudmundset, locality number 12265, near Ålesund, Norway). Egg strings were placed in aerated glass beakers containing filtered seawater in a climate chamber at 12°C. Seawater was exchanged every few days.

b. Nauplius video collection: Videos were collected of salmon louse nauplii approximately 10-40 day-degrees after hatching, using the data acquisition platform.

c. Copepodites video collection: The transformation from nauplii to copepodites occurred after accumulating approximately 50-70 day degrees. Video recordings of copepodites were then obtained using the data acquisition platform.

d. Video pre-processing: Images were extracted from the video recordings frame by frame and saved as a series of images. Images contained some other particles (Figure 2.4). The identification of salmon lice can be complicated by larvae freely rotating in three dimensions. Labeling of salmon louse larvae was done by a marine biologist trained in salmon lice identification to ensure proper detection and labeling of salmon louse nauplius and copepodite stages, as well as nauplius shells.

2.2.2 Datasets construction

In this study, emphasis was placed on the detection of three classes: nauplius live, nauplius shell, and copepodite. LabelMe,[3] an image annotation tool, was used to label the images. LabelMe uses a bounding box to mark the object, and the enclosed part of the mark is the foreground, including "Nauplius Live," "Nauplius Shell," and "Copepodite" in this study. The rest of the image is the background. In this paper, models use two different data standards, textual format and COCO format (Lin et al., 2014).

The annotated images formed a dataset labeled as Dataset NC (Nauplii & Copepodite). The Dataset NC consists of 8316 images containing "Nauplius Live" (n=4660), "Nauplius Shell" (n=5065), and "Copepodite"(n=2049) (Figure 2.4). In computer vision tasks, the dataset is normally divided into three separate parts, namely the training set, the validation set, and the test set. The training dataset is used to train the model through a series of iterations known as epochs. During each epoch, the model goes through the whole training set and updates the weights and biases. The validation set is used to monitor the training process and select the best model among the models produced for all epochs. The test set is used to evaluate the performance of the final model. In machine learning, the training set commonly comprises anywhere from 80% to 50% of the total dataset, with the remaining 20% to 50% allocated for validation and testing (Joseph, 2022). However, there is no clear guidance for the specific partitioning of the dataset. Therefore, based on the size of the dataset, this study adopts the commonly used ratio of 7:1:2, where 70% of the dataset is allocated for training purposes. In this paper, Dataset NC was randomly split, resulting in a training set containing 5821 images, a validation set of 831 images, and a test set with 1664 images (Table 2.1). Due to the later collection of copepodite than nauplius, a preliminary dataset labeled Dataset NLS (Nauplius Live & N. Shell) was created before constructing the final Dataset NC (Table 2.1). The Dataset NLS is a subset of Dataset NC and contains 6515 images with only "Nauplius Live" (n=4386) or "Nauplius Shell" (n=4559) (Figure 2.4). The models selected for this study were also trained and tested on Dataset NLS separately. Detailed test results for each model on the dataset NLS are shown in A.3.1.

2.2.3 Salmon lice detection methods

In this paper, 16 object detection models were selected to detect salmon lice. The models include 14 Convolutional Neural Network (CNN)- and two Transformer-based models. CNNs have been widely used in the field of object detection for a long time. CNNs have powerful spatial feature extraction capabilities that rely on convolution

[3] `https://github.com/wkentaro/labelme` (As of Oct. 2023).

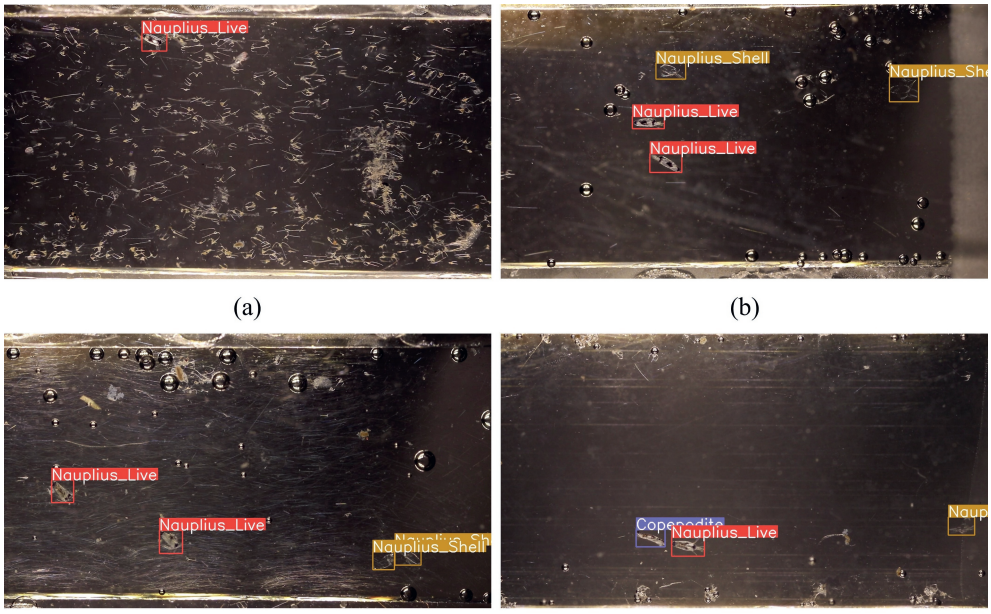


Figure 2.4: Example images of Dataset NLS and Dataset NC. Our initial Dataset NLS contains either "Nauplius Live," or "Nauplius Shell," or both, as shown in Figure 2.4(a), (b), and (c). Our final Dataset NC contains any one or more of three annotated objects namely "Nauplius Live," "Nauplius Shell," or "Copepodite." See Figure 2.4(d) for an example.

Table 2.1: Description of Dataset NLS and NC.

Dataset	Set	Number of instances			Number of images
		Nauplius_Live	Nauplius_Shell	Copepodite	
Dataset NLS	Train	3063	3157	-	4560
	Validation	431	483	-	651
	Test	892	919	-	1304
	Total	4386	4559	-	6515
Dataset NC	Train	3262	3521	1426	5821
	Validation	461	515	198	831
	Test	937	1029	425	1664
	Total	4660	5065	2049	8316

kernels to continuously extract features. CNN-based models can be categorized into two groups: two-stage detection models and one-stage detection models. The two-stage CNN-based models first generate region proposals of salmon lice and then predict the objects and bounding boxes for each region proposal (Diwan et al., 2023). This kind of model has the potential to achieve higher accuracy than the one-stage CNN-

based model but is slower due to multi-stage calculation. Faster R-CNN (Ren et al., 2015), Sparse R-CNN (Sun et al., 2021), and Cascade R-CNN (Cai & Vasconcelos, 2018) were chosen as two-stage CNN-based models in this study. In contrast, the one-stage CNN-based model can generate outputs both with bounding boxes of salmon lice and classifications with only feeding input to the network once. These models are fast but with a relatively low accuracy. In this work, Single Shot MultiBox Detector (SSD) (Liu et al., 2016), RetinaNet (Lin et al., 2017b) and You Only Look Once (YOLO) series were chosen as one-stage CNN-based models. The Transformer-based models rely on the attention mechanism, comprising both self-attention and cross-attention, to draw global dependencies between input and output (Meng et al., 2021). The Transformer-based models started out as an application for machine translation tasks (Vaswani et al., 2017). Later, it was gradually applied to various fields including natural language processing and computer vision. Transformer-based object detection models require positional encoding to understand the spatial arrangement of objects within the image, as they do not have inherent spatial understanding like CNN-based models do. Therefore, Transformer-based models are excellent at handling global context and long-range dependencies, making them suitable for capturing relationships between objects in a whole image. In this study, Deformable DEtection TRansformer (Deformable DETR) (Zhu et al., 2021) and Realtime-DEtection TRansformer (RT-DETR) (Zhao et al., 2024b) were incorporated as Transformer-based models. A more detailed description of the chosen models can be found in A.1.1.

2.2.4 Model training

In this study, all experiments were conducted using the PyTorch framework (Paszke et al., 2019). The computer configuration utilized for these experiments is detailed as follows: an Intel i9-13950HX CPU, 32GB of RAM, an NVIDIA RTX A5500 GPU, and the Pop!_OS 22.04 LTS operating system. During training, the weights of the network are updated based on data, which can pose challenges in achieving good performance on a small dataset. Transfer learning mitigates this issue by allowing models pre-trained on larger datasets to fine-tune their weights on smaller datasets, thereby enhancing the performance of models trained on the small dataset. In this paper, all models were pre-trained on the COCO dataset. Then, training was performed on the salmon lice dataset using transfer learning. More details on the training process and the parameters of each model employed in this paper can be found in A.1.2.

2.2.5 Ground truth and evaluation metrics

2.2.5.1 Ground truth

In this paper, salmon lice as identified by an expert in marine biology are used as ground truth. Due to the relatively simple background of salmon lice images and easy-to-identify salmon louse larvae, the ground truth can be considered highly accurate.

2.2.5.2 Evaluation metrics

All models have been tested on the test dataset. In this paper, precision, recall, Average Precision (AP), F1 score, and speed were used to evaluate the performance of each model. Precision is the probability of being a positive sample out of all the samples predicted to be positive. It represents the precision of predicting the outcome of a positive sample. The samples can be "Nauplius Live," "Nauplius Shell," or "Copepodite." A positive sample refers to an image region in the model's predicted results, where the Intersection over Union (IoU) with the ground truth bounding box labeled by the marine biologist exceeds a threshold. Recall is the probability of being predicted as a positive sample among the actual positive samples. Precision and recall have opposite tendencies, and when precision increases, often the value of recall decreases. The PR (Precision-Recall) curve represents the relationship between precision and recall. Average Precision (AP) refers to the area under the PR curve. The formulas for precision, recall, AP, and F1 are shown below. AP_i represents the value of AP when the IoU is equal to i (Figure 2.5). mAP refers to the mean value of AP. The F1 score, also known as the Balanced F-Score, is the harmonic mean of precision and recall. Frame Per Second (FPS) was used to denote the inference speed of the model.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.4}$$

$$AP = \int_0^1 Precision(Recall) dRecall \tag{2.5}$$

$$mAP = \frac{\sum_1^N AP}{N} \tag{2.6}$$

$$mAP_{50:95} = \frac{1}{10}(mAP_{50} + mAP_{55} + \cdots + mAP_{90} + mAP_{95}) \tag{2.7}$$

where True Positive (TP) indicates the number of positive samples that the model correctly predicts as positive. False Negative (FN) indicates the number of positive samples that the model predicts as negative. False Positive (FP) indicates the number of negative samples that the model predicts as positive. True Negative (TN) indicates the number of negative samples that the model correctly predicts as negative.

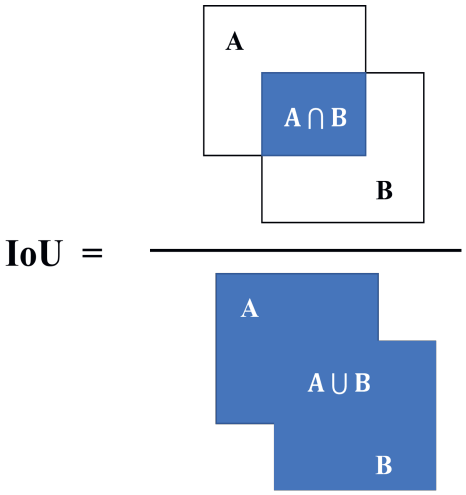


Figure 2.5: Calculation of IoU (Intersection over union). A represents the ground truth bounding box and B represents the predicted bounding box. ⋂ refers to "and." ⋃ refers to "or."

2.3 Results

2.3.1 Salmon louse detection results in Dataset NC

2.3.1.1 Overall detection results

All 16 models were trained, validated, and tested on the Dataset NC. The overall results on the performance of detection and inference speed are shown in Table 2.2.

The recall values obtained from the two-stage model ranged from 0.922 to 0.952, and the precision ranged from 0.938 to 0.953. Among all the two-stage CNN-based

Table 2.2: Overall detection results on the test set of Dataset NC.

Model category	Model	mAP_{50}	$mAP_{50:95}$	Precision	Recall	F1	Model size (MB)	FPS	Inference time (ms)
Two-stage CNN-based	Faster-R-CNN 50	0.962	0.635	0.950	0.927	0.938	167.3	24.6	40.65
	Faster-R-CNN 101	0.956	0.629	0.938	0.922	0.930	399.5	18.6	53.76
	Cascade R-CNN	0.961	0.666	**0.953**	0.938	0.945	354.1	15.7	63.69
	Sparse R-CNN50	**0.970**	**0.668**	0.948	**0.952**	**0.950**	425.5	**25.0**	40.00
	Sparse R-CNN101	0.967	0.666	**0.953**	0.946	**0.950**	501.7	18.7	53.48
One-stage CNN-based	SSD300	0.919	0.551	0.944	0.890	0.916	137.2	122.2	8.18
	SSD512	0.903	0.542	0.910	0.854	0.880	144.2	66.8	14.97
	RetinaNet 50	0.963	0.638	0.948	0.916	0.932	152.1	27.5	36.50
	RetinaNet 101	0.966	0.654	0.938	0.928	0.933	228.4	19.6	51.02
	YOLOv3	0.953	0.576	0.959	0.952	0.955	236.6	144.5	6.92
	YOLOv5su	**0.984***	**0.759***	**0.977**	0.964	0.970	17.6	212.8	4.70
	YOLOX	0.974	0.702	0.974	0.969	**0.971**	36.1	151.6	6.70
	YOLOv7	**0.984**	0.731	0.968	**0.970**	0.969	75.6	196.1	5.10
	YOLOv8n	0.981	0.740	0.972	0.965	0.968	6.5	**416.0***	2.40
Transformer	Deformable DETR	0.975	0.662	**0.978***	0.960	0.969	254.7	13.8	72.46
	RT-DETR	**0.980**	**0.736**	0.972	**0.978***	**0.975***	66.5	**46.9**	21.32

Note: A green background with $*$ highlights that this value represents the highest among all models for that specific metric. A value in bold font signifies that it holds the highest position for that metric within the model category, encompassing two-stage CNN-based models, one-stage CNN-based models, and Transformer-based models. The mAP_{50} stands for the mean value of average precision of all classes when the intersection over union (IoU) is equal to 50%. The $mAP_{50:95}$ stands for the average value of from mAP_{50} to mAP_{95} in 5 % intervals of IoU. Precision stands for the probability of being a positive sample out of all the samples predicted to be positive. Recall is the probability of being predicted as a positive sample among the actual positive samples. The F1 score stands for the harmonic mean of precision and recall. The model size stands for the space needed for model storage. FPS (Frame Per Second) stands for the number of images the model can process per second. The inference time stands for the time used by the model to infer one image to the output result, including classes of objects and locations of objects.

models, Sparse R-CNN 50 excelled in both mAP_{50}, $mAP_{50:95}$, recall, and F1 metrics. Both Sparse R-CNN 101 and Sparse R-CNN 50 achieved identical F1 scores of 0.95. In terms of precision, both Cascade R-CNN and Sparse R-CNN 101 achieved the highest value of 0.953. Faster R-CNN 101 achieved the lowest values among all metrics, indicating some objects were not detected correctly. Faster R-CNN 50 with fewer parameters, exhibited superior performance metrics compared to Faster R-CNN 101 across the board. It is also worth noting that the two-stage CNN-based models generally exhibited relatively low running speeds with lower than 25 FPS, which poses a challenge for real-time detection.

The one-stage CNN-based models, except for SSD, had similar or better performance compared to the two-stage CNN-based models. All YOLO series models, excluding YOLOv3, showed superior performance, with F1 scores over 0.968. YOLOv5su achieved the best values in $mAP_{50:95}$ and precision at 0.759 and 0.977, respectively. YOLOv7 achieved the best value of recall of 0.97. YOLOX, with its balanced and high values in both precision and recall, attained the top position in the F1 score. Due to the low recall value obtained by YOLOv5su, the F1 score was also affected, but still close to the best value. The SSD models exhibited suboptimal performance. RetinaNet surpassed both the SSD models and YOLOv3 but fell short of the performance achieved by other models in the YOLO series. Notably, the YOLO series models outperformed the other types of models in terms of speed, with all YOLO models achieving an FPS of over 140 for real-time detection. YOLOv8n stood out in particular, with an impressive FPS of 416 for a minimum model size of 6.5 MB.

In contrast, the SSD models were over 60 FPS, but the detection performance was not as good and the RetinaNet models fell behind with speeds below 27.5 FPS for detection.

Among Transformer-based models, RT-DETR showed a good performance in mAP_{50}, $mAP_{50:95}$, recall, and F1 metrics. However, in terms of precision, Deformable DETR achieved a value of 0.978, which was higher than the RT-DETR of 0.972. In terms of speed, RT-DETR achieved 46.9 FPS while Deformable DETR achieved a speed of only 13.8 FPS.

Figure 2.6 shows an example of predicted results from YOLOv5su and RT-DETR for two salmon lice images. In Figure 2.6(a), which includes some impurities and other organisms, YOLOv5su and RT-DETR successfully detected all the objects. In Figure 2.6(b) with a clean background but many objects, YOLOv5su and RT-DETR could detect all instances of the classes "Nauplius Live" and "Copepodite," but it incorrectly identified debris in the background as "Nauplius Shell."

2.3.1.2 Detection results for each class

The results of F1 scores obtained by each model for each class in the test set of Dataset NC and inference speed are visualized in Figure 2.7, Figure 2.8, and Figure 2.9.

For the results of the class "Nauplius Live," both YOLO series models and Transformer-based models outperformed the two-stage CNN-based models in terms of F1 scores (Figure 2.7). The F1 scores for the YOLO series ranged from 0.972 to 0.988. On the other hand, the two-stage CNN-based models achieved F1 scores in the range of 0.954 to 0.969. The SSD series achieved the lowest F1 scores of all tested models. Both YOLOX and RT-DETR obtained the highest F1 score of 0.988. YOLOv7 followed closely with an F1 score of 0.984. Deformable DETR and YOLOv5su shared the same F1 score of 0.979.

The results of class "Nauplius Shell" are shown in Figure 2.8. Similar to the results of "Nauplius Live," both YOLO series models and Transformer-based models outperformed the two-stage CNN-based models in terms of F1 scores. The SSD series model obtained the lowest F1 scores among all models. Both YOLOv7 and RT-DETR obtained the highest F1 score of 0.953, with YOLOv5su, Deformable DETR, YOLOX, and YOLOv8n following closely.

The results of class "Copepodite" are illustrated in Figure 2.9. RT-DETR consistently outperformed the other models with the highest F1 score of 0.983. Notably, the YOLO series and Transformer-based models maintained robust performance, achieving F1 scores ranging from 0.970 to 0.981 for YOLO models and 0.976 to 0.983 for

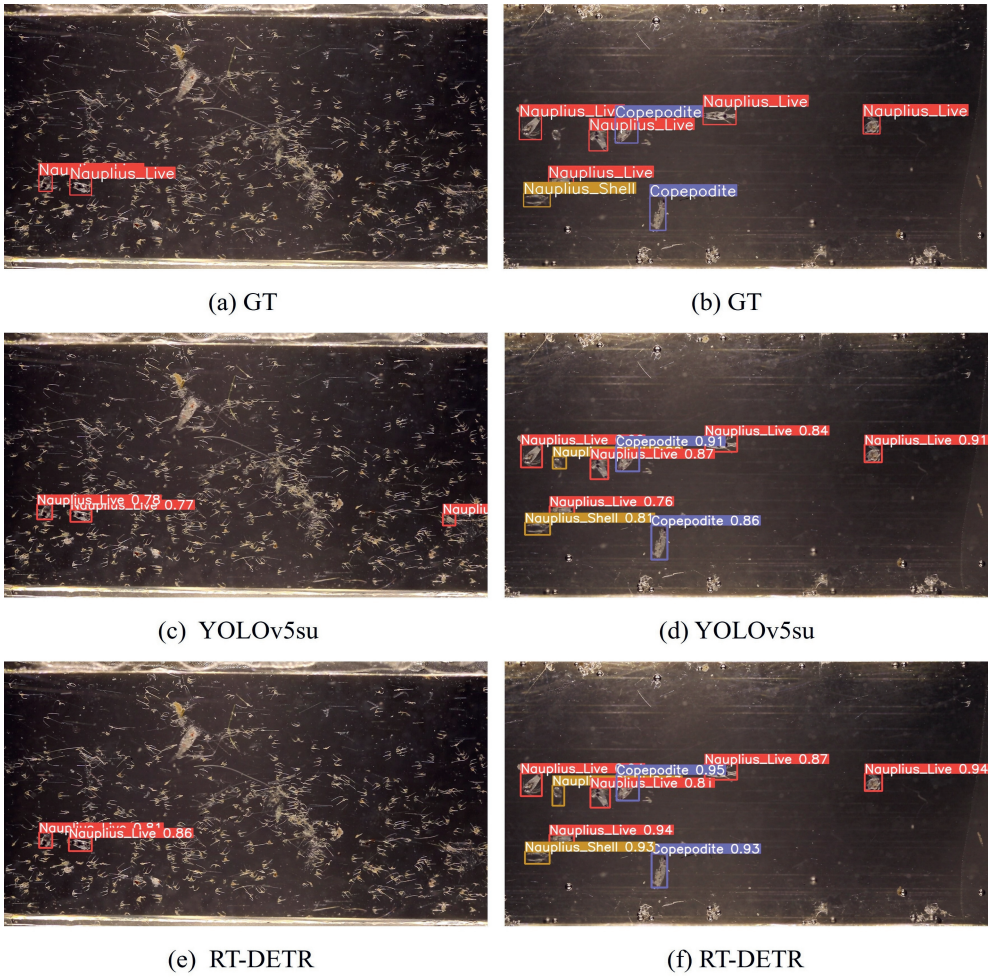


Figure 2.6: Detection results of YOLOv5su and RT-DETR on input images (a) and (b).

Transformer-based models. Compared to the results for "Nauplius Live" and "Nauplius Shell," some of the two-stage CNN-based models also achieved better performance for "Copepodite," such as Sparse R-CNN 50 and Sparse R-CNN 101, with an F1 score of 0.976 and 0.970, respectively. The SSD models exhibited the lowest F1 score among all models. The SSD300 obtained an F1 score of 0.918, while the SSD512, with a deeper network and more parameters, got a lower score of 0.840.

In addition to the F1 scores, the results for each metric obtained by each model for the "Nauplius Live," "Nauplius Shell" and "Copepodite" in the test set of dataset NC are shown in Tables A.5, A.6, and A.7.

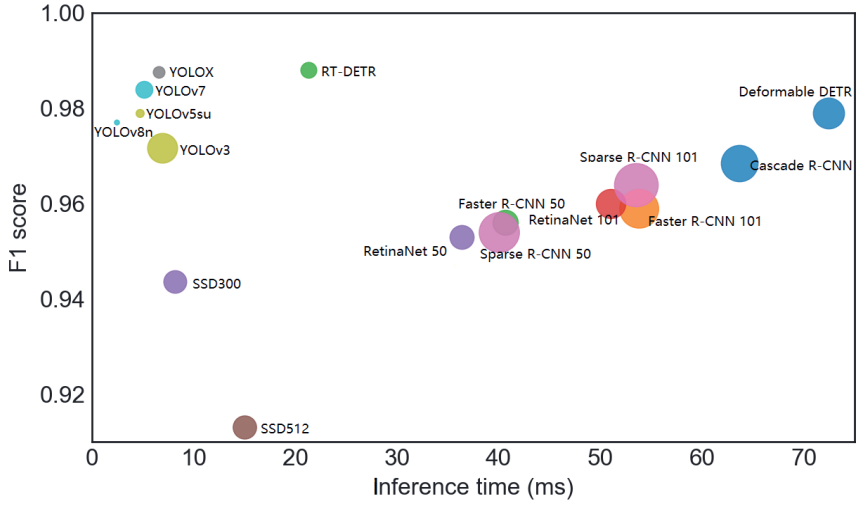


Figure 2.7: Detection results of Nauplius Live in Dataset NC. The size of the bubbles represents the size of the models. Bubbles positioned closer to the top-left corner refer to models that show better performance.

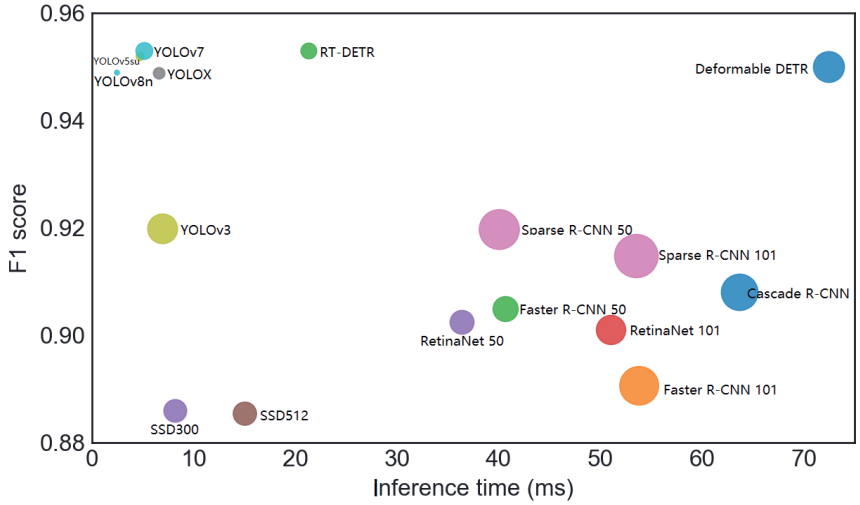


Figure 2.8: Detection results of Nauplius Shell in Dataset NC. The size of the bubbles represents the size of the models. Bubbles positioned closer to the top-left corner refer to models that show better performance.

2.3.2 Qualitative analysis

To comprehensively assess the effectiveness of machine learning for detecting 'wild' salmon lice in real seawater (rather than the laboratory-grown sea louse larvae used

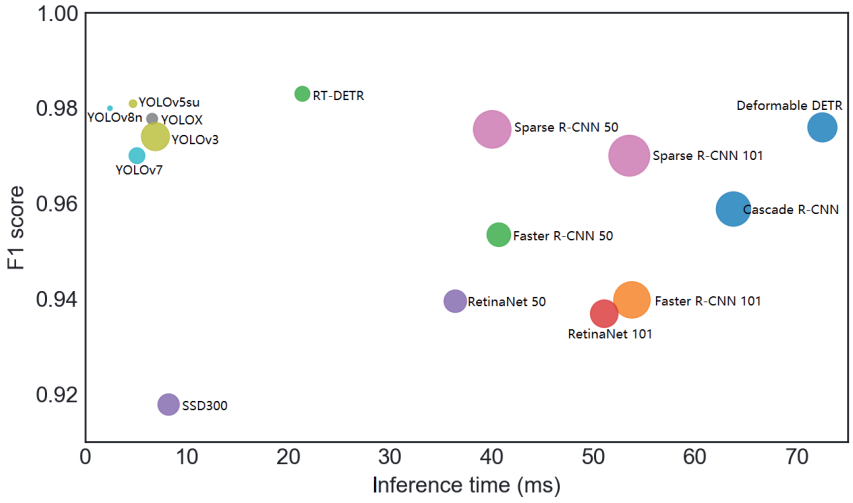


Figure 2.9: Detection results of Copepodite in Dataset NC. The size of the bubbles represents the size of the models. A bubble positioned closer to the top-left corner signifies the model shows better performance. SSD512 only obtained an F1 score of 0.84, which is not shown in this graph.

to train the models), we conducted video acquisition using seawater obtained from a salmon farm. Approximately 18 and 21 cubic meters of seawater were pumped respectively on different dates from inside a commercial fish cage (20.000 cubic meters in volume), stocked with about 76.000 Atlantic salmon with an average weight of about 3700 g (biomass of about 281 metric tons). The seawater was filtered over a plankton net with a mesh opening size of 140 µm. The samples were then filtered again through a filter with a mesh size of 1000 µm, so that the remaining sample contained all particles and organisms from the pumped water volume in the size range 140 - 1000 µm. The data acquisition platform described in Figure 2.3 was subsequently employed to record video footage of the two filtered samples. Both samples had very rich plankton communities, which were very different from the images in the Datasets NLS and NC. These samples were collected with video durations of 22 and 26 minutes, respectively. The videos were manually analyzed and counted for salmon louse numbers. Subsequently, we applied the YOLOv8n model trained on Dataset NC to predict and analyze the obtained footage. Both YOLOv8n and YOLOv5su performed automatic processing of the videos and cropped out predicted salmon lice images from the videos and saved them locally. Subsequently, the marine biologist counted the number of salmon lice in the video by analyzing the cropped images. In this complex background case, the model was still able to effectively detect salmon lice (Figure 2.10). In the two samples, YOLOv5su sucessfully detected all salmon lice with

Table 2.3: Detection results of 'wild' (i.e. non-laboratory bred) salmon louse larvae in filtered and concentrated seawater obtained from a commercial salmon farm using YOLOv5su and YOLOv8n. Confidence score is the probability of predicting that an anchor frame contains an object. During the prediction process, only a target with a confidence score above the confidence threshold is retained. Video duration refers to the length of the acquired video of the concentrated seawater samples collected using the data acquisition platform. Time cost refers to the time required to process the video using the model and to manually count the number of salmon louse larvae after processing.

Model	Confidence threshold	Sample	Video duration (minutes)	Time cost (minutes)	Number of salmon lice in sample	Number of salmon lice detected
YOLOv5su	0.75	NO.1	22	25	7	7
		NO.2	26	29	5	5
YOLOv8n	0.75	NO.1	22	25	7	7
		NO.2	26	28	5	3
	0.40	NO.1	22	27	7	7
		NO.2	26	31	5	5

a model confidence threshold set to 0.75. For YOLOv8n, successful detections were achieved for 7 out of 7 and 3 out of 5 salmon lice at the same confidence threshold, respectively However, when the confidence threshold was lowered to 0.4, YOLOv8n also detected all salmon lice in both videos, albeit with an increase in false positives (Table 2.3).

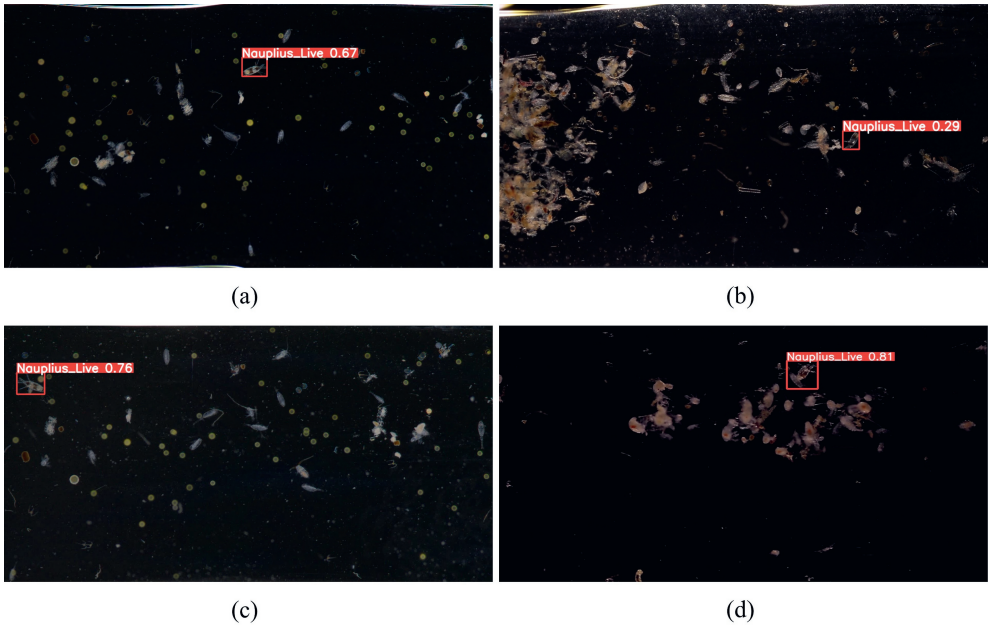


Figure 2.10: Examples of detection results of 'wild' salmon lice in the filtered and concentrated seawater using YOLOv8n.

2.4 Discussion

This study introduces an automated salmon lice detection system employing machine learning-based approaches, marking a pivotal advancement for salmon lice detection and quantification in the sea. Pioneering the use of machine learning models for this purpose, machine learning approaches not only offer a more efficient and accurate detection solution compared to traditional methods but also hold the potential for large-scale, cost-effective automation in salmon lice detection. Beyond the technological implications, our research establishes a foundation for the broader integration of machine learning in aquaculture monitoring. The significance of detecting salmon lice in the sea extends to critical applications, such as analyzing production and transport mechanisms, conducting risk assessments for the spread and infection of other farmed and wild salmonids, spatial planning of sea areas for salmonid production, and implementing operational measures to reduce infection risks at farm level. Currently, the reliance on secondary information, including the counting of salmon lice on farmed fish and the use of sentinel fish in small cages over short time periods to estimate salmon lice in the sea, underscores the need for proper primary data to enhance the accuracy and reliability of salmon lice spreading and development models.

For instance, the salmon lice map[4] is derived from the counting of salmon lice on farmed fish, but validating its accuracy across large areas in detail poses a significant challenge.

2.4.1 Performance analysis and application of different models

In terms of salmon lice detection methods, 16 different models were evaluated, each presenting unique advantages for specific usage requirements. When there is a need for extensive and flexible deployment of salmon lice detection devices, especially in large-scale scenarios, the YOLO series, particularly YOLOv8n with the inference speed of 416 FPS, emerges as a suitable choice. In situations where flexibility is paramount, emphasizing reduced false negatives (higher recall) and swift detection, models like YOLOv5su with an inference speed of 213 FPS and recall of 0.964 or YOLOv7 with an inference speed of 196 and recall of 0.97 become viable alternatives. The diverse selection of models allows for a nuanced approach, addressing varied needs in salmon lice detection applications.

Two-stage CNN-based models, from an architectural perspective, have a more complex computational process compared to one-stage models (Wu et al., 2020; Zou et al., 2023). This is reflected in the "coarse-to-fine" processing flow of the two-stage models (Zou et al., 2023). Two-stage CNN-based models generate the Region of Interest (RoI) using the Region Proposal Network (RPN) and then predict the objects and bounding boxes for the proposed region (Diwan et al., 2023; Joseph et al., 2021; Xiao et al., 2020). The one-stage models extract the features directly through the neural network and predict the classification and bounding boxes of the objects (Diwan et al., 2023; Xiao et al., 2020). The architecture of two-stage models ensures that the models are expected to achieve high recall or precision in some tasks, but it also introduces a higher level of complexity leading to a slower inference speed (Wu et al., 2020; Zou et al., 2023). The simple structure of the single-stage models, on the other hand, brings lower computational effort and faster inference but may also tend to negatively affect recall and precision to some extent (Wu et al., 2020; Zou et al., 2023). In this study, however, the opposite result was observed. The lower speed of the two-stage CNN-based model is mainly attributed to its two-stage CNN-based classification and boundary regression process, which increases the model size and affects its speed. Nonetheless, the two-stage CNN-based models did not perform better than the one-stage CNN-based models, based on the metrics employed, including mAP, precision, and recall. This discrepancy may be attributed to the release timeline, where the older two-stage CNN-based models used in this study face opti-

[4] `https://www.hi.no/forskning/marine-data-forskningsdata/lakseluskart/html/lakseluskart.html` (As of Jan. 2024).

mization challenges compared to the more recent one-stage CNN-based models, such as the YOLO series (Wang et al., 2023). The architecture of one-stage CNN-based models has undergone substantial optimization in feature extraction, bounding box fitting, and regression, apparently eliminating the anticipated advantage of two-stage CNN-based models in salmon louse detection tasks. In addition, Faster R-CNN50 outperformed Faster R-CNN101 on all metrics in both datasets and a similar trend was observed for the Sparse R-CNN 50 and Sparse R-CNN 101. This indicates that augmenting the depth and parameters of the two-stage models does not necessarily result in improved performance in the salmon lice detection task.

The YOLO series models exhibited the fastest inference speeds among all tested models, consistently exceeding 100 FPS. These models are versatile enough to be deployed on low-powered computers while enabling real-time operation. This allows various applications in the field using battery-powered computers, including sea cages and other sea regions. Sampling in cages will give information about the production of nauplii on-site and the concentration of "incoming" copepodites. Measuring upstream from farms will give the possibility of operational measures to be employed to lower infection risks when elevated concentrations of infectious copepodites are being detected. Large-scale monitoring to map a region can help shed light on production, spreading, and infection risks for farmed and wild salmonids. This knowledge may help to develop more complex and comprehensive solutions against salmon lice. Particularly, YOLOv8n stands out with an impressive inference speed of up to 416 FPS, with a model size of only 6.5 MB. This enables real-time monitoring based on low computational costs, making it exceptionally well-suited for large-scale, cost-effective deployment in offshore net cages or sea regions with fish farms. YOLOv5su stands out as the second-fastest model in terms of inference speed among all models utilized in this study. YOLOv5su achieved an F1 score that closely approached the top-performing RT-DETR, and its inference speed and model size have more significant advantages over RT-DETR. The results shown in Figure 2.6 also demonstrated that both YOLOv5su and RT-DETR can detect salmon lice well. Therefore, YOLOv5su is also a very suitable model for end-use applications. Recall, as one of the most important metrics in the salmon lice detection process, can directly demonstrate how many salmon lice have been missed for detection. All YOLO models except YOLOv3 obtain a recall higher than 0.96, which means that less than 4% of salmon lice may be missed for detection. This would seem to be acceptable for the salmon lice detection task. In addition, the YOLO series exhibits very good performance in all metrics except for YOLOv3. This indicates that the network architecture of the YOLO series can process the images effectively, extract enough features, and achieve accurate predictions. In all one-stage CNN-based models, the SSD series performed poorly in terms of all metrics on the two datasets, which is because the architecture of VGG16 (Mas-

carenhas & Agarwal, 2021) is not able to extract sufficient features from the images in the salmon lice detection task.

Transformer-based models excel in detection precision, recall, and various metrics, particularly for RT-DETR. However, they have an inference speed of less than 47 FPS and will decrease as computer power decreases, resulting in more time needed to process the videos. This somewhat renders application scenarios of RT-DETR limited compared to the YOLO series models. For example, RT-DETR is less suitable for use on low-energy computing devices and large-scale deployment.

The models show a variable performance for the detection of different stages of salmon lice (e.g., "Nauplius Live," "Nauplius Shell," and "Copepodite"). Across all models, a common observation is a relatively weaker performance in identifying "Nauplius Shell" compared to "Nauplius Live," and "Copepodite." This discrepancy may arise from the transparent nature of "Nauplius Shell" in images, which contains less morphological and color information compared to "Nauplius Live" and "Copepodite." The high performance of the models such as YOLOv5su and YOLOX on "Nauplius Live" and "Copepodite" suggests that in the collected images, living nauplii and copepodites possess distinctive features significant enough to be utilized by the majority of the machine learning models in this study. Thus, the ability of these models to distinguish between the similar shapes of Nauplius Live and Nauplius Shell on the one hand, and between Nauplius Live or Shell and Copepodite on the other hand indicates that the models may also be able to make future distinctions not only between sea louse larvae and various kinds of plankton, but also between sea louse larvae and other, perhaps rather similarly looking, fish parasite larvae.

In machine learning, the model itself has to find patterns in the data and make predictions about future data based on the patterns obtained. The models' performance is heavily influenced by the diversity and informativeness (e.g., sufficient viewpoints of salmon lice and other organisms) of the training image set. A diverse image set with sufficient information allows the model to learn essential features, ensuring accuracy during predictions. Thus, training images should closely resemble the images the model will encounter in practical applications. In this study, the images in Dataset NC are cleaner and simpler compared to real seawater rich in plankton. Therefore, although both dataset NC and real seawater images have salmon lice information, dataset NC lacks other information, such as other planktonic organisms, different salmon lice angles, etc. compared to real seawater images. This limitation restricts the features the model can learn from dataset NC, leading to some false positives and missed detection (false negatives). If real seawater images are added to the training dataset, the model fit will probably be more suitable for real seawater images and thus have a better performance. Nevertheless, preliminary application of the machine

learning-based salmon louse detection approaches appeared to have satisfactory performance in the detection of seawater. In tests for the detection of "wild" salmon lice in real seawater, a total of 39 cubic meters of seawater was pumped and filtered from a commercial fish cage, and two videos of filtered samples were captured using the data acquisition platform (Figure 2.3). The videos were analyzed manually, and 7 and 5 salmon louse nauplii were found in the two samples, respectively. The YOLOv5su and YOLOv8n trained on the NC dataset were chosen for testing the videos. Even though the two test samples had much more complex backgrounds (plankton) than the NC dataset, the models still detected all salmon louse nauplii in the two samples when the confidence was set appropriately. With the fast detection speed of the YOLO series, the entire detection takes approximately the same amount of time as the video duration. Due to the extreme abundance of plankton in seawater, some misdetection (false positives) are inevitable. However, an operator with only several hours of training can easily and quickly remove false positives during the verification process and retain the correct results. A 26-minute video takes only roughly 1-5 minutes to verify and count. A manual comparison revealed that the salmon louse nauplii detected by the model and the manual analysis were the same organisms in the video. In the future, by augmenting the model's training data with additional images, such as more images of salmon lice that also include various populations of plankton in the seawater, the model will likely have much fewer false positives when analyzing real seawater samples. This enhancement is expected to lead to a further improvement in both precision and recall. While a proper evaluation of the performance of our model for the analysis of real seawater samples containing numerous variations of diverse plankton demands the comparison of many and diverse samples, these first results confirm the potential and applicability of machine learning-based models for the detection of salmon lice in the sea.

2.4.2 Comparison with traditional methods

Traditional salmon louse detection methods include visual (light microscopy, fluorescence microscopy, and automatic fluid imaging) (Bui et al., 2021; Thompson et al., 2022) and molecular methods (droplet digital PCR (McBeath et al., 2006), quantitative fraction PCR (qfPCR) and quantitative PCR (qPCR)) (Bui et al., 2021; Truett et al., 2000). The visual methods use a microscope for salmon lice detection and the molecular techniques involve a series of complex chemical processes in the laboratory to extract DNA and calculate relevant indicators to determine the concentration of salmon lice in seawater (Krolicka et al., 2022; Bui et al., 2021). According to reports by Bui and Thompson (Bui et al., 2021; Thompson et al., 2022), they utilized the traditional methods mentioned above and achieved detection speeds ranging from approximately 1100 to 23,000 organisms (including salmon louse larvae and other

plankton) per hour. In contrast, when testing a real seawater sample in this study, estimated at 119,000 organisms (22 minutes of video), the detection speed reached 325,000 organisms per hour. In addition, the data acquisition platform designed in this study allows for a further increase of the flow rate when increasing shutter speeds in combination with brighter illumination, providing increased detection speed and enabling real-time analysis concurrently with video capture.

In addition, the precision obtained through traditional methods varied from 27.3% (qPCR) to 77.4% (light microscopy) while the recall (visual methods only) ranged between 41.6% (automatic fluid imaging) and 84.8% (light microscopy) (Bui et al., 2021). In contrast, the machine learning models tested in this study, particularly the YOLO series models, consistently achieved precision and recall rates of over 90% on the test set of Dataset NC. On the actual seawater sample, the precision obtained by YOLOv5su and YOLOv8n with manual counting reached 100% while the range of recall was between 60% and 100%. This value is expected to continue to improve as the training dataset expands in the future.

The machine learning-based object detection approach investigated in this study is an image-based method. It offers greater flexibility and stability and is much cheaper than previously reported methods (Bui et al., 2021). It is an automated salmon lice detection method where the model automatically detects and differentiates different stages of salmon louse larvae in seawater. This may greatly increase the efficiency and largely eliminate the demand for trained personnel. In this paper, even though some other planktonic contextual factors have been taken into account in our image data, our dataset still needs to be tested on rich plankton communities in real-life seawater. In the next step, we, therefore, intend to explore the challenges of lice detection in more complex seawater environments. Despite this, the model in this paper, despite being trained on a dataset with a relatively clean background, still achieved quite satisfactory results when tested provisionally on real seawater samples rich in plankton. This may indicate that the model is robust and scalable. Further work is needed to confirm this.

2.5 Conclusion

In this study, we established a system for acquiring video footage of salmon lice in seawater, generating a dataset comprising both nauplius and copepodite stages. We also introduced rapid machine learning-based salmon lice detection approaches, implementing and systematically comparing 16 object detection algorithms. The results demonstrate the effectiveness of the machine learning-based approaches in swiftly and accurately identifying salmon lice and distinguishing their main developmental stages,

particularly favoring the YOLO series models. Our research addresses the challenges associated with salmon lice detection in seawater, offering an innovative solution for a pressing issue in salmon farming (Olaussen, 2018). In future research, we will further explore the impact of factors, such as dataset partitioning strategy and number of images, on the performance of the models. Machine learning-based approaches stand out in terms of speed, cost-effectiveness, efficiency, and automation. Beyond assessing the current prevalence of salmon lice in water bodies, the widespread application enables the creation of accurate and regularly updated salmon lice maps, supporting regional planning, production strategies, and proactive responses to infection risks. By minimizing salmon lice infections in both farmed and wild salmon, machine learning-based approaches may contribute significantly to improving the welfare of these fish and promoting sustainable salmon farming practices.

Chapter 3

An image synthesis framework for enhanced salmon louse larvae (*Lepeophtheirus salmonis*) detection in complex seawater conditions

This chapter has been published as:

Zhang, C., Gansel, L. C., Bracke, M., & da Silva Torres, R. (2025). An image synthesis framework for enhanced salmon louse larvae (*Lepeophtheirus salmonis*) detection in complex seawater conditions. *Computers and Electronics in Agriculture*, 239(B), 110985. https://doi.org/10.1016/j.compag.2025.110985

Abstract

Salmon lice (*Lepeophtheirus salmonis*) infections pose significant threats to farmed and wild salmon populations, making early detection during pre-infective and infective stages essential for sustainable management. Traditional methods, such as PCR, are costly and unsuitable for large-scale deployment, while existing machine-learning approaches are limited by the lack of annotated data from complex seawater environments and are largely restricted to simplified laboratory conditions. To address these challenges, this study presents an automated synthetic data generation method based on the Segment Anything Model (SAM), specifically designed to improve the detection of multi-stage salmon louse larvae in real-world and complex seawater environments. A total of 54 orthogonal experiments were conducted to optimize the factor configuration for synthetic data generation, resulting in a high-quality dataset comprising 120,864 images. To maximize the utility of synthetic data and enhance model performance, we evaluated YOLO series models and the transformer-based RT-DETR-L model. Our experiments revealed that pretraining on synthetic data before fine-tuning (or hybrid training) consistently improved model performance across multiple evaluation metrics. The YOLOv8n's F1-score increased from 70.6% to 87.7%, a notable relative improvement of 24.22%. In real-world seawater tests, models trained with synthetic data improved recall by 11.0% to 87.0% compared to models trained exclusively with the original data and outperformed well-trained biologists, further validating the effectiveness and practicality of the synthetic approach. This study provides a scalable solution for monitoring salmon louse larvae in large-scale seawater environments, improving the welfare of farmed and wild salmon. The source code is publicly available at `https://github.com/jay-zc/synthetic-dataset` (as of July 2025).

3.1 Introduction

Norway is the world's largest producer and exporter of salmon (Food and Agriculture Organization of the United Nations (FAO), 2025; Taranger et al., 2015). In 2023, sales of salmon in Norway reached 1.5 million tons (Fiskeridirektoratet), underlining the importance of salmon farming in the country's economy (Fiskeridirektoratet, 2023). However, salmon lice (*Lepeophtheirus salmonis*), natural parasites of salmonids in seawater, are recognized as one of the major threats to salmon farming (Taranger et al., 2015; Costello, 2006), threatening both salmon production and its long-term sustainability (Torrissen et al., 2013).

The popularity of open-net farming leads to high salmon densities, creating ideal conditions for salmon lice reproduction (Costello, 2009; Torrissen et al., 2013). Excessive

salmon lice recruitment affects both farmed and wild salmon. The life cycle of salmon louse includes eight stages, beginning as non-parasitic, free-swimming nauplii (Hamre et al., 2013). In the following copepodite stage, salmon lice seek out hosts, attaching themselves to fish and feeding on their skin and mucus. This weakens the host's immune system, increasing mortality (Schram, 1993; Jeong et al., 2021; Costello, 2006). While salmon lice density in seawater is relatively low due to dispersal, high recruitment rates from intensive salmon farming can result in widespread infections (Murray & Moriarty, 2021; Morton et al., 2004). Farmed salmon, lacking the ability to escape from parasites, are particularly susceptible to salmon lice infestations (Shephard & Gargan, 2017). Wild populations near farms face similar risks as lice transmission can occur over extensive distances (Salama et al., 2011). On the Norwegian coast, a single salmon farm can contain millions of salmon and may release millions of salmon lice per day. These larvae can travel 20–200 km within 10 days, approximately the time required for them to become infectious under favourable conditions (Taranger et al., 2015; Serra-Llinares et al., 2014; Salama et al., 2011). To mitigate this, Norwegian regulations mandate maintaining lice levels below 0.5 adult females per fish, with stricter limits of 0.2 adult females during key migration weeks (Torrissen et al., 2013; Lerøy Seafood Group, 2022; Jeong et al., 2023). However, effective control remains challenging due to rapid and extensive parasite spread (Salama et al., 2011).

Accurate mapping of salmon lice distribution is essential for risk mitigation and management of both wild and farmed salmon. Monitoring salmon lice transmission dynamics supports regional planning, optimizes farm layouts (Parent et al., 2024; Adams et al., 2015; Samsing et al., 2017), and informs management decisions, such as fry release and delousing (Kragesteen et al., 2018; Jeong et al., 2021). In Norway, the "Lakseluskartet" (Salmon Lice Map) (Havforskningsinstituttet) integrates lice counts from farms, biological development models, and oceanographic data to predict lice distribution. However, current methods used in the "Lakseluskartet" primarily rely on lice counts from a limited number of farmed and wild fish, representing a secondary measure that does not accurately reflect pre-infection larval abundance in seawater (Gupta et al., 2022; Jeong & Revie, 2020). Direct detection of larvae in seawater could improve model accuracy and enable real-time monitoring, enhancing farm management and lice control strategies (Nelson et al., 2018).

Traditional methods include manual observation (Thompson et al., 2022) and molecular-based techniques (Krolicka et al., 2022; Roy & Boulding, 2024). Salmon louse larvae are small during free-swimming stages and occur at low densities in seawater, which complicates detection and often requires sampling large water volumes followed by filtration. Manual observation methods usually identify targets by light microscopy (Thompson et al., 2022). However, this method is labor-intensive and re-

quires expertise, making it unsuitable for large-scale monitoring. Molecular methods are expensive, laboratory-based, and unable to distinguish between developmental stages (Krolicka et al., 2022). Machine learning, particularly vision-based techniques, offers a promising alternative. For instance, Zhang et al. (2024a) introduced a machine vision method for detecting salmon lice larvae and distinguishing developmental stages. However, these methods relied on extensive datasets, which are challenging to generate due to low larval abundance, high densities, the presence of similarly sized planktonic organisms, and the need for manual labeling. In addition, current approaches often simplify environmental conditions, which limits their applicability in real-world aquaculture settings.

Synthetic data generation has proven effective in addressing data scarcity in various real-world scenarios (Kirillov et al., 2023; da Silva et al., 2023; Jung et al., 2024; Picon et al., 2022). Dyrstad & øye (2025) used 3D fish models and domain randomization to generate diverse training data for fish segmentation. However, the gap between synthetic and real-world data remains a challenge, particularly for small-object detection in plankton-rich seawater. The dominance of non-target organisms complicates both detection and realistic synthetic data generation. Zheng et al. (2024) observed performance gains using synthetic data for salmon louse detection, but their approach relied on clean and uniform backgrounds and was limited to the nauplius stage, thereby reducing ecological realism. Therefore, further research should address complex backgrounds and include multiple larval stages to better reflect real-world conditions.

The Segment Anything Model (SAM) (Kirillov et al., 2023) is a cutting-edge image segmentation model capable of zero-shot generalization. It enables accurate object segmentation from prompts (e.g., points, bounding boxes, or text) without additional training. Building on SAM, Osco et al. (2023) combined SAM with text prompts and one-shot training to automate object counting in remote sensing, significantly reducing manual annotation efforts. Similarly, Rayamajhi et al. (2024) applied SAM to segment dense tree canopy images captured by drones. Given its zero-shot capability and prompt-based segmentation features, SAM holds the potential for extracting information from images and leveraging it in synthetic data generation.

To address the challenges of limited access to training data and the high cost of manual annotation required for salmon louse larvae detection, we developed a SAM-based image synthesis framework for scalable, biologically realistic dataset generation. We used SAM to perform high-quality, prompt-guided instance segmentation using bounding box prompts for target organisms and random point prompts for non-target particles. Synthetic images were generated by compositing segmented instances onto background-only images, with Poisson-disk sampling used to ensure

ecologically plausible distributions. To enhance realism, we introduced a suite of physically grounded augmentations, including biologically constrained scaling and rotation, illumination modulation, motion blur, and other effects, thereby capturing the complexity of real-world marine environments. Furthermore, we systematically investigated the influence of factor configurations on the synthetic images through orthogonal experimental design, optimizing the balance between augmentation intensity and biological fidelity. Extensive experiments on real datasets and field validations show that, even with scarce real data, our data synthesis method markedly enhances detection performance, providing a cost-effective and scalable tool for salmon louse larvae surveillance and ecological management. The primary contributions of this study are as follows:

1. A SAM-based image synthesis framework is proposed, integrating dual-prompt segmentation and physically-grounded augmentations, to generate biologically realistic training data of multi-stage salmon louse larvae.

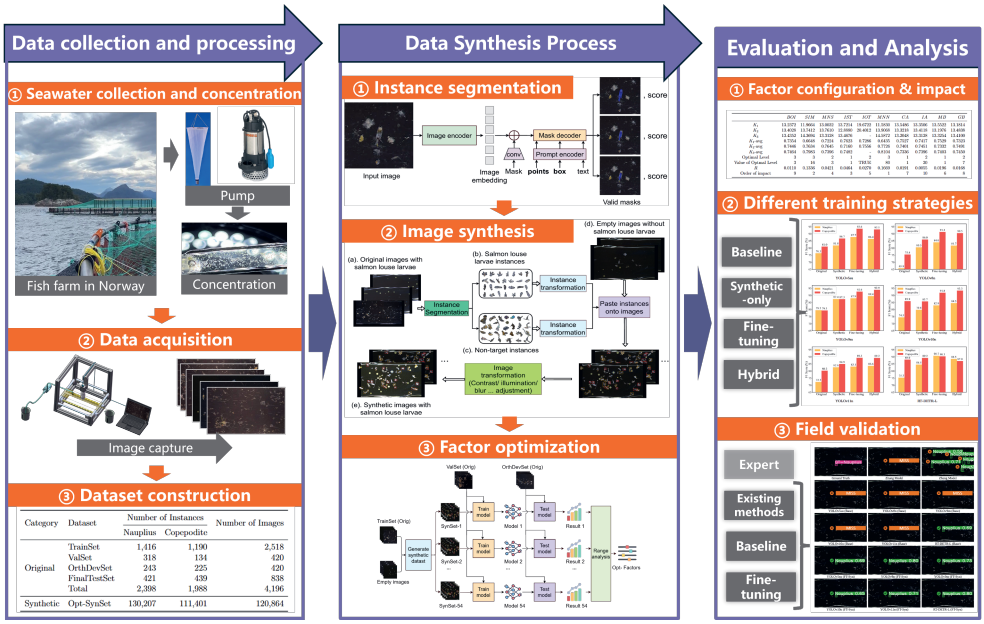
2. Extensive field validation experiments are conducted, evidencing superior detection performance on complex real-world seawater samples.

3. Different training strategies is introduced and systematically evaluated, demonstrating substantial improvements in model generalization.

4. An orthogonal experimental design is employed to examine the influence of multiple factors and identify the optimal parameter configuration.

3.2 Materials and methods

The overall workflow is illustrated in Figure 3.1. Seawater samples were collected from a salmon farm in Norway, and original images were obtained after filtration and concentration. Using SAM, instances of salmon louse nauplii, copepodites, and non-target organisms were segmented and composed onto new backgrounds to generate synthetic images reflecting complex marine environments. Ten key factors influencing the quality of synthetic images were optimized through an orthogonal experimental design. Based on the optimal configuration, a synthetic dataset was constructed. Various training strategies were systematically evaluated, and the resulting model was further validated using real-world seawater samples.



Category	Dataset	Number of Instances		Number of Images
		Nauplius	Copepodite	
Original	TrainSet	1,416	1,190	2,518
	ValSet	318	134	420
	OrthDevSet	243	225	420
	FinalTestSet	421	439	838
	Total	2,398	1,988	4,196
Synthetic	Opt-SynSet	130,207	111,401	120,864

Figure 3.1: Overview of the proposed synthetic data generation and model evaluation workflow.

3.2.1 Data preparation

3.2.1.1 Data collection

To enable rapid collection of salmon louse images, we used a custom data acquisition device (Figure 3.2) that allows seawater to flow through a glass channel, with (videos) images captured using a high-magnification microscope camera positioned above. Images were captured at a resolution of 1920×1080 pixels (5 µm per pixel). The system supports a steady throughput of approximately 500 ml per hour. Seawater samples were collected in June 2024 at Gudmundset, locality number 12265, near Ålesund, Norway.

Sample preparation followed four main steps: (1) Seawater was pumped using a submersible pump (13 cubic meters per hour) fitted with a fine mesh filter (120 µm) to retain small particles, including salmon louse larvae. (2) After a collection interval, the filter was rinsed with clean seawater to recover retained particles, producing a wash solution. (3) The solution was further filtered through a 500 µm mesh to exclude larger debris (any particles larger than 0.5 mm), enriching the sample with smaller organisms. Some laboratory-hatched salmon louse nauplii and copepodites were added to increase louse density. (4) Videos of the final filtered solution were captured using the data acquisition device. Images were subsequently extracted from the videos frame by frame. An example image of a concentrated sample is shown in

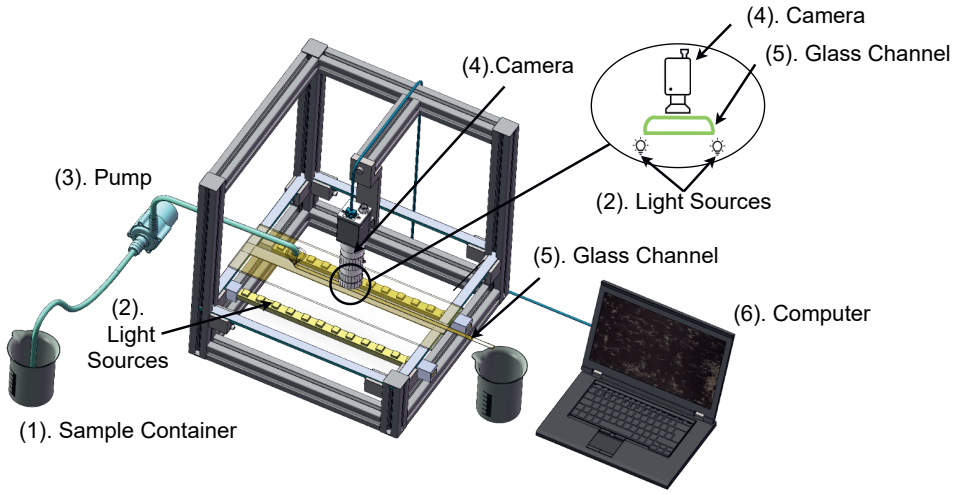


Figure 3.2: Data acquisition device. A small pump (3) transfers the concentrated seawater sample from the sample container (1) into the glass channel (5). A camera (4) with a macro lens is positioned above the glass channel (5), while a light source (2) is placed below. As the seawater flows through, the camera (4) continuously captures images or videos of the sample.

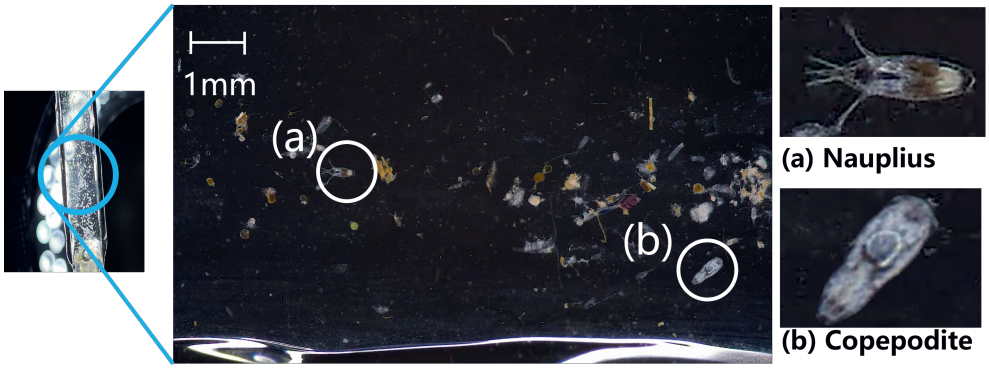


Figure 3.3: An example image of concentrated seawater sample containing salmon louse nauplius (a) and copepodite (b).

Figure 3.3.

3.2.1.2 Dataset construction

This study focused on detecting the nauplius and copepodite stages of salmon louse larvae. Concentrated seawater samples often contain diverse plankton, some of which closely resemble salmon lice, making precise annotation essential. All annotations were performed by trained marine experts using Visual Object Tagging Tool

Table 3.1: Overview of original and synthetic datasets used for salmon louse larvae detection.

Category	Dataset	Number of Instances		Number of Images
		Nauplius	Copepodite	
Original	TrainSet	1,416	1,190	2,518
	ValSet	318	134	420
	OrthDevSet	243	225	420
	FinalTestSet	421	439	838
	Total	2,398	1,988	4,196
Synthetic	Opt-SynSet	130,207	111,401	120,864

Note: 'Original' datasets refer to the datasets manually collected and annotated from real seawater samples. 'TrainSet' refers to the training set. 'ValSet' indicates validation set. 'OrthDevSet' denotes the orthogonal development set. 'FinalTestSet' refers to the final test set. 'Opt-SynSet' indicates the dataset automatically generated using the optimal factor configuration presented in Section 3.3.1. 'FinalTestSet' refers to the final test set.

(VOTT) (Microsoft, 2020), a platform designed for video annotation. Objects were labeled with bounding boxes as either "Nauplius" or "Copepodite," while the remaining regions were classified as background. To ensure annotation consistency across different annotators, all experts underwent a standardized training process to recognize the nauplius and copepodite stages of salmon lice accurately. Following the training, each annotator was required to label a small, curated subset of data. These initial annotations were compared and evaluated to establish a consistent labeling standard and resolve any ambiguities. Once the full dataset was annotated, a final quality control step was conducted by a designated expert, who reviewed all annotations and removed inconsistent or incorrect labels.

As shown in Table 3.1, the original dataset contained 4196 images and was randomly divided into four subsets following a ratio of 6:1:1:2. The training set (TrainSet) was used for model fitting, while the validation set (ValSet) was used to monitor the training process and select the best-performing model across all epochs. The orthogonal development set (OrthDevSet) was utilized to evaluate models trained on synthetic datasets generated with different factor configurations. The final test set (FinalTestSet) was used to compare the performance of models trained on the optimal synthetic dataset (Opt-SynSet) and the original TrainSet. The optimal factor configuration for the synthetic data process was determined by maximizing performance on the OrthDevSet.

3.2.2 Data synthesis methods

The overall workflow for generating synthetic images is shown in Figure 3.4. The process employs a segmenting and pasting paradigm. Specifically, instances of "Nauplius," "Copepodite," and non-target particles were first segmented from original images $\mathcal{I}_{\text{original}}$. These instances were transformed through transformations (e.g., rotation and scaling) and composited onto background-only images $\mathcal{B}$. To increase variability and realism, the resulting composites were further enhanced with brightness modulation and noise injection, yielding the final synthetic images $\mathcal{I}_{\text{synthetic}}$.

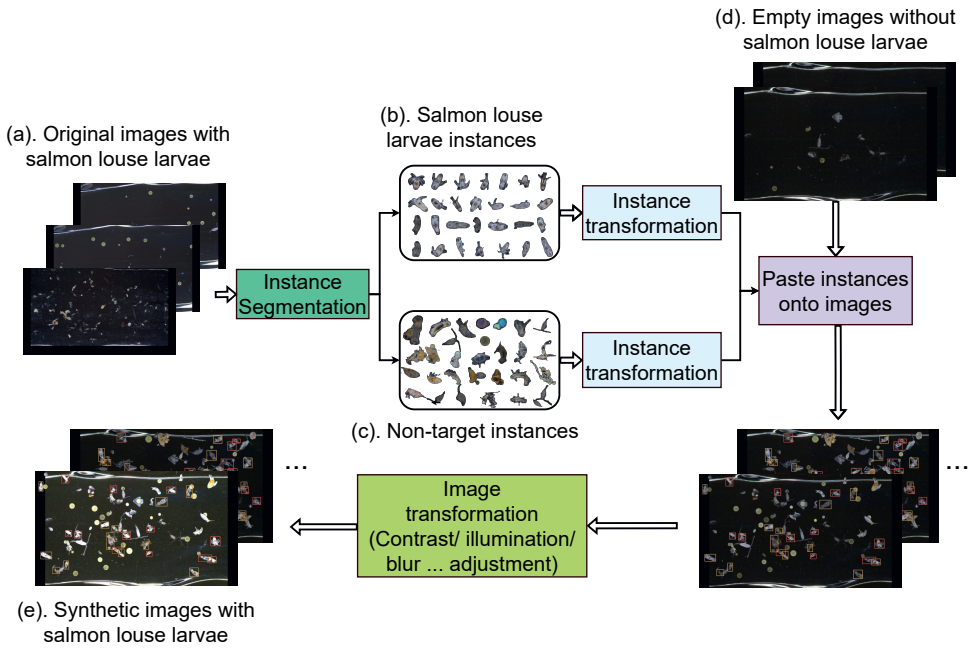


Figure 3.4: Flowchart of the synthetic image generation. Original images containing instances of salmon lice (b) and non-target particles (c) were utilized as sources to generate synthetic images (e).

3.2.2.1 Instance segmentation

We employed SAM (Kirillov et al., 2023) for efficient extraction of instance of salmon louse larvae and non-target particles. SAM accepts various types of prompts, including points, bounding boxes, masks, and text, to generate high-quality segmentation masks. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ denote the set of n prompts (e.g., bounding boxes or points) for SAM. For each prompt p_i, SAM generates a set of masks $\mathcal{M}_i = \{m_{i1}, m_{i2}, m_{i3}\}$ with corresponding confidence scores $\{c_{i1}, c_{i2}, c_{i3}\}$. The final segmentation mask m_i^* is selected as: $m_i^* = \text{argmax}_{m_{ij} \in \mathcal{M}_i} c_{ij}$. The set of segmented

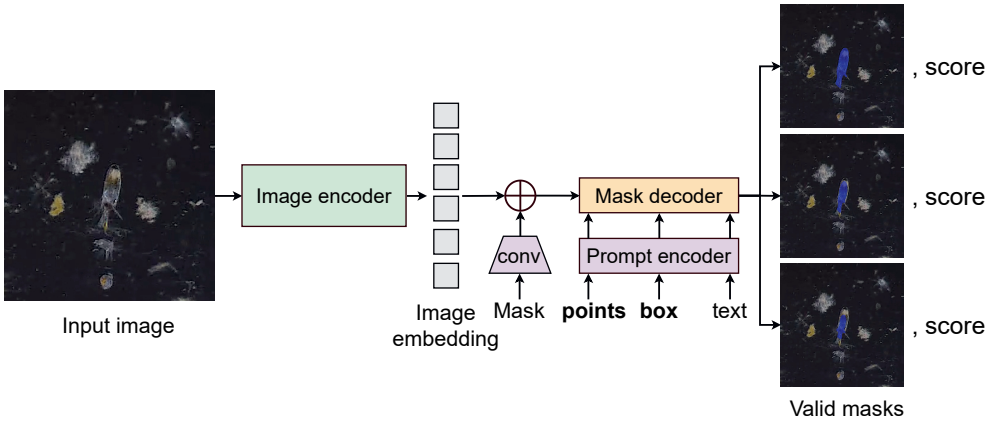


Figure 3.5: Segmentation of salmon louse larva using SAM. Labeled bounding boxes were used as prompts to generate segmentation masks for salmon louse larva instances automatically. For non-target instances, randomly sampled point coordinates served as prompts for mask generation.

instances was extracted from $\mathcal{I}_{\text{original}}$ using the selected masks m_i^*.

As shown in Figure 3.5, labeled bounding boxes were used as prompts for SAM to generate segmentation masks for the target organisms ("Nauplius" and "Copepodite"). For non-target particles, random point coordinates were generated within the images as prompts due to the absence of annotations. Non-target particles were also obtained from background-only images $\mathcal{B}$ (without salmon lice).

Extracted instances were saved as transparent background images (Figure 3.6). To ensure the integrity of the segmented salmon lice, each segmented instance was checked, and those with less than 50% completeness were removed. Segmentation was performed only on the original TrainSet, resulting in a salmon louse set $\mathcal{S}$ containing 2606 instances and a non-target set $\mathcal{N}$ containing 2309 instances.

3.2.2.2 Object paste and image transformation

A specified number of salmon louse instances $s_i \in \mathcal{S}$ and non-target instances $n_j \in \mathcal{N}$ were selected and pasted onto background-only images $b \in \mathcal{B}$, with their spatial distribution following a Poisson-disk sampling distribution to prevent overlapping:

$$\forall (u, v) \in \mathcal{U} \times \mathcal{U},\ u \neq v,\ \|p(u) - p(v)\|_2 > r_{\min}, \tag{3.1}$$

where $\mathcal{U} = \mathcal{S} \cup \mathcal{N}$ is the set of all instances, $p(u)$ and $p(v)$ are centroid coordinates, $\|p(u) - p(v)\|_2$ is the Euclidean distance between the centroids of u and v, $r_{\min}$ is the minimum separation distance.



Figure 3.6: Transparent background images of salmon louse larva instances and non-target instances.

The annotation of newly generated images was derived based on the class, size, and position of the pasted instances. Image augmentation techniques, such as changing the orientation of the instances and adjusting contrast and brightness, were applied to enhance image diversity:

$$I_{\text{synthetic}} = T_{\text{transform}}\left(b \oplus \left(\{s_1, s_2, \ldots, s_{N_{\text{SL}}}\} \cup \{n_1, n_2, \ldots, n_{N_{\text{NT}}}\}\right)\right), \tag{3.2}$$

where $I_{\text{synthetic}}$ is the generated synthetic image, $b \in \mathcal{B}$ is the background image, $s_i \in \mathcal{S}$ is sampled from a fixed set of salmon louse larva instances, $n_j \in \mathcal{N}$ was sampled from a fixed set of non-target instances, N_{SL} and N_{NT} are the number of salmon louse and non-target instances pasted onto $\mathcal{B}$, respectively, $\oplus$ denotes the pasting operation, and $T_{\text{transform}}$ is a composition of transformations:

$$T_{\text{transform}} = IST \circ IOT \circ CA \circ IA \circ MB \circ GB \tag{3.3}$$

where $\circ$ denotes function composition, and the individual transformations are shown in Table 3.2.

The synthetic dataset generation process was further controlled by four factors (Table 3.2) to ensure data quality and biological realism. The Background-only Image Ratio (BOI) was the proportion of background images relative to the original dataset, balancing environmental diversity with computational efficiency. The Synthetic Image Multiplier (SIM) determined the number of synthetic images generated per background image. Biological fidelity was enforced through the Maximum Salmon Lice per Image (MNS), which constrained the density of salmon louse larvae based on observed distributions. Environmental complexity was introduced via the Maximum

Table 3.2: Controllable factors and value ranges in synthetic image generation of salmon lice for enhanced dataset diversity and quality.

Factor	Full Name	Explanation	Value Range
IST	Instance Size Transformation	Scaling factor for instance	$IST \in [0.8, 1.2]$
IOT	Instance Orientation Transformation	Rotates instances in random orientations	$IOT \in \{\text{False}, \text{True}\}$
CA	Contrast Adjustment	Modifies image contrast through linear scaling	$CA \in [1, 3]$
IA	Illumination Adjustment	Alters brightness levels through additive bias	$IA \in [0, 40]$
MB	Motion Blur	Simulating directional blur effects from relative motion	$MB \in [1, 7]$
GB	Gaussian Blur	Applying isotropic smoothing for defocus effects	$GB \in [1, 15]$
BOI	Background-Only Image Ratio	Proportion between background images and original dataset	$BOI \in [1, 3]$
SIM	Synthetic Image Multiplier	Expansion for synthetic image generation	$SIM \in [1, 16]$
MNS	Maximum Target Instances	Maximum number of salmon louse larva instances introduced per image	$N_{SL} \in [1, MNS],\ MNS \in [1, 5]$
MNN	Maximum Non-target Instances	Maximum number of non-target instances introduced per image	$N_{NT} \in [0, MNN],\ MNN \in [0, 80]$

Non-target Instances (MNN), controlling the number of background objects to enhance realism. An example comparison between the synthetic image and the original image is shown in Figure 3.7.

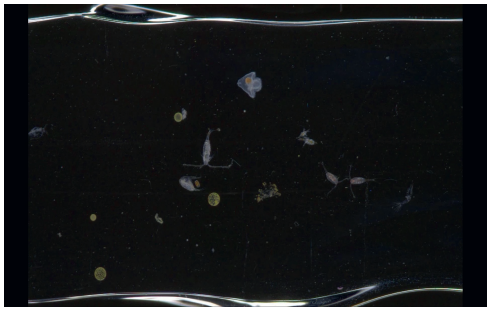
3.2.3 Experimental setup

3.2.3.1 Orthogonal experiment

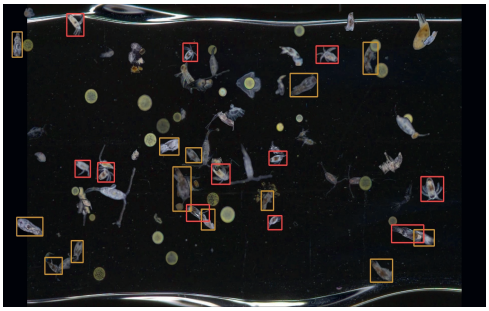
We conducted experiments to evaluate the synthetic method under limited data, and to identify the optimal factor configuration of the synthesis factors. As illustrated in Figure 3.8, we employed an orthogonal design (Wu, 2013; Franek & Jiang, 2013) to systematically explore the effects of different factors. Each factor was assigned discrete levels based on its characteristics (Table 3.3). Using the L54 ($2^1 3^9$) orthogonal array, we generated 54 distinct combinations of factor levels (Table B.1), resulting in corresponding synthetic datasets. Models were trained on these datasets and evaluated on OrthDevSet. We applied range analysis to quantify each factor's impact on dataset quality. The overall F1-score (Eq. 3.6) served as the evaluation metric to identify the optimal level for each factor. Finally, an optimal synthetic dataset was generated using the best-performing combination.

3.2.3.2 Model training strategies

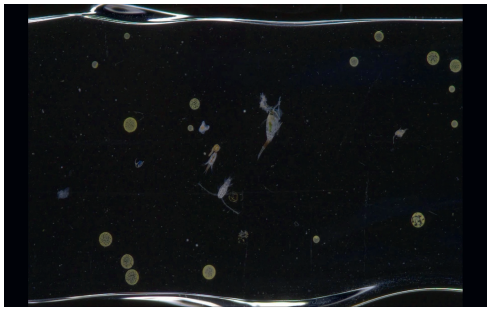
Building on the optimized factor configuration identified in Section 3.3.1, we generated an optimal synthetic dataset (Opt-SynSet) using the proposed synthesis method (Table 3.1). To evaluate its effectiveness, we selected six models, including five CNN-based architectures (YOLOv5su, YOLOv8n, YOLOv9m, YOLOv10n, and



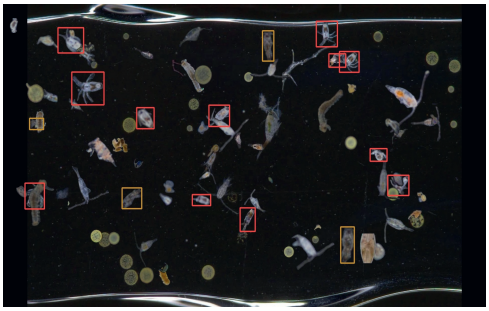
(a) Background-only image without salmon lice.



(b) Synthetic image generated from (a).



(c) Background-only image without salmon lice.



(d) Synthetic image generated from (c).

Figure 3.7: Example comparison of original image (a, c) and corresponding synthetic image (b, d). In the images, red bounding boxes indicate "Nauplius," while brown bounding boxes denote "Copepodite." Image transformations, including brightness and blur adjustments, were randomly applied to enhance data diversity.

Table 3.3: Factor levels used in the orthogonal experiment.

	Factor	*BOI*	*SIM*	*MNS*	*IST*	*IOT*	*MNN*	*CA*	*IA*	*MB*	*GB*
	1	1	1	1	1	True	0	1	0	1	1
Level	2	2	8	3	0.9–1.1	False	40	2	20	3	7
	3	3	16	5	0.8–1.2	—	80	3	40	7	15

Note: In the table, *BOI* stands for Proportion of Background-only Images. *SIM* stands for Synthetic Image Multiplier. *MNS* stands for Maximum Number of Salmon Lice Introduced per Image. *MNN* stands for Maximum Number of Non-target Instances Introduced per Image. *IST* stands for Instance Size Transformation. *IOT* stands for Instance Orientation Transformation. *CA* stands for Contrast Adjustment. *IA* stands for Illumination Adjustment. *MB* stands for Motion Blur. and *GB* stands for Gaussian Blur.

YOLOv11n) and one Transformer-based RT-DETR-L. While directly training models on the synthetic dataset is straightforward, this approach may not fully leverage the potential advantages of synthetic data. To explore the impact of different ways of incorporating synthetic data, we designed four distinct model training strategies,

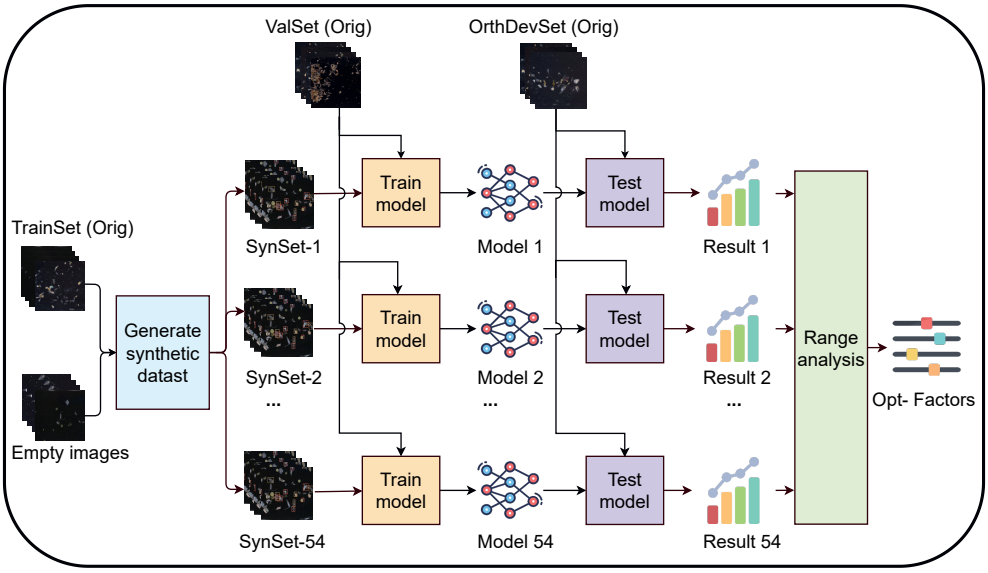


Figure 3.8: Optimization of synthetic data generation factors using orthogonal experiments. 'Orig' datasets refer to datasets manually collected and annotated from real seawater samples. TrainSet represents the original training set, and ValSet represents the original validation set. SynSet- denotes different synthetic datasets generated during the experiment. The optimal factor combination (Opt-Factors) is selected based on the performance on the orthogonal development set (OrthDevSet).

described as follows: (1) Baseline Training: As shown in Figure 3.9a, the model was trained only on the original TrainSet and evaluated on FinalTestSet. (2) Synthetic-Only Training: As shown in Figure 3.9b, the model was trained solely on Opt-SynSet and subsequently evaluated on FinalTestSet. (3) Fine-Tuning Training: As shown in Figure 3.9c, the model was initially trained on Opt-SynSet and subsequently fine-tuned on the original TrainSet. Evaluation was conducted on the FinalTestSet. (4) Hybrid Training: As presented in Figure 3.9d, the original training set and Opt-SynSet were combined to form a hybrid dataset (HybridSet) for model training, with subsequent evaluation on the FinalTestSet.

3.2.3.3 Field validation

To evaluate the generalizability of our models and the effectiveness of the synthetic data approach, we performed field validation using two seawater samples collected from Norwegian salmon farming regions in August 2024 and February 2025 (Table 3.4). These samples were collected more than two months after the original datasets (Table 3.1) and from different locations, introducing both temporal and spatial domain shifts. Both samples underwent identical filtration and preprocessing procedures

Table 3.4: Characteristics of seawater samples used for field validation across different temporal and spatial domains.

Attribute	Sample 1	Sample 2
Collection time	August 2024	February 2025
Temporal distance from TrainSet	∼2 months later	∼8 months later
Spatial distribution shift	Yes	Yes
Larvae augmentation	No	Lab-hatched nauplius and copepodite larvae
Video duration	40 minutes	14 minutes
Frame count	60,000	21,000
Plankton diversity	High	High

as described in Section 3.2.1.1, but differed in biological composition and in the labeling protocol.

Sample 1 (Natural Sample): This unaltered seawater sample contained a natural mix of organisms. To establish ground truth, a marine biology expert performed manual detection from the video, with timestamps recorded for each identified salmon louse larva. Model predictions were generated frame-by-frame, and predicted bounding boxes were extracted, filtered, and timestamped. The expert annotations and model predictions were then cross-validated by marine biologists using the original videos. While not exhaustive, this synthesis of basic facts based on video validation provides a reliable and sufficiently accurate reference point (Fernandez-Gonzalez et al., 2022).

Sample 2 (Spiked Sample): This sample initially tested negative for salmon louse larvae by both expert and model inspections. Subsequently, a known quantity of laboratory-hatched salmon louse larvae (36 Nauplii and 10 Copepodites) was introduced to simulate a higher-density infestation. The ground truth was thus defined by the number of introduced larvae, eliminating the need for further manual verification.

Both samples exhibited much higher biological diversity and particle density than the original TrainSet, with Sample 1 presenting particularly complex backgrounds. In total, 40 minutes of video footage from Sample 1 and 14 minutes from Sample 2 were analyzed.

We conducted comparative evaluations using expert assessments and three model categories: (1) previously published models (Zhang et al., 2024a; Zheng et al., 2024), (2) models trained solely on the original TrainSet, and (3) our synthetic data-enhanced models with a fine-tuning strategy. Expert detection results were used only for Sample 1, since Sample 2 had a predefined ground truth.

3.2.3.4 Experimental environment and evaluation metrics

Zhang et al. (2024a) evaluated 16 object detection models for salmon louse detection, highlighting the superior speed and accuracy of the YOLO series and RT-DETR-L (Zhao et al., 2024b). Based on these findings, we initially employed the YOLOv11 model (Khanam & Hussain, 2024) to identify the optimal factor configuration through orthogonal experiments. After generating the optimized synthetic dataset (Opt-SynSet), we extended our evaluation to additional architectures, including RT-DETR-L (Zhao et al., 2024b) and YOLO models (Wang et al., 2024c,a; Khanam & Hussain, 2024), to assess the generalizability of synthetic data benefits across model types.

All models were implemented in PyTorch and executed on a Linux-based Pop!_OS operating system, equipped with an A5500 GPU with 16GB of memory, an Intel i9-13950HX CPU, and 32GB of RAM. All models were pre-trained on the COCO dataset (Lin et al., 2014). For standard dataset evaluations, we used precision, recall, F1-score, and mAP_{50} as key metrics:

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.5}$$

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

$$AP = \int_0^1 Precision(Recall) dRecall \tag{3.7}$$

$$mAP = \frac{\sum_1^N AP}{N} \tag{3.8}$$

where True Positive (TP) represents the number of positive instances of larvae correctly identified by the model. False Positive (FP) represents the number of instances of larvae incorrectly identified as positive by the model. False Negative (FN) represents the number of positive instances incorrectly identified as negative by the model.

In the orthogonal experiment, the overall F1-score was used to quantify the influence of each factor on the synthetic dataset quality. Range analysis was conducted based

on the sum of F1-scores (K_i) across all experiments at each factor level. The average score per level (K_i-avg) was computed by dividing K_i by the number of relevant experiments. The range (R) was then calculated as the difference between the highest and the lowest K_i-avg values for each factor. A larger R indicates greater influence of that factor on model performance.

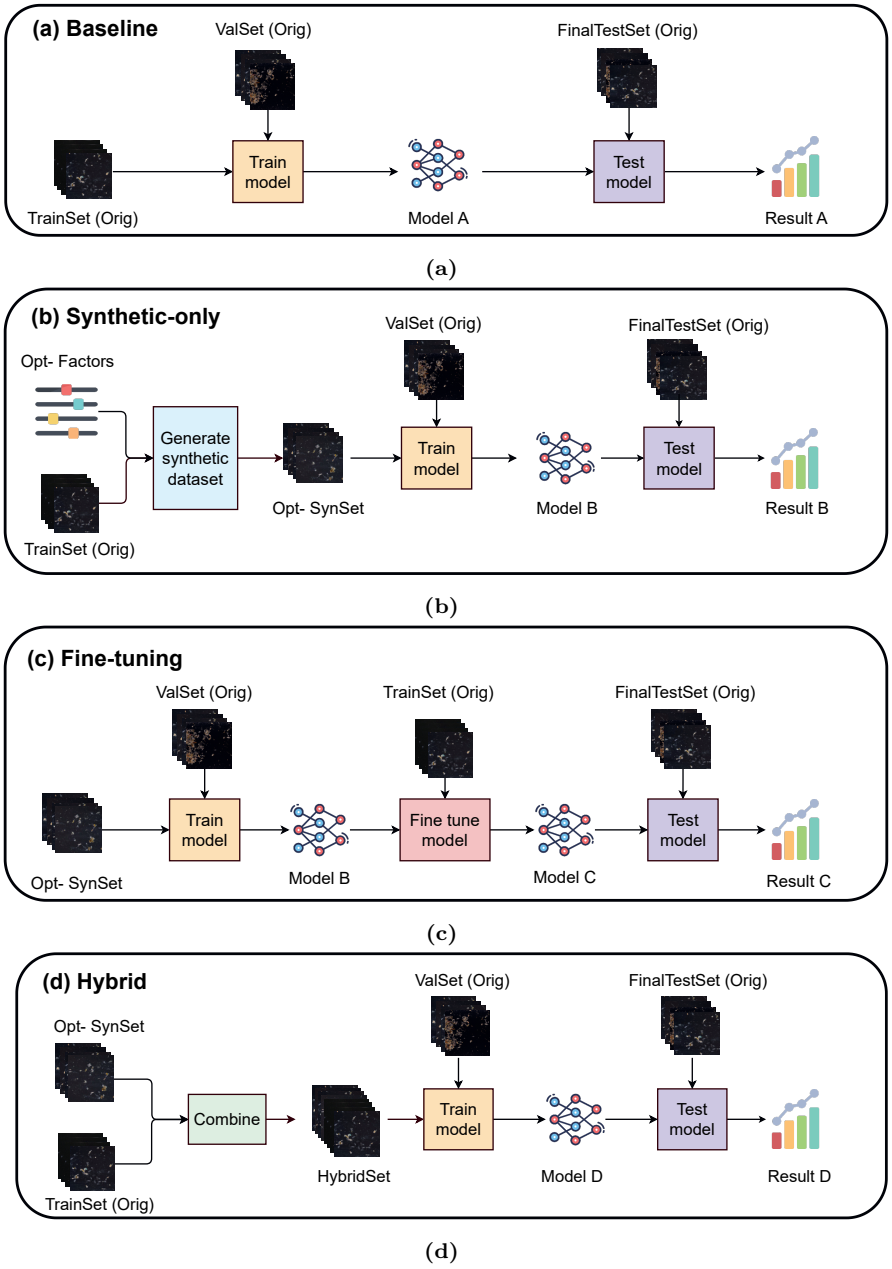


Figure 3.9: Training model using different training strategies. TrainSet represents the original training set, and ValSet represents the original validation set. Opt-Factors indicates the optimal factor configuration obtained for the synthetic data generation. Opt-SynSet refers to the optimal synthetic dataset generated using the optimal factor configuration. FinalTestSet represents the final test set. HybridSet is the combined training dataset consisting of Opt-SynSet and TrainSet.

3.3 Results

3.3.1 Orthogonal experiment

The YOLOv11n model was trained on 54 synthetic datasets and evaluated on OrthDevSet. Appendix Table B.1 presented the detection results for each individual orthogonal combination of factors. Based on these results, Table 3.5 summarized the computed k-values and ranges, quantifying the relative influence of each factor. The orthogonal analysis revealed substantial differences in factor impact, ranked as follows: $MNN > SIM > IST > MNS > IOT > MB > CA > GB > BOI > IA$.

Among these, MNN exerted the most substantial influence, with a range (R) of 0.1669, substantially higher than other factors. The K-level averages showed that Level 3 (K_3-avg = 0.8104) outperformed Level 1 (0.6435) and Level 2 (0.7726), confirming that setting $MNN = 80$ yields optimal results. The performance of the model improved as the value of MNN increases, indicating that a higher number of non-target instances enhances training efficacy. The SIM ranked second ($R = 0.1336$), exhibiting a similar positive correlation with performance, where larger SIM values, corresponding to increased synthetic data volume, lead to improved model performance. The optimal value for the IST was 1 (Level 1), suggesting instance resizing during synthesis may hinder training. The optimal value of MNS was 3 (Level 2), which implied that excessively increasing the number of target instances did not necessarily improve dataset quality. In contrast, the BOI and IA exhibited minimal influence, with ranges of 0.0110 and 0.0055, respectively. The final optimal values for each factor were determined as follows: $MNN = 80$, $SIM = 16$, $IST = 1$, $MNS = 3$, $IOT = True$, $MB = 1$, $CA = 1$, $GB = 7$, $BOI = 3$, $IA = 20$. Based on the optimal values for each factor, an optimal synthetic dataset containing 120,864 images was generated (Table 3.1).

3.3.2 Model training strategies

3.3.2.1 Synthetic-Only training

The experimental results (Table 3.6) showed consistent performance gains across all evaluation metrics when models were trained on the Opt-SynSet (Figure 3.9b) compared to the TrainSet (Figure 3.9a). Both CNN- and Transformer-based architectures exhibited improvements ranging from 1.43% to 13.85% in precision, 3.7% to 21.92% in recall, 1.18% to 13.89% in mAP_{50}, and 3.12% to 17.99% in F1-score.

An inverse relationship was observed between baseline performance and the benefits of synthetic data. Models with lower initial performance on the original training data showed the most substantial gains. For instance, YOLOv8n, which had the

Table 3.5: Range analysis of 54 orthogonal experiments based on overall F1-scores: YOLOv11n on OrthDevSet.

	BOI	*SIM*	*MNS*	*IST*	*IOT*	*MNN*	*CA*	*IA*	*MB*	*GB*
K_1	13.2372	11.9664	13.0032	13.7214	19.6722	11.5830	13.5486	13.3506	13.5522	13.1814
K_2	13.4028	13.7412	13.7610	12.8880	20.4012	13.9068	13.3218	13.4118	13.1976	13.4838
K_3	13.4352	14.3694	13.3128	13.4676	-	14.5872	13.2048	13.3128	13.3254	13.4100
K_1-avg	0.7354	0.6648	0.7224	0.7623	0.7286	0.6435	0.7527	0.7417	0.7529	0.7323
K_2-avg	0.7446	0.7634	0.7645	0.7160	0.7556	0.7726	0.7401	0.7451	0.7332	0.7491
K_3-avg	0.7464	0.7983	0.7396	0.7482	-	0.8104	0.7336	0.7396	0.7403	0.7450
Optimal Level	3	3	2	1	2	3	1	2	1	2
Value of Optimal Level	3	16	3	1	TRUE	80	1	20	1	7
R	0.0110	0.1336	0.0421	0.0464	0.0270	0.1669	0.0191	0.0055	0.0196	0.0168
Order of impact	9	2	4	3	5	1	7	10	6	8

Note: In the table, the abbreviations are defined as follows: *BOI* stands for Proportion of Background-only Images, *SIM* stands for Synthetic Image Multiplier, *MNS* stands for Maximum Number of Salmon Lice Introduced per Image, *MNN* stands for Maximum Number of Non-target Instances Introduced per Image, *IST* stands for Instance Size Transformation, *IOT* stands for Instance Orientation Transformation, *CA* stands for Contrast Adjustment, *IA* stands for Illumination Adjustment, *MB* stands for Motion Blur, and *GB* stands for Gaussian Blur.

lowest baseline F1-score (70.6%), improved to 83.3% after using synthetic data (+12.7 percentage points), matching the performance of YOLOv10n (82.7%) and YOLOv11n (83.7%).

Model size did not consistently correlate with performance when using the original TrainSet. Larger models did not systematically outperform smaller ones in baseline evaluations but benefited more from synthetic data augmentation. For example, YOLOv9m and YOLOv10n performed similarly on original data (79.4% vs. 80.2% F1-score). However, after synthetic training, YOLOv9m improved to 87.3% (+7.9 percentage points), surpassing YOLOv10n (82.7%). This suggested that larger architectures may better exploit synthetic data.

3.3.2.2 Fine-Tuning training

As shown in Table 3.6, the fine-tuning approach involving initial training on the Opt-SynSet followed by fine-tuning with the original TrainSet (Figure 3.9c) consistently outperformed both baseline approaches. Compared to models trained solely on original data, the fine-tuned models achieved substantial improvements ranging from 6.71% to 18.24% in precision, 8.15% to 30.03% in recall, 6.04% to 21.50% in mAP_{50}, and 8.85% to 24.22% in F1-score. When compared to models trained exclusively on the Opt-SynSet, the fine-tuned models showed additional gains of 2.70% to 5.76% in precision, 3.15% to 10.29% in recall, 2.14% to 6.68% in mAP_{50}, and 2.99% to 7.47% in F1-score. These results indicate that synthetic pretraining and subsequent fine-tuning on original data provide complementary benefits, yielding superior performance relative to either approach alone.

Fine-tuning also reduced performance differences across models. The F1-score of



Figure 3.10: Comparison of detection results between models trained solely on the original TrainSet (Base) and models pre-trained on the Opt-SynSet and fine-tuned on the original TrainSet (FT-SYT). ✔ indicates correct detections, while ✘ denotes missed or incorrect detections. Among all base models, only YOLOv9m and RT-DETR-L correctly detected the salmon louse nauplius, while other models exhibited false negatives or false positives (✘). In contrast, all FT-SYT models accurately detected the salmon louse nauplius (✔).

all fine-tuned models converged within a narrow range of 86.2% to 90.6%. Notably, larger models achieved comparable F1-scores, such as YOLOv5su (90.6%), YOLOv9m (90.3%), and RT-DETR-L (90.5%), suggesting that fine-tuning minimized the impact of parameter differences on final performance. As shown in Figure 3.10, models trained exclusively on the original TrainSet often exhibited missed detections, with the exception of RT-DETR-L and YOLOv9m. In contrast, models incorporating a fine-tuning strategy consistently detected all target instances.

Table 3.6: Detection performance of YOLO and RT-DETR models on FinalTestSet across four training strategies: Original, Synthetic, Fine-tuned, and Hybrid.

Model	Parameters	GFLOPS	Exp	Precision(%)	Recall(%)	mAP_{50}(%)	F1-score(%)
YOLOv5su	7,826,326	19	Original	79.5	78.0	79.4	78.6
			Synthetic	87.1 (+7.6)	81.6 (+3.6)	86.7 (+7.3)	84.3 (+5.7)
			Fine-tuning	91.3 (+11.8)	90.0 (+12)	92.0 (+12.6)	90.6 (+12)
			Hybrid	91.2 (+11.7)	88.3 (+10.3)	92.4 (+13.0)	89.7 (+11.0)
YOLOv8n	2,690,598	6.9	Original	75.1	66.6	74.9	70.6
			Synthetic	85.5 (+10.4)	81.2 (+14.6)	85.3 (+10.4)	83.3 (+12.7)
			Fine-tuning	88.8 (+13.7)	86.6 (+20)	91.0 (+16.1)	87.7 (+17.1)
			Hybrid	86.9 (+11.8)	85.3 (+18.7)	88.7 (+13.8)	86.1 (+15.5)
YOLOv9m	16,723,566	61.1	Original	86.4	74.4	81.2	79.4
			Synthetic	89.0 (+2.6)	85.7 (+11.3)	88.9 (+7.7)	87.3 (+7.9)
			Fine-tuning	92.2 (+5.8)	88.4 (+14)	90.8 (+9.6)	90.3 (+10.9)
			Hybrid	93.4 (+7)	**90.1 (+15.7)**	**92.8 (+11.6)**	**91.7 (+12.3)**
YOLOv10n	2,707,820	8.4	Original	83.9	76.7	81.8	80.2
			Synthetic	85.1 (+1.2)	80.5 (+3.8)	85.2 (+3.4)	82.7 (+2.5)
			Fine-tuning	90.0 (+6.1)	85.0 (+8.3)	89.1 (+7.3)	87.3 (+7.1)
			Hybrid	89.1 (+5.2)	88.7 (+12)	92.5 (+10.7)	88.9 (+8.7)
YOLOv11n	2,590,230	6.4	Original	77.5	75.3	80.0	76.2
			Synthetic	85.3(+7.8)	82.3 (+7)	86.0 (+6)	83.7 (+7.5)
			Fine-tuning	87.6 (+10.1)	84.9 (+9.6)	91.4 (+11.4)	86.2 (+10)
			Hybrid	87.5 (+10)	85.4 (+10.1)	90.1 (+10.1)	86.4 (+10.2)
RT-DETR-L	32,810,186	108	Original	82.5	81.0	84.5	81.7
			Synthetic	89.9 (+7.4)	84.0 (+3)	85.5 (+1)	86.9 (+5.2)
			Fine-tuning	**93.7 (+11.2)**	87.6 (+6.6)	89.6 (+5.1)	90.5 (+8.8)
			Hybrid	91.5 (+9.0)	84.4 (+3.4)	85.0 (+0.5)	87.8 (+6.1)

3.3.2.3 Hybrid training

As shown in Table 3.6, models trained on the HybridSet (Figure 3.9d) outperformed those trained solely on the Opt-SynSet or original TrainSet. Compared to models trained solely on original data, the HybridSet training led to gains ranging from 6.20% to 15.71% in precision, 4.20% to 28.08% in recall, 0.59% to 18.42% in mAP_{50}, and 7.47% to 21.95% in F1-score. Over Opt-SynSet trained models, HybridSet training further improved performance, with the precision increased by 1.64% to 4.94%, recall by 0.48% to 10.19%, mAP_{50} by -0.58% to 8.57%, and F1-score by 1.04% to 7.50%. These results confirmed the complementary benefits of combining synthetic and original data.

Compared to fine-tuned models, HybridSet training produced mixed results. YOLOv8n and RT-DETR-L showed slight F1-score declines from 87.66% to 86.08% and 90.47% to 87.76%, respectively. In contrast, YOLOv9m and YOLOv10n improved from 90.26% to 91.65% and 87.35% to 88.93%, respectively.

3.3.2.4 Life stages detection under different training strategies

Figure 3.11 presents the detection results for "Nauplius" and "Copepodite" larvae under various training strategies. Models trained with synthetic data consistently outperformed those trained solely on the original TrainSet across both "Nauplius" and "Copepodite." Fine-tuning yielded optimal performance for YOLOv5su, YOLOv8n, and RT-DETR-L, while YOLOv9m, YOLOv10n, and YOLOv11n performed best with the hybrid training strategy.

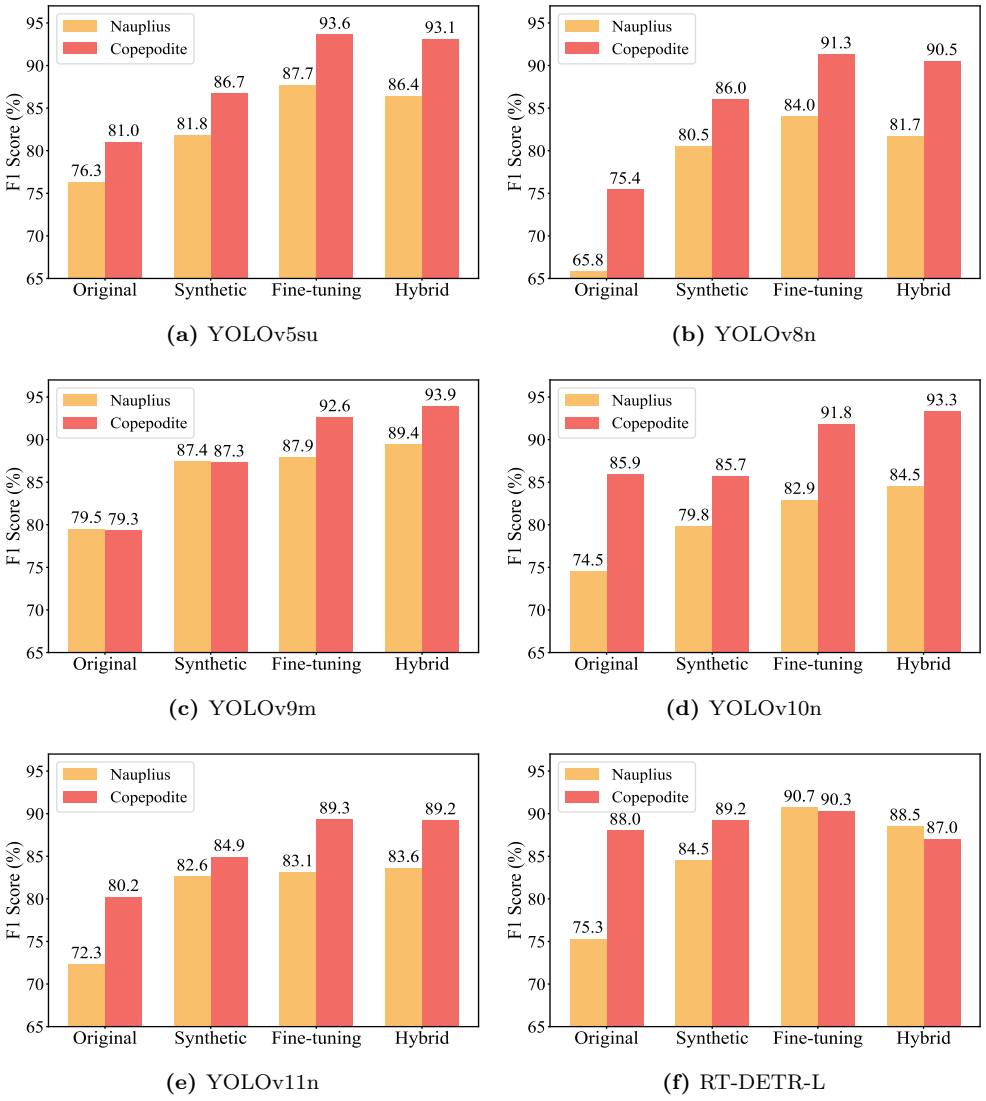
For "Nauplius," fine-tuned models achieved F1-scores of 82.9% to 90.7% and hybrid models reached 81.7% to 89.4%, both outperforming models trained solely on the original TrainSet (65.8% to 79.5%) or on the Opt-SynSet (79.8% to 87.4%). For "Copepodite," the same trend was observed. Fine-tuned models yielded F1-scores between 89.3% to 93.6%, and hybrid models 89.2% to 93.9%. Both approaches exceeded the models solely on the original TrainSet (75.4% to 88.0%) or on the Opt-SynSet (84.9% to 89.2%).

Overall, "Copepodite" detection showed a higher F1-score than "Nauplius," but the relative improvement in "Nauplius" detection was greater: fine-tuning and hybrid strategies increased F1-score by 16.4% on average for "Nauplius," compared to 12.2% for "Copepodite." Among all models, RT-DETR-L achieved the highest F1-score for "Nauplius" detection using the fine-tuning training strategy (90.7%), while YOLOv9m performed best for "Copepodite" detection with the hybrid training strategy, achieving an F1-score of 93.9%.

3.3.3 Field validation

As shown in Table 3.7, models trained with Opt-SynSet and a fine-tuning strategy outperformed those trained exclusively on the original TrainSet and existing benchmark models. In both samples, the synthetic data-enhanced models detected between 71 and 79 salmon louse nauplius instances, achieving recall rates between 87.7% and 97.5%. This substantially improved upon models trained solely on original data, which detected 45 to 69 instances with recall rates ranging from 55.6% to 85.2%, and existing methods, which detected 43 to 49 instances with recall rates of 53.1% to 60.5%. A similar trend was observed for "Copepodite" detection, where synthetic data-enhanced models identified 8 to 10 instances, achieving recall rates between 80.0% and 100.0%. In contrast, models trained solely on the original TrainSet detected 0 to 4 instances with recall rates between 0% and 40%, while existing methods exhibited recall rates ranging from 0% to 60%.

Beyond the overall improvement in recall, the integration of synthetic data also reduced performance disparities across models with varying parameter sizes. Smaller



(a) YOLOv5su **(b)** YOLOv8n

(c) YOLOv9m **(d)** YOLOv10n

(e) YOLOv11n **(f)** RT-DETR-L

Figure 3.11: Detection performance for "Nauplius" and "Copepodite" stages under four training strategies: Original, Synthetic, Fine-Tuning, and Hybrid.

models benefited the most from synthetic data, with YOLOv11n achieving an 87.0% relative performance improvement. Likewise, both YOLOv8n and YOLOv10n exhibited relative recall improvements that exceeded 60.0%. In contrast, models trained solely on the original TrainSet exhibited substantial performance variation, with absolute recall differences reaching 29.7 percentage points (50.5% to 80.2%). After training with the Opt-SynSet, this gap was reduced to 8.8 percentage points (89.0% to 97.8%).

Table 3.7: Detection recall comparison for salmon louse "Nauplius" and "Copepodite" across models in two field-collected samples.

		Sample 1		Sample 2		Overall Recall			
Model	**Thr.**	**Nau**	**Cop**	**Nau**	**Cop**	**Nau (%)**	**Cop (%)**	**All (%)**	**ΔAll**
Ground Truth	–	45	0	36	10	100.0	100.0	100.0	–
Expert	–	37	0	–	–	–	–	–	–
Existing Models									
Zhang-YOLOv8n (Zhang et al., 2024a)	0.583	26	0	23	6	60.5	60.0	60.4	–
Zheng-YOLOv8n (Zheng et al., 2024)	0.451	22	0	21	0	53.1	0.0	47.3	–
Original TrainSet Only									
YOLOv5su	0.432	36	0	33	4	85.2	40.0	80.2	*Base*
YOLOv8n	0.248	25	0	27	3	64.2	30.0	60.4	*Base*
YOLOv9m	0.332	36	0	34	0	86.4	0.0	76.9	*Base*
YOLOv10n	0.308	20	0	26	4	56.8	40.0	54.9	*Base*
YOLOv11n	0.191	18	0	27	1	55.6	10.0	50.5	*Base*
RT-DETR-L	0.589	27	0	34	4	75.3	40.0	71.4	*Base*
Opt-SynSet + original TrainSet (Ours)									
YOLOv5su	0.451	37	0	34	**10**	87.7	**100.0**	89.0	+8 (+11.0%)
YOLOv8n	0.210	**44**	0	**35**	**10**	**97.5**	**100.0**	**97.8**	+34 (+61.8%)
YOLOv9m	0.184	**44**	0	**35**	**10**	**97.5**	**100.0**	**97.8**	+19 (+27.1%)
YOLOv10n	0.256	36	0	**35**	**10**	87.7	**100.0**	89.0	+31 (+62.0%)
YOLOv11n	0.446	42	0	**35**	9	95.1	90.0	94.5	**+40 (+87.0%)**
RT-DETR-L	0.472	42	0	**35**	8	95.1	80.0	93.4	+20 (+30.8%)

Note: 'Original TrainSet Only' refers to models trained solely on the original training set. 'Opt-SynSet + Original TrainSet' refers to models pretrained on the optimized synthetic dataset and fine-tuned on the original data. 'Thr.' denotes the confidence threshold used for detection, selected to maximize F1-score on the validation set. 'Nau' and 'Cop' indicate nauplius and copepodite stages, respectively. 'ΔAll' represents the absolute and relative improvement in overall recall compared to models trained only on the original data. Bold values indicate the best-performing model in each category. The row with a blue background marks the best overall model.

Among all models, YOLOv8n and YOLOv9m trained with the Opt-SynSet and fine-tuning strategy achieved the highest performance, with an overall recall of 97.8%, detecting 79 out of 81 "Nauplius" and all 10 "Copepodite" instances. Compared to models trained solely on the original TrainSet, this represented a 61.8% and 27.1% improvement in overall recall, respectively.

Notably, during ground truth preparation, human experts achieved a recall rate of 82.2% for "Nauplius" detection in Sample 1. This suggests that the synthetic data-enhanced models outperformed human experts in detecting salmon louse larvae in complex real-world samples.

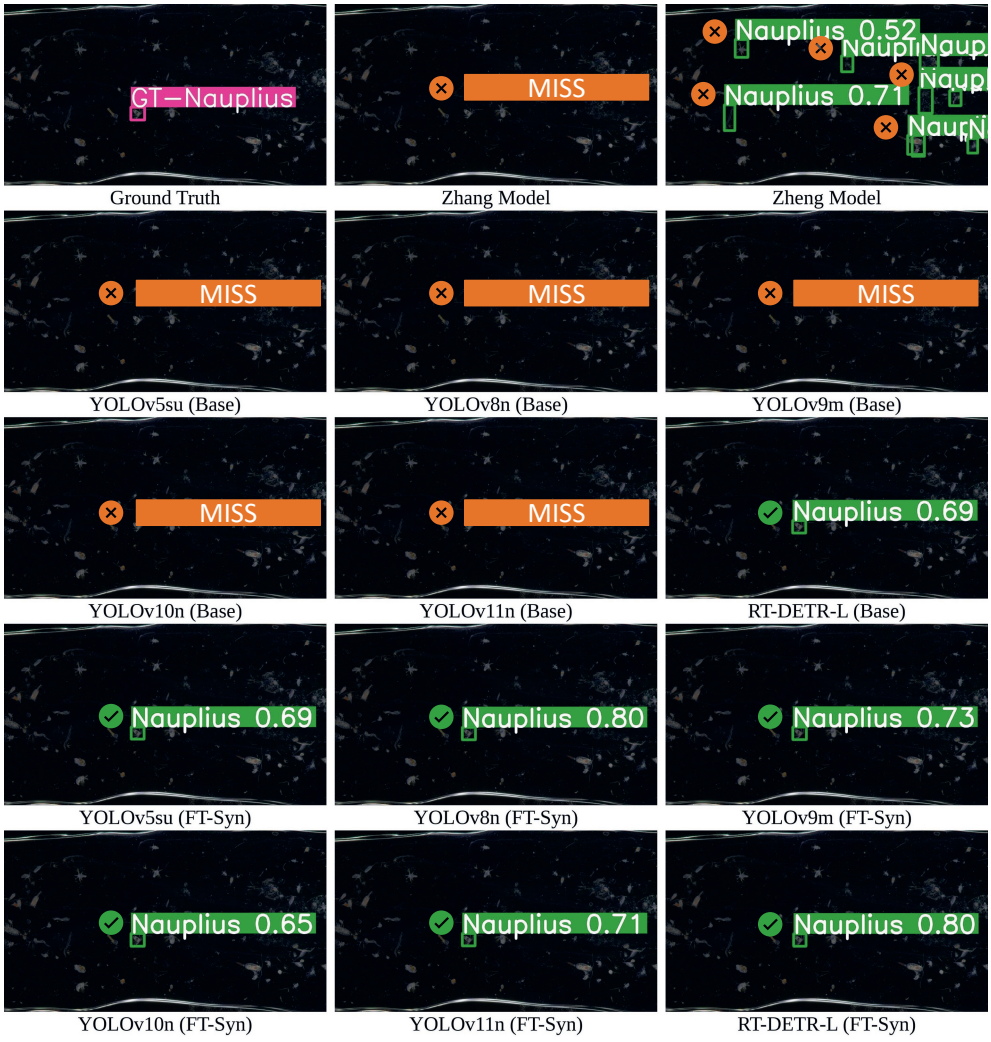


Figure 3.12: Detection results of multiple models on real seawater images. ✔ indicates correct detections, while ✘ denotes missed or incorrect detections. Base refers to models trained solely on the original TrainSet, while FT-SYT indicates models pre-trained on the Opt-SynSet and fine-tuned on the original TrainSet. The Zhang and Zheng models, along with most other Base models (except RT-DETR-L), exhibited false negatives or false positives (✘). In contrast, all FT-SYT models accurately detected the salmon louse nauplius (✔).

3.4 Discussion

3.4.1 The impact of image synthesis factors on the quality of synthetic datasets

The orthogonal experiment results (Table 3.5) demonstrate that different factor configurations greatly influence the quality of synthetic data (R ranging from 0.0110 to 0.1669). Among these, MNN has the most substantial impact on dataset quality ($R = 0.1669$). Compared to datasets with lower background complexity, those containing 80 non-target plankton instances per image (MNN=80) led to a 25.9% improvement in the YOLOv11n F1-score. A possible explanation is that the introduction of non-target instances increased background diversity, compelling the model to focus on the intrinsic features of the salmon louse rather than relying on superficial correlations with the surrounding environment. This mitigates overfitting to background noise and enhances the model's generalization ability. This finding is consistent with (Hendrycks et al., 2021), who showed that training in a more diverse environment improves model robustness and generalizability.

The SIM variable also substantially impacts dataset quality ($R = 0.1336$), with higher SIM values resulting in better model performance with more synthetic data. This aligns with Singh et al. (2024), which reported that expanding the synthetic training dataset from 1 million to 16 million samples led to a notable improvement in SynViT-B classification accuracy, from 44% to 51%. This is likely because increased randomness in synthetic data introduced greater scene and condition variations, thereby enhancing data diversity. This effect acts similarly to implicit regularization (Geiping et al., 2023), enabling the model to learn more generalizable features and improving resilience to domain shift. Similarly, Tang & Jia (2023) found that diverse synthetic datasets mitigated reliance on low-level texture features, prompting the learning of more abstract representations and ultimately enhancing model robustness and adaptability.

In contrast, object-specific factors such as MNS ($R = 0.0421$) and IST ($R = 0.0464$) have relatively minor effects. Increasing MNS or IST did not necessarily lead to positive outcomes and might even have degraded model performance. This could be attributed to structural discrepancies between synthetic and real-world data distributions. In natural seawater, salmon louse typically occurs at low-density distributions with relatively stable sizes (Gjerde & Aslam, 2025). Excessive insertion of salmon louse instances or size variation in synthetic datasets may disrupt these natural patterns, leading the model to learn features that deviate from real-world scenarios. Consequently, this deviation increases the model's sensitivity to anomalies or deformed instances while reducing its ability to capture the naturally sparse and standardized

morphological characteristics of the salmon louse. Such discrepancies ultimately led to generalization errors and reduced model adaptability (Mohamadi et al., 2025).

Traditional augmentation factors, such as MB (R=0.0196) and CA (R=0.0191), exhibit a relatively limited impact compared to factors like MNN and SIM. This likely arises from fundamental differences in their operating mechanisms. While traditional augmentation primarily modifies superficial image characteristics, synthetic data generation fundamentally expands both the structural and semantic content of the training data distribution. As a result, although traditional augmentation enhances model adaptability to environmental variations, its ability to improve learning capabilities remains inherently constrained. In contrast, synthetic data approaches introduce novel, information-rich samples, thereby offering a more profound enhancement in model generalization and robustness.

In summary, optimizing MNN and SIM is more effective than increasing object-specific factors or relying solely on traditional augmentation techniques when using synthetic data for salmon louse detection. Furthermore, the synthetic data generation approach may be applicable to other plankton detection tasks, with similar factor optimization strategies offering promising potential for improving model performance across diverse applications.

3.4.2 Comparative analysis of different model training strategies

The results in Table 3.6 demonstrated that models trained with fine-tuning or hybrid training strategies consistently outperform those trained solely on synthetic or original data. Training exclusively on synthetic data improved F1-score by up to 17.99% over using only original data. Fine-tuning further increased the F1-score by 8.85% to 24.22%, while the hybrid training yielded gains of 7.47% to 21.95%. This superiority arises because training data quality depended not only on the diversity but also on its alignment with real-world feature distributions. Fine-tuning leverages the complementary feature distributions of synthetic and real data, effectively bridging the gap between synthetic and real-world domains (Tremblay et al., 2018). This enables the model to refine its learned representations and capture subtle features that synthetic data alone may fail to encompass (Li et al., 2023; Mishra et al., 2022). Similarly, hybrid training ensures balanced exposure to both domains, leading to substantial performance improvements. These findings are consistent with Jung et al. (2024), who reported improved performance in pine wilt disease detection by combining synthetic and real data.

Smaller models (e.g., YOLOv8n) benefited more from synthetic data than larger models (e.g., YOLOv9m and RT-DETR-L). This is likely because lightweight models, with

fewer parameters, struggle to learn complex features and contextual semantics, especially those involving higher-order statistical patterns like intricate parasite morphologies. Therefore, their learning process relies heavily on the visibility and diversity of object structures in the training data. Synthetic data mitigates these limitations by enriching sample diversity and expanding input coverage, thereby reducing generalization errors and compensating for the limited feature extraction capability of small models (Baradad Jurjo et al., 2021; Zhang et al., 2025c). This observation aligns with Ba et al. (2024), who found that diverse training data, including synthetic data, enhances model robustness in a manner similar to dropout. In contrast, larger models have stronger nonlinear modeling capabilities on the original data set. For them, synthetic data primarily improves robustness through enhanced generalization to edge cases and atypical targets, resulting in smaller gains in evaluation metrics. This is particularly beneficial for developing lightweight models such as YOLOv8n, which can achieve high accuracy with minimal computational resources. As a result, such models are well-suited for deployment in battery-powered sea-cage monitoring systems, enabling continuous tracking of salmon lice populations in remote or resource-limited environments.

In summary, we recommend strategically incorporating synthetic data into training pipelines, particularly for lightweight models. While synthetic data cannot fully substitute real-world data, it serves as a valuable resource for pre-training, enabling models to learn diverse and robust features. Furthermore, integrating synthetic and real-world data, either through hybrid training or fine-tuning, maximizes model performance by combining the strengths of both data types (Ruis et al., 2024).

3.4.3 Discussion on detection performance in real-world scenarios

Models trained with synthetic data outperformed traditional microscopic visual inspection methods that rely on expert manual observations, particularly in real-world seawater sample tests. While microscopic visual inspection remains a standard method for estimating salmon louse density, it is highly dependent on trained personnel and is both time-consuming and inefficient (Thompson et al., 2021). In our field validation, experts required 10 hours over three days to manually inspect the first sample, achieving a recall of only 82.2%(Table 3.7), consistent with previous reports of inefficiency in large-scale monitoring (Bui et al., 2021). In contrast, our model identified 97.8% of salmon louse larvae in the same sample within 30 minutes. Although expert inspection may yield accurate results for small datasets—where attention can be fully focused—it becomes increasingly error-prone when scaled. Manual analysis of plankton-rich samples involves scrutinizing tens to thousands of microscopic particles per image, demanding sustained concentration. Over prolonged sessions,

human annotators are susceptible to fatigue, cognitive overload, and attention drift, leading to potential oversight or misdetection. Furthermore, variability in individual judgment can introduce inconsistencies, especially in borderline cases. In contrast, machine learning models apply consistent criteria and maintain uniform detection performance across entire datasets, thoroughly scanning each region of every image without fatigue. These results indicate that machine learning-based methods supported by synthetic data can surpass traditional expert-based inspection in complex, real-world sample analysis.

In addition, our model demonstrates robustness in handling partially occluded or borderline cases. While performance decreases under severe occlusion due to the absence of key morphological features, such instances are mitigated by the dynamic nature of marine environments. As larvae move with the flow, their orientation and visibility change across frames. This temporal variation enables the model to successfully detect targets in at least one frame, compensating for missed detections and enhancing overall robustness in real-world scenarios.

Compared to existing machine learning models, our method exhibits superior adaptability and robustness. Field validation shows that our synthetic data-based model improves recall rates over previous machine learning implementations. Existing models (Zhang et al., 2024a; Zheng et al., 2024) are limited by simplistic laboratory conditions and incomplete developmental stage coverage. For instance, the model developed by Zhang et al. (2024a) did not account for environmental factors such as plankton occlusion and turbidity gradients, resulting in a recall rate of only 60.4% in our field validation (Table 3.7). Similarly, Zheng et al. (2024) also focused on simplified laboratory environments and excluded the copepodite stage, restricting domain adaptation. In contrast, our synthetic data approach incorporates both nauplius and copepodite stages and is optimized for high-density plankton environments. The use of synthetic data effectively expands the distribution space of the training data, enhancing the model's ability to adapt to data biases and environmental changes. This approach allows the model to learn the impact of complex factors such as environmental noise and occlusion on salmon louse larvae detection while maintaining reliable performance in high-biomass water samples.

Our synthetic data approach offers a flexible and scalable solution for detecting marine parasites, with potential for transfer across locations, time periods, and species. Although our models were initially developed using samples from a specific geographic region, they likely capture the majority of local plankton diversity. To extend applicability to other regions or target organisms, the models can be adapted by incorporating representative samples from the new environment and compositing the relevant organisms into these samples using synthetic data for further training. This

enables rapid generation of samples associated with new regions or seasonal variations, while continued integration of real-world data further refines model performance over time. Thus, the synthetic data method not only addresses the challenges of limited annotated data but also enhances model robustness and transferability, supporting large-scale, real-time monitoring of salmon lice and other marine organisms across diverse aquaculture environments.

In summary, the use of synthetic data addresses key challenges in data collection and annotation, enabling robust model performance in real-world conditions and facilitating rapid adaptation to new environments. This approach has surpassed expert performance in salmon louse larvae detection, demonstrating strong potential for large-scale application in seawater environments.

3.4.4 Limitations and future improvements

False positives remain a challenge in machine learning-based salmon lice detection due to the high variability in seawater conditions and the morphological similarity of many plankton species to salmon louse larvae (Fernandez-Gonzalez et al., 2022). Although the use of synthetic data can reduce false positives by an average of 80% and minimize the need for manual post-processing, further minimizing false detections is essential for fully automated salmon lice counting. False positives can disrupt tracking and counting algorithms, ultimately reducing overall system accuracy.

To quantify this limitation, we conducted a frame-level analysis of the fine-tuned YOLOv9m model on Sample 1 from the field validation phase. The model identified 587 true positives and approximately 315 false negatives. Within frames where salmon lice were present, 109 false positives were recorded. Notably, due to the extreme sparsity of targets, approximately 98% of frames contained no lice. When considering the full video, a total of 2,549 false positives were detected, corresponding to an average of 0.042 false positives per frame. While relatively low at the frame level, such false detections may accumulate in large-scale deployments and impact downstream tracking and counting tasks. Nevertheless, in real-world settings involving human-in-the-loop verification, even a single-frame detection is often sufficient to confirm lice presence, partially mitigating the practical impact of false positives and false negatives.

To further address this issue, future work will focus on improving the data synthesis approach by incorporating a broader diversity of background plankton into the synthetic data. This will enhance the quality of synthetic datasets and improve model performance under complex marine conditions. Additionally, we will leverage the temporal consistency present in salmon lice videos to refine detection accuracy. By

integrating multi-frame information, the system can effectively differentiate between true detections and transient noise, reducing false positives and false negatives.

Another limitation of the current method is its reliance on standard light sources for detecting salmon louse larvae based on morphological features. While effective in many scenarios, this approach provides limited discrimination in environments with dense plankton populations or poor water clarity. Fluorescent lighting presents a potential solution, as salmon louse larvae emit a brighter and more distinct color compared to other plankton under such conditions, offering an additional means of differentiation. A promising direction for future research is the integration of standard and fluorescent illumination. By synchronously capturing the bio-optical characteristics of the target organisms in both the visible and fluorescent spectral bands through feature fusion, detection accuracy can be further improved.

3.5 Conclusion

We proposed a SAM-based synthetic data generation framework to simulate diverse seawater conditions by segmenting and compositing organisms into a realistic marine background. This approach addresses the scarcity of annotated training data for salmon louse larvae detection in seawater, resulting in measurable improvements in model performance. Our key findings are as follows:

1. Through orthogonal experiments, we identified that MNN ($R = 0.1669$) and SIM ($R = 0.1336$) have the most significant impact on synthetic data quality. Higher values of MNN and SIM consistently lead to improved synthetic dataset quality and model performance. In contrast, the effect of MNS ($R = 0.0421$) is relatively limited, and excessive adjustment ($MNS > 3$) can degrade dataset quality and reduce model performance. Conventional augmentation methods such as MB and CA contribute minimally ($R < 0.02$). These findings suggest that, for salmon louse larvae detection in seawater, synthetic data generation should prioritize increasing background complexity. In particular, incorporating a greater number of non-target organisms is more effective than solely increasing target instances.

2. The use of synthetic data significantly enhances the models' performance in detecting salmon louse larvae. Applying fine-tuning or hybrid training strategies with a synthetic data approach boosted the model's performance, with F1-score improvements of 8.85% to 24.22% and 7.47% to 21.95%, respectively, over the baseline method.

3. In real-world evaluations, the model pre-trained with synthetic data and fine-

tuned on original data detected 89.0% to 97.8% of salmon louse nauplius and copepodite, outperforming well-trained marine experts. Compared to baseline models, this represents a detection improvement of 11.0% to 87.0%, significantly exceeding previous methods (47.3% to 60.4%). This confirms the robustness and practicality of our approach in dynamic, large-scale marine monitoring.

In summary, our synthetic data method provides a practical and scalable solution to data scarcity in marine organism detection. It establishes a solid foundation for deploying machine learning-based systems in complex aquatic environments and is adaptable for broader ecological monitoring tasks beyond salmon lice detection.

Chapter 4

Online synthetic training for annotation-scarce salmon lice detection

This chapter has been accepted as:

Zhang, C., Stival, L., Gansel, L. C., Bracke, M., & da Silva Torres, R. (2026). Online synthetic training for annotation-scarce salmon lice detection. *SIBGRAPI Conference on Graphics, Patterns and Images*.

Abstract

Automatic detection of early-stage salmon lice in seawater using computer vision is challenging due to the small object size, complex aquatic backgrounds, and limited availability of annotated training data. In existing synthetic-data approaches, which alleviate annotation scarcity by generating additional training images offline, the resulting dataset remains fixed throughout training and may provide limited diversity under extremely small-label regimes. Here, we propose **SDOST** (**S**almon Lice **D**etection based on **O**nline **S**ynthetic **T**raining), a simple yet effective framework that synthesizes training samples on the fly by compositing segmented lice and impurity instances onto target-free real backgrounds. SDOST further employs feature-proximal replay (FPR) to reuse realistic synthetic samples and Fast Gradient Sign Method (FGSM) regularization to enhance adversarial robustness. Experiments on five annotation-scarce settings show that SDOST consistently outperforms both source-only and offline synthetic training baselines. It achieves the best mAP_{50} and F1 scores in all settings and the best $mAP_{50–95}$ in four out of five settings. Compared with offline-only training, SDOST improves F1 by 4.3%–17.7%, while yielding substantially larger gains of 13.8%–166.4% over source-only training.

4.1 Introduction

Salmon lice (*Lepeophtheirus salmonis*) severely impact marine salmon aquaculture, causing substantial economic losses (Taranger et al., 2015). During their early planktonic stages, larvae freely disperse, seek host fish, suppress host immunity, and increase infection pressure on wild salmon populations (Jeong et al., 2021). Early detection of planktonic-stage lice appears, therefore, critical for timely intervention, assessing parasite pressure and recruitment, and developing new control measures, particularly as regulatory authorities strictly limit lice counts per fish (Jeong et al., 2023).

Existing detection methods include manual observation (Thompson et al., 2022), molecular assays (Krolicka et al., 2022), and computer vision (CV) (Zhang et al., 2024a, 2025b). Manual and molecular approaches are labor-intensive, time-consuming, and costly (Bui et al., 2021). In contrast, CV-based methods offer rapid, scalable alternatives, but their performance is often constrained by limited training data and degraded by false positives in complex environments. To address these challenges, recent studies showed substantial benefits of using synthetic data (Zheng et al., 2024; Zhang et al., 2025b; Souza et al., 2025). However, these approaches typically rely on offline synthetic training, where pre-generated samples remain fixed. In label-scarce regimes, static datasets restrict appearance diversity, limit exposure to unseen

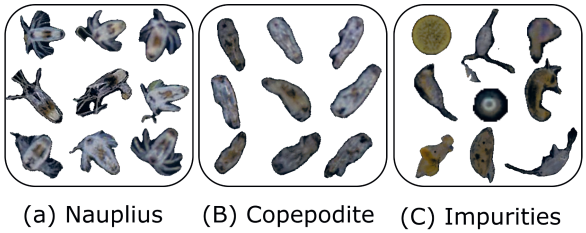


Figure 4.1: Examples of nauplius, copepodite, and impurity instances.

visual configurations, and cause models to overfit to artificial object–background associations (Kollar et al., 2021).

To address these limitations, we now propose SDOST (Salmon lice Detection based on Online Synthetic Training), a framework for annotation-scarce environments. SDOST first uses the Segment Anything Model (SAM) (Kirillov et al., 2023) to extract lice and impurity instances (Figure 4.1) from a small annotated set. During training, SDOST dynamically transforms and composites these instances onto real background images that contain no target objects. Unlike offline training, this generates a continuously varying synthetic image stream that prevents dataset stagnation.

To bridge the synthetic-to-real domain gap, SDOST incorporates Feature-Proximal Replay (FPR). Inspired by memory-replay techniques (Liu et al., 2020; Wang et al., 2021), FPR identifies and stores synthetic samples whose feature representations closely match real images, revisiting them to maintain diversity and domain relevance. Finally, we apply the lightweight Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) regularization to mini-batches, generating adversarial samples that further mitigate overfitting and enhance generalization.

The main contributions of this paper are as follows:

- We propose SDOST, an *online* synthetic training framework that dynamically generates diverse training samples on the fly, overcoming the limitations of fixed offline datasets for annotation-scarce salmon lice detection.
- We propose FPR, a novel replay mechanism that retains and revisits synthetic samples with realistic feature representations, effectively bridging the synthetic-to-real domain gap.
- We integrate lightweight FGSM-based regularization during synthesis to improve detector robustness against input perturbations and enhance real-world generalization.
- We systematically demonstrate that SDOST outperforms offline synthesis across

multiple annotation scales and backbone architectures, proving its efficacy under severe label scarcity.

4.2 Methodology

Figure 4.2 outlines SDOST, which integrates three elements into a unified training pipeline: (i) data preparation and online synthesis, (ii) FPR, (iii) FGSM training. Given a limited annotated dataset $\mathcal{D} = \{(I_i, \mathbf{b}_i)\}_{i=1}^{N}$, where $\mathbf{b}_i = \{b_{iu}\}_{u=1}^{m_i}$ denotes the target bounding boxes for image I_i, we construct a lice instance library $\mathcal{T}$, an impurity instance library $\mathcal{P}$, and a pool of real target-free backgrounds $\mathcal{D}_{\text{bg}}$. SDOST then dynamically generates mixed synthetic batches, stabilizes training via feature replay, and applies adversarial perturbations for robust optimization.

4.2.1 Online Synthetic Image Generation

Instead of pre-generating a fixed dataset (Zhang et al., 2025b), SDOST integrates image synthesis into training to alleviate annotation scarcity. We adopt the augmentation parameterization of (Zhang et al., 2025b) but reformulate it as an online component, drawing all parameters independently for each generated image. Figure 4.3 shows examples of real backgrounds, resulting synthetic images, and generated ground-truth boxes.

Instance library construction We apply SAM (Kirillov et al., 2023) to annotated target regions to extract foreground masks, forming the lice library $\mathcal{T}$. For the impurity library $\mathcal{P}$, we prompt SAM at coordinates sampled outside target regions to segment non-target objects, filtering out overly large segments. This offline step occurs once before the training begins. Figure 4.1 shows example instances of salmon louse nauplius, copepodite, and impurities.

Synthetic image generation At each step, we sample a background I_{bg} from $\mathcal{D}_{\text{bg}}$, draw n_t lice and n_p impurity templates ($\{T_j\}_{j=1}^{n_t} \subset \mathcal{T}$, $\{P_k\}_{k=1}^{n_p} \subset \mathcal{P}$), apply random geometric transformations (ϕ_j, ψ_k), and composite them onto the background via alpha blending. The synthetic image x and label y are:

$$(x, y) = \mathcal{G}\left(I_{\text{bg}}, \{(T_j, \phi_j)\}_{j=1}^{n_t}, \{(P_k, \psi_k)\}_{k=1}^{n_p}; \xi\right)$$

where $\mathcal{G}$ is the compositing function and ξ denotes appearance perturbations (e.g., additive brightness and Gaussian blur). We sample $n_t \in 1, 2, 3$ with rotations in $[0°, 360°]$ and up to 80 impurities per image, all placed at random locations. The detection label maps transformed lice masks to axis-aligned boxes:

$$y = \left\{ \left(\text{box}(\phi_j(T_j)),\, c_j\right) \right\}_{j=1}^{n_t}$$

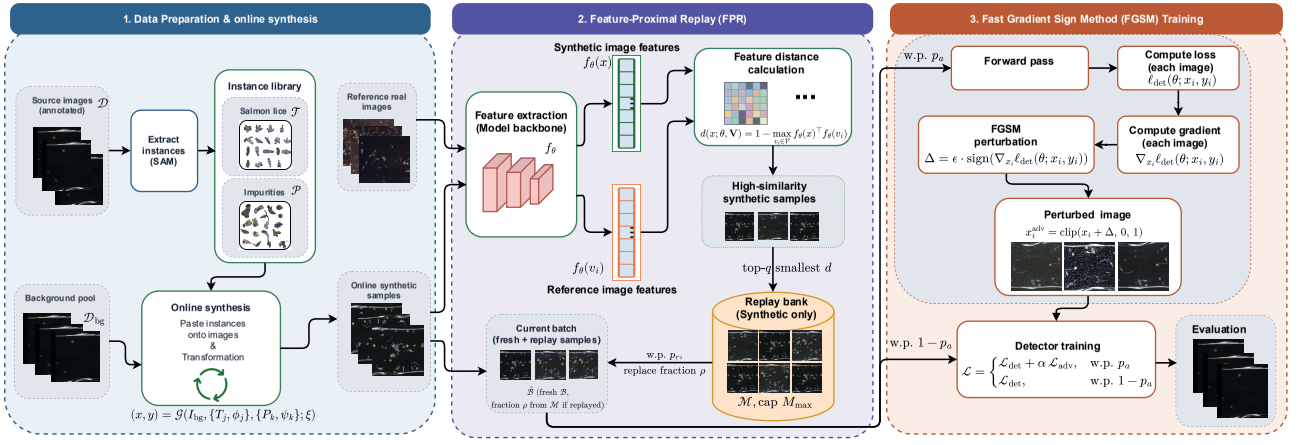
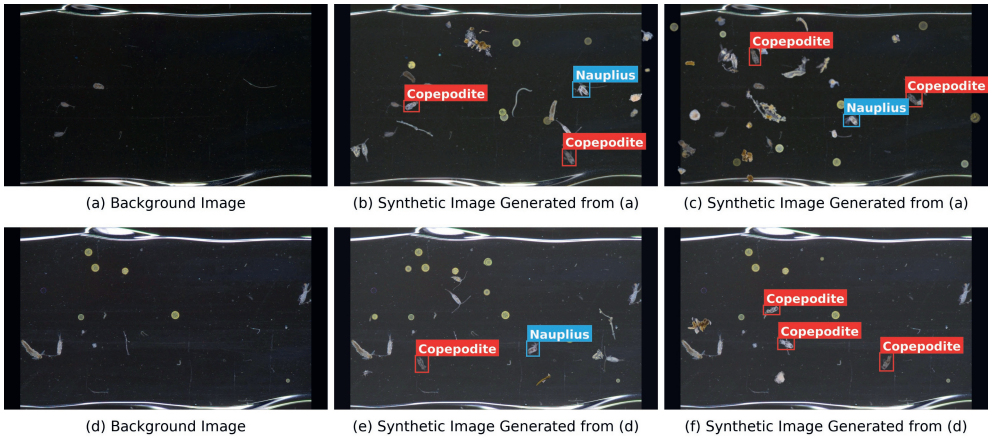


Figure 4.2: Overview of the proposed SDOST framework. SDOST first constructs lice and impurity instance libraries from a small annotated dataset, then synthesizes training images online by compositing sampled instances onto target-free real backgrounds. A FPR memory re-injects real-like synthetic images into training, and FGSM adversarial regularization is applied with a fixed probability to improve robustness of the model.



(a) Background Image (b) Synthetic Image Generated from (a) (c) Synthetic Image Generated from (a)

(d) Background Image (e) Synthetic Image Generated from (d) (f) Synthetic Image Generated from (d)

Figure 4.3: Examples of background and the corresponding synthetic images.

where c_j is the class ID. Impurities act purely as distractors without labels.

4.2.2 Lightweight FPR

To stabilize training, FPR maintains a memory $\mathcal{M}$ of synthetic samples that closely match real images in the detector's feature space, re-injecting them into subsequent mini-batches.

Let $f_\theta(\cdot)$ be the ℓ_2-normalized image-level representation from the detector θ, obtained by global-average-pooling and concatenating the backbone feature maps across all pyramid levels. Using a subset of n_r real images $\mathcal{V} = \{v_i\}_{i=1}^{n_r}$ (drawn from the validation set for convenience; only image content is used, and no labels are accessed), we build a reference feature bank $\mathbf{V} = \{f_\theta(v_i)\}_{i=1}^{n_r}$. The feature distance for a synthetic sample x is:

$$d(x; \theta, \mathbf{V}) = 1 - \max_{v_i \in \mathcal{V}} f_\theta(x)^\top f_\theta(v_i)$$

A smaller distance indicates higher similarity to the nearest real image. The memory $\mathcal{M} = \{(x_j, y_j, d_j, t_j)\}_{j=1}^{|\mathcal{M}|}$ stores the image, label, distance, and insertion step t_j. Every C steps, we score the current mini-batch and insert the q samples with the smallest distances into $\mathcal{M}$. Because the representation evolves, we recompute $\mathbf{V}$ and d_j every R steps. To prioritize recent samples, we cap capacity at $M_{\max}$ by retaining items with the smallest effective distance:

$$\tilde{d}_j(s) = \frac{d_j}{\lambda^{s-t_j}}, \qquad 0 < \lambda < 1$$

where s is the current step and λ is the freshness decay factor.

Let $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B}$ be the current synthetic mini-batch. With probability p_r, FPR replaces a fraction ρ of $\mathcal{B}$ with memory samples, yielding a mixed batch $\tilde{\mathcal{B}}$ of the same

size. The optimization batch $\hat{\mathcal{B}}$ is thus $\tilde{\mathcal{B}}$ if FPR triggers or $\mathcal{B}$ otherwise.

4.2.3 FGSM Regularization

After the replay warm-up stage, we trigger adversarial training with probability p_a at each step, independently of replay. For the active optimization batch $\hat{\mathcal{B}}$ (size B), we compute gradients of the detection loss ℓ_{det} (with θ held fixed) to generate perturbed images:

$$x_i^{\text{adv}} = \text{clip}(x_i + \epsilon \cdot \text{sign}(\nabla_{x_i} \ell_{\text{det}}(\theta; x_i, y_i)), 0, 1)$$

where ϵ controls the perturbation magnitude. We optimize the detector using a weighted combination of clean and adversarial losses:

$$\mathcal{L}_{\text{det}} = \frac{1}{B} \sum_{i=1}^{B} \ell_{\text{det}}(\theta; x_i, y_i), \quad \mathcal{L}_{\text{adv}} = \frac{1}{B} \sum_{i=1}^{B} \ell_{\text{det}}(\theta; x_i^{\text{adv}}, y_i)$$

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{det}} + \alpha \, \mathcal{L}_{\text{adv}}, & \text{if FGSM is triggered (prob. } p_a) \\ \mathcal{L}_{\text{det}}, & \text{otherwise} \end{cases}$$

where α is the adversarial weight. This regularizes the entire batch, encouraging robustness to small appearance perturbations while reusing the original annotations.

4.3 Experiments

4.3.1 Dataset Setting

We use the salmon louse larvae dataset from (Zhang et al., 2025b), which contains 2,518 training images, 420 validation images, and 838 test images, containing 1,416/318/421 Nauplius and 1,190/134/439 Copepodite instances, respectively. To evaluate annotation-scarce detection, we sample small subsets from the original training set instead of using all 2,518 training images, forming five low-label data settings.

We denote each data setting as **S**x**-B**y, where x and y indicate the number of annotated lice instances and target-free background images available for training, respectively. The instance library is extracted from the corresponding source images. Source-only uses the source images directly, while the offline synthetic method and SDOST use the extracted instances and background pool to synthesize training images. Fixed validation and test sets are used for model selection and final evaluation. Table 4.1 summarizes the five data settings.

Table 4.1: Annotation-scarce data settings sampled from the original training dataset (Zhang et al., 2025b). Source-only uses the source images, while the offline synthetic training method uses the listed fixed synthetic datasets (last column), and SDOST generates synthetic samples online.

Setting	Source Images	Instances	Background Images	Offline Synthetic Images
S200-B2140	200	200	2140	34240
S100-B1070	100	100	1070	17120
S20-B1070	19	20	1070	17120
S10-B1070	10	10	1070	17120
S4-B1070	4	4	1070	17120

4.3.2 Implementation Details

Unless otherwise stated, all main experiments were performed using RetinaNet (Lin et al., 2017b) with a ResNet-50-FPN backbone (He et al., 2016). All results are averaged over three seeds (42, 43, 44). The detector head was trained from scratch, while the backbone was initialized with ImageNet-pretrained weights (Deng et al., 2009). Input images were resized to a short side of 512 pixels, with the long side capped at 896 pixels. Models were trained for up to 3,250 steps with batch size of 32. We optimized the detector with AdamW (learning rate 10^{-4}, weight decay 10^{-4}). An exponential moving average (EMA) of the model weights was maintained for evaluation.

For FPR, memory capacity was $M_{\max} = 256$ samples, with insertion beginning after a 125-step warm-up. Every $C = 3$ synthetic steps, the current mini-batch was scored against the reference bank, inserting the top $q = 2$ candidates (smallest feature distances) into $\mathcal{M}$. Replay occurred with probability $p_r = 0.25$, replacing a fraction $\rho = 0.25$ of the mini-batch with memory samples. We used a freshness decay $\lambda = 0.992$ and limited the reference bank to $n_r = 64$ validation images. The reference bank $\mathbf{V}$ and distances d_j were recomputed every $R = 125$ steps. Replay samples were stored on the GPU to minimize data-transfer overhead.

For the final configuration, we enabled FGSM regularization after step 125 to match the replay start. We applied FGSM with probability $p_a = 0.50$, perturbation magnitude $\epsilon = 2/255$, and adversarial loss weight $\alpha = 0.25$, and ran all experiments on an NVIDIA A100 80GB PCIe GPU. With a mini-batch size of 32, training took ≈14.7 ms per image for the offline baseline and ≈32.0 ms per image for the full SDOST. All hyperparameters were tuned on the validation set, and the reported results use this optimized configuration.

4.3.3 Main Results

Table 4.2 summarizes our evaluation of all methods across five source-limited settings using a ResNet-50 RetinaNet. The source-only baseline relies exclusively on real annotated images, whereas offline synthesis pre-generates a fixed dataset. SDOST achieves the highest mAP50, precision, recall, and F1 across all settings, improving F1 by 4.3%–17.7% over offline synthesis and 13.8%–166.4% over source-only, and obtains the best mAP50–95 in four out of five settings. These results suggest that online synthesis is more effective than relying only on scarce real annotations or using fixed synthetic data. As shown in Figure 4.4, SDOST detects targets missed by the baselines while avoiding their misclassified false positives, in line with the quantitative results.

SDOST excels particularly in low-label regimes, where the baselines degrade substantially. For instance, on **S20-B1070** (only 20 instances of salmon louse larvae across 19 images), source-only training drops to 0.462 mAP50 and 0.504 F1, while SDOST reaches 0.787 mAP50 and 0.765 F1, maintaining competitive performance despite severe annotation scarcity. We observe similar gains on **S10-B1070** and **S4-B1070**, where SDOST consistently outperforms both baselines.

While all methods improve with more source data, the shrinking performance gap between **S100-B1070** and **S200-B2140** indicates diminishing returns from scaling annotations. In the largest setting (**S200-B2140**), the source-only approach yields the highest mAP50–95, but SDOST maintains the best mAP50, precision, recall, and F1. This suggests that while abundant real images improve localization accuracy, SDOST better balances precision and recall. Overall, SDOST is most beneficial under annotation scarcity, where fixed synthetic datasets are less effective.

4.3.4 Ablation Study

Table 4.3 details our ablation study on the **S200-B2140** setting. Compared to the source-only baseline, offline synthetic training improves most object-level metrics—boosting mAP50, precision, recall, and F1 but substantially drops mAP50–95 from 0.362 to 0.267. This suggests that fixed synthetic datasets enhance recognition but restrict localization diversity.

Transitioning to online synthetic training improves all metrics over the offline approach, raising mAP50–95 to 0.352, mAP50 to 0.844, and F1 to 0.830. While its mAP50–95 remains slightly below the source-only baseline, it surpasses it in all other metrics. We conclude that dynamically refreshing samples provides greater diversity and mitigates the localization degradation caused by fixed datasets.

Table 4.2: Main results across annotation-scarce training settings. Values are mean ± standard deviation over three random seeds. Best values are highlighted.

Dataset	Method	mAP50–95	mAP50	Precision	Recall	F1
S200-B2140	Source only	**0.362 ± 0.0113**	0.765 ± 0.0098	0.740 ± 0.0231	0.756 ± 0.0282	0.744 ± 0.0080
	Offline Syn.	0.267 ± 0.0047	0.794 ± 0.0077	0.859 ± 0.0160	0.760 ± 0.0100	0.806 ± 0.0056
	SDOST (Ours)	0.360 ± 0.0046	**0.859 ± 0.0075**	**0.891 ± 0.0201**	**0.808 ± 0.0316**	**0.847 ± 0.0107**
S100-B1070	Source only	0.306 ± 0.0206	0.684 ± 0.0272	0.627 ± 0.0345	0.700 ± 0.0211	0.658 ± 0.0172
	Offline Syn.	0.284 ± 0.0129	0.795 ± 0.0222	0.851 ± 0.0223	0.752 ± 0.0213	0.798 ± 0.0065
	SDOST (Ours)	**0.371 ± 0.0236**	**0.851 ± 0.0065**	**0.884 ± 0.0099**	**0.786 ± 0.0098**	**0.832 ± 0.0080**
S20-B1070	Source only	0.134 ± 0.0106	0.462 ± 0.0270	0.530 ± 0.0249	0.486 ± 0.0268	0.504 ± 0.0239
	Offline Syn.	0.266 ± 0.0093	0.732 ± 0.0263	0.759 ± 0.0271	0.671 ± 0.0131	0.710 ± 0.0181
	SDOST (Ours)	**0.324 ± 0.0095**	**0.787 ± 0.0082**	**0.812 ± 0.0194**	**0.726 ± 0.0076**	**0.765 ± 0.0102**
S10-B1070	Source only	0.097 ± 0.0072	0.363 ± 0.0058	0.413 ± 0.0051	0.416 ± 0.0189	0.406 ± 0.0132
	Offline Syn.	0.205 ± 0.0470	0.607 ± 0.0439	0.665 ± 0.0257	0.544 ± 0.0214	0.586 ± 0.0105
	SDOST (Ours)	**0.300 ± 0.0279**	**0.744 ± 0.0403**	**0.736 ± 0.0458**	**0.649 ± 0.0456**	**0.684 ± 0.0195**
S4-B1070	Source only	0.042 ± 0.0121	0.180 ± 0.0572	0.380 ± 0.1128	0.215 ± 0.0632	0.220 ± 0.0406
	Offline Syn.	0.162 ± 0.0112	0.498 ± 0.0067	0.630 ± 0.0287	0.425 ± 0.0115	0.498 ± 0.0298
	SDOST (Ours)	**0.221 ± 0.0377**	**0.605 ± 0.0547**	**0.644 ± 0.0239**	**0.554 ± 0.0692**	**0.586 ± 0.0452**

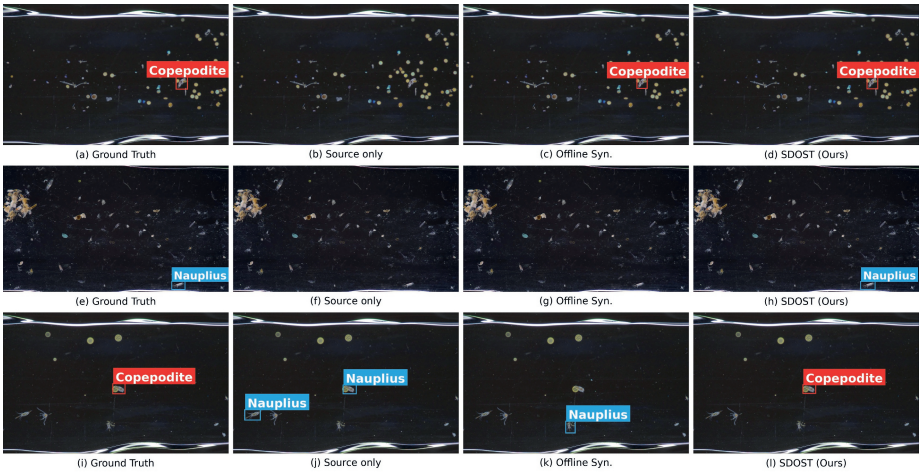


Figure 4.4: Qualitative comparison on the test set under the **S200-B2140** setting. For each example (row), we show, from left to right, the ground truth and the predictions of the source-only baseline, the offline synthetic baseline (Offline Syn.), and our SDOST. Red boxes denote *Copepodite* and blue boxes denote *Nauplius*. SDOST detects targets that the baselines miss (rows 1–2) and avoids the misclassified false positives produced by the baselines (row 3).

Adding either FGSM or FPR individually enhances the online baseline. Online + FGSM increases mAP50–95, mAP50, and F1, while Online + FPR similarly boosts recall and mAP scores. Combining both components yields our full SDOST model, which achieves peak mAP50 (0.859), recall (0.808), and F1 (0.847). Although SDOST

Table 4.3: Ablation study on **S200-B2140**. Values are mean ± standard deviation over three random seeds; best values are highlighted.

Method	Offline	Online	FPR	FGSM	mAP50–95	mAP50	Precision	Recall	F1
Source only					**0.362 ± 0.0113**	0.765 ± 0.0098	0.740 ± 0.0231	0.756 ± 0.0282	0.744 ± 0.0080
Offline Syn.	✓				0.267 ± 0.0047	0.794 ± 0.0077	0.859 ± 0.0160	0.760 ± 0.0100	0.806 ± 0.0056
Online		✓			0.352 ± 0.0029	0.844 ± 0.0095	0.887 ± 0.0172	0.779 ± 0.0272	0.830 ± 0.0091
Online + FGSM		✓		✓	0.359 ± 0.0081	0.857 ± 0.0115	**0.897 ± 0.0114**	0.800 ± 0.0144	0.845 ± 0.0122
Online + FPR		✓	✓		0.357 ± 0.0034	0.846 ± 0.0053	0.870 ± 0.0073	0.795 ± 0.0143	0.830 ± 0.0107
SDOST (Ours)		✓	✓	✓	0.360 ± 0.0046	**0.859 ± 0.0075**	0.891 ± 0.0201	**0.808 ± 0.0316**	**0.847 ± 0.0107**

slightly trades off precision (0.897 down to 0.891) compared to Online + FGSM, it improves overall recall and F1. Ultimately, FGSM drives precision and confidence, while FPR balances recall and overall detection when combined with online synthesis.

4.3.5 Backbone Generalization

Table 4.4 evaluates SDOST across different detector backbones on **S200-B2140**. SDOST consistently outperforms both source-only and offline synthetic baselines across lightweight and standard architectures, including EfficientNet-B0 (Tan & Le, 2019), MobileNetV3 (Howard et al., 2019), ResNet-18, and ResNet-50 (He et al., 2016).

The gains are consistent regardless of model size. For MobileNetV3, SDOST improves mAP50–95 to 0.354 compared with 0.277 for offline synthesis and 0.283 for source-only training. Similar improvements are observed with ResNet-18, where SDOST reaches a mAP50–95 of 0.355. These results indicate that online synthesis is effective across different backbone capacities. ResNet-50 achieves the strongest overall performance, yielding the highest mAP50 (0.859), precision (0.891), recall (0.808), and F1 (0.847). Overall, SDOST's consistent gains across compact and standard backbones demonstrate its architecture-agnostic behavior and suitability for resource-constrained deployments.

Table 4.4: Backbone comparison on **S200-B2140**. Values are mean ± standard deviation over three random seeds; best values within each backbone are highlighted.

Backbone	Method	mAP50–95	mAP50	Precision	Recall	F1
EfficientNet-B0	Source only	0.349 ± 0.0113	0.747 ± 0.0090	0.699 ± 0.0164	0.723 ± 0.0055	0.710 ± 0.0066
	Offline Syn.	0.274 ± 0.0190	0.790 ± 0.0237	0.828 ± 0.0150	0.761 ± 0.0238	0.792 ± 0.0102
	SDOST (Ours)	**0.361 ± 0.0088**	**0.849 ± 0.0115**	**0.853 ± 0.0074**	**0.776 ± 0.0117**	**0.813 ± 0.0050**
MobileNetV3	Source only	0.283 ± 0.0039	0.628 ± 0.0169	0.604 ± 0.0273	0.623 ± 0.0164	0.610 ± 0.0209
	Offline Syn.	0.277 ± 0.0051	0.799 ± 0.0044	0.849 ± 0.0111	0.724 ± 0.0198	0.782 ± 0.0095
	SDOST (Ours)	**0.354 ± 0.0076**	**0.845 ± 0.0033**	**0.859 ± 0.0257**	**0.772 ± 0.0225**	**0.812 ± 0.0084**
ResNet-18	Source only	0.339 ± 0.0053	0.705 ± 0.0099	0.700 ± 0.0356	0.683 ± 0.0228	0.683 ± 0.0135
	Offline Syn.	0.274 ± 0.0211	0.794 ± 0.0144	0.827 ± 0.0196	0.738 ± 0.0091	0.779 ± 0.0114
	SDOST (Ours)	**0.355 ± 0.0065**	**0.858 ± 0.0063**	**0.875 ± 0.0238**	**0.780 ± 0.0079**	**0.825 ± 0.0061**
ResNet-50	Source only	**0.362 ± 0.0113**	0.765 ± 0.0098	0.740 ± 0.0231	0.756 ± 0.0282	0.744 ± 0.0080
	Offline Syn.	0.267 ± 0.0047	0.794 ± 0.0077	0.859 ± 0.0160	0.760 ± 0.0100	0.806 ± 0.0056
	SDOST (Ours)	0.360 ± 0.0046	**0.859 ± 0.0075**	**0.891 ± 0.0201**	**0.808 ± 0.0316**	**0.847 ± 0.0107**

4.4 Conclusion

We presented **SDOST**, an online synthetic training framework for annotation-scarce detection of salmon lice in seawater. By using SAM to extract limited instances and dynamically compositing them onto target-free backgrounds, SDOST generates a continuously varying training distribution that outperforms fixed offline datasets. To bridge the synthetic-to-real gap, Feature-Proximal Replay (FPR) revisits synthetic samples with realistic features, while Fast Gradient Sign Method (FGSM) regularization boosts adversarial robustness.

Evaluations across five annotation-scarce settings demonstrate SDOST's consistent superiority over source-only and offline baselines. Notably, SDOST boosts mAP50 to 0.851 (up from 0.684) and F1 to 0.832 (up from 0.658) in **S100-B1070**. Even under extreme scarcity (**S20-B1070**), it maintains a highly competitive 0.787 mAP50 and 0.765 F1 using just 20 annotations, proving its exceptional annotation efficiency.

Future work can explore SDOST's limits with expanded real-world data, evaluate Transformer-based architectures and novel datasets, and develop adaptive synthesis strategies that adjust sampling and replay using real-time detector feedback.

Chapter 5

Automated extraction of scientific statements for integrated assessment of animal welfare using language models

This chapter has been published as:

Zhang, C., da Silva Torres, R., Gansel, L. C., & Bracke, M. (2026). Automated extraction of scientific statements for integrated assessment of animal welfare using language models. *Smart Agricultural Technology*, 13, 101814. https://doi.org/10.1016/j.atech.2026.101814

Abstract

Animal welfare is a key concern in livestock farming, and a semantic modeling approach for welfare assessment is recognized as scientifically grounded. However, extracting supporting scientific evidence still relies heavily on manual review of the literature, limiting scalability and efficiency. To address this challenge, we propose a natural language processing (NLP)-based framework for the automatic extraction and classification of welfare-related scientific statements. To mitigate the lack of negative samples, we adopt a positive-unlabeled (PU) learning strategy enhanced by adversarial contrastive sampling, enabling the automatic extraction of high-confidence negative samples from unstructured documents. Three classification paradigms are evaluated: (1) few-shot learning with large language models (LLMs), (2) a two-stage method combining Transformer-based encoders with traditional classifiers, and (3) end-to-end fine-tuning of pre-trained Transformer models, including BERT, RoBERTa, BGE, and E5. Our experiments reveal that the end-to-end models achieve the best performance, with the E5 reaching 96.0% precision and 92.4% recall on the test set. The adversarial sampling strategy also demonstrates strong robustness across different similarity thresholds. In a challenging scenario where semantic differences between positive and negative statements were small, the BGE-based model outperformed the other models, achieving 59.1% precision and 76.5% recall. To support practical deployment, we also introduce an interactive visualization interface for real-time statement extraction and inspection. Overall, the proposed framework offers a scalable and generalizable solution for extracting welfare-related scientific statements, thereby advancing AI-assisted modeling in animal welfare research. The source code is available at `https://github.com/jay-zc/SM_classification.git` (as of September 2025).

5.1 Introduction

Animal welfare has received growing concern in many industrialised countries, particularly regarding intensive livestock farming. In response, numerous policies, regulations, and certification schemes have been developed to safeguard and promote animal welfare (Ingenbleek et al., 2012).

Animal welfare is a multifaceted concept that remains challenging to quantify objectively (Browning, 2022). The animals' welfare state may be influenced by numerous interrelated factors, including environmental conditions, human-animal interactions, stocking density, nutrition, and health status (Zulkifli, 2013). Different definitions of animal welfare have been proposed. One that captures both physical and emotional states is that welfare can be understood as the quality of life as perceived by the animal itself (Bracke et al., 1999b). However, the assessment of animal welfare is somewhat (inter-)subjective because perceptions of pain, distress, emotional states, and feelings of animals often cannot be measured directly. Consequently, evaluating emotional well-being requires taking into account available scientific knowledge about relationships between environmental conditions and a multitude of behavioural and (patho-)physiological welfare measures. In addition, researchers have developed various frameworks to guide animal welfare assessment, such as the Five Freedoms (Webster, 1994), Three Conceptions of Welfare (Fraser, 2003), Animal Needs Index (Bartussek, 1999), Welfare Quality (Blokhuis, 2007), and the Five Domains Models (Mellor et al., 2020). These frameworks differ in their scientific grounding, comprehensiveness, and practical applicability, which in turn may affect their perceived validity, reliability, and feasibility in various contexts.

Semantic modeling for animal welfare assessment (SemMod-AWA) was introduced by Bracke et al. (2002) as a science-based, formalised and transparent framework for integrated animal-welfare assessment. Using the SemMod-AWA approach, the modeller extracts and decomposes scientific statements from literature and scientific reports to assess the states of the animals' welfare needs. The assessment takes into account environment-based welfare attributes, animal-based welfare measures grouped into weighting categories, and their interrelationships (Collins & Part, 2013). The formalization ensures that the modeling process remains evidence-based, reproducible, and adaptable to emerging research. A distinctive strength of SemMod-AWA lies in its modular and standardized modeling procedure, which typically involves the following steps: (1) establishing the assessment domain; (2) searching for relevant literature on animal welfare; (3) extracting scientific statements pertinent to assess the state of the welfare needs within the domain; (4) decomposing statements into attributes, welfare measures, and weighting categories; (5) assigning scores to weighting cate-

gory levels; (6) defining and ranking attribute levels and calculating their scores; (7) calculating the weighting factor for each attribute; (8) describing and benchmarking housing systems based on their welfare-relevant properties; (9) deriving attribute-level scores for each system based on the inputs; (10) calculating the welfare score for each housing system; (11) benchmarking the scores against other systems, expert opinion, or other relevant sources of knowledge (Bracke et al., 2002; Stien et al., 2013). Although expert opinion may be incorporated, SemMod-AWA is primarily grounded in a systematically reviewed body of the scientific literature, enhancing objectivity and credibility. Over the past two decades, multiple semantic models have been constructed for different species and welfare contexts, including pigs (SOWEL (Bracke et al., 2002), PIGTAIL (Bracke et al., 2004) and RICHPIG (Bracke, 2008)), dairy cattle (COWEL) (Ursinus et al., 2009), laying hens (FOWEL) (De Mol et al., 2006) and fish (SWIM) (Stien et al., 2013; Pettersen et al., 2014; Yildiz et al., 2021). More recently, ANyWEL has been proposed as a generalized SemMod-AWA framework suitable for application to any farm animal (Benthin et al., 2026; Vonholdt-Wenker et al., 2026).

Scientific evidence plays a pivotal role in enhancing the validity, credibility, and interpretability of SemMod-AWA outcomes (Bracke et al., 1999b). However, it also imposes constraints on the model's development and scalability. SemMod-AWA relies heavily on input derived from scientific publications, requiring the modeler to engage in extensive retrieval, manual screening, and extraction of welfare-relevant content. To be suitable for model construction, a broad spectrum of statements must be extracted, covering all resources that are known to affect welfare in the assessment domain, such as the provision of food, water, shelter, space, conspecifics, among others.

The criteria for selecting welfare-relevant statements may vary between modelers and modelling objectives, potentially introducing inconsistency. Figure 5.1 provides a typical example of a welfare-relevant statement (Zonderland et al., 2008; Day et al., 2008). Many different types of statements may be considered valuable for constructing a semantic model to assess animal welfare. Therefore, the selection process demands sentence-level scrutiny and interpretation, often requiring subject-matter expertise. This task is time-consuming and labor-intensive, rendering it impractical to manually process all potentially relevant publications. As a result, modelers typically work with a limited subset of the available literature. The challenge is further augmented by the need for model updates when new research emerges, often necessitating the repetition of an elaborate statement extraction process to maintain the model. In addition, many welfare-relevant statements are implicitly embedded within unstructured text, rather than presented in a standardized or easily identifiable format. Consequently,

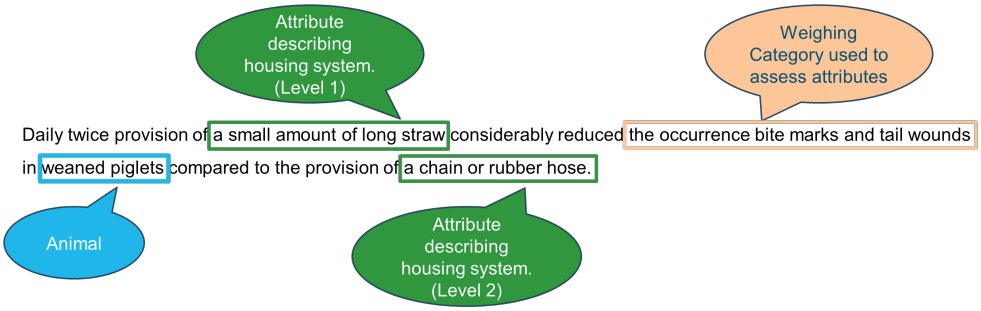


Figure 5.1: Illustrative scientific statement used for semantic modelling of animal welfare assessment (SemMod-AWA), adapted from Zonderland et al. (2008). This statement classifies "enrichment materials" as a welfare-relevant attribute of housing systems, with two levels: "with a small amount of long straw" (Level 1) and "with either a chain or a rubber hose" (Level 2). The phrase "the occurrence of bite marks and tail wounds" functions as the welfare measure, while the comparison indicates a considerable reduction from one enrichment material (straw) compared to two other types of material (chain or hose).

the development of automated tools for extracting statements suitable for modeling from unstructured scientific texts would significantly enhance the scalability and applicability of SemMod-AWA.

Recent advances in natural language processing (NLP) (Hirschberg & Manning, 2015; Chowdhary & Chowdhary, 2020) offer promising avenues for automated extraction and classification of scientific statements related to animal welfare, thereby addressing a key bottleneck in SemMod-AWA. NLP technologies have significantly improved across core tasks such as text classification, named entity recognition (NER) (Ushio & Camacho-Collados, 2022), question answering (Singhal et al., 2025; Namazifar et al., 2021), and summarization (Ghadimi & Beigy, 2022). A major shift occurred with the introduction of the Transformer architecture in 2017, which largely replaced previous recurrent neural network (RNN) approaches as a dominant model in NLP. Among these, Bidirectional Encoder Representations from Transformers (BERT) has demonstrated state-of-the-art performance across numerous NLP benchmarks, owing to its bidirectional self-attention mechanism. Variants such as Sentence-BERT (Reimers & Gurevych, 2019)and KG-BERT (Yao et al., 2019) have extended these capabilities to more specialized tasks, including semantic similarity, relation extraction, and knowledge graph completion. More recently, large language models (LLMs) pre-trained on massive corpora—such as the DeepSeek series—have demonstrated impressive zero- and few-shot capabilities in tasks ranging from summarization to open-domain question answering (Guo et al., 2024; Larionov et al., 2025; Yue, 2025). Unlike earlier models that required extensive fine-tuning, LLMs can be adapted via prompting

to perform a wide array of downstream tasks with minimal or no additional training (Qin et al., 2026). These advancements not only improve general language understanding but also provide powerful tools for domain-specific challenges, such as extracting structured scientific knowledge about animal welfare from unstructured text sources.

The primary objective of this study is to develop an NLP-based framework for the automated extraction and classification of scientific statements from large-scale textual corpora, to support SemMod-AWA for integrated animal welfare assessment. A challenge in this task is the scarcity of fully labeled datasets, particularly the absence of explicit negative samples. To address this, we adopt a positive-unlabeled (PU) learning strategy, leveraging adversarial sampling techniques to construct a high-confidence negative set from unlabeled data. The proposed pipeline begins with sentence-level extraction from unstructured reference documents using MinerU. These candidate samples, alongside manually annotated positive samples, are embedded into dense vector representations using pre-trained sentence encoders. Cosine similarity is then used to score and select candidate negative samples with minimal semantic overlap with the positive samples. To determine the most effective modeling approach, we conduct a comparative evaluation across three modeling paradigms: (1) few-shot learning leveraging the generalization capabilities of LLMs, including DeepSeek V2 (Liu et al., 2024), Gemma3 (Team et al., 2025), Qwen3 (Yang et al., 2025), and Phi-4 (Abdin et al., 2024) with minimal supervision; (2) a hybrid two-stage approach combining transformer-based encoders for feature extraction with conventional machine learning classifiers and optimized via k-fold cross-validation; and (3) an end-to-end transfer learning paradigm involving fine-tuning various pre-trained transformer encoders (including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), BGE (Xiao et al., 2024), and E5 (Wang et al., 2022b)) on our curated dataset. To evaluate the practical utility, the framework is applied to the abstracts of a new set of scientific papers about animal welfare. A domain expert judged the sampled statements to be largely semantically accurate and contextually relevant. This study presents a robust, scalable NLP framework for classifying domain-specific scientific statements under limited supervision, advancing the automation of SemMod-AWA in animal welfare research. The contribution of this work is as follows:

1. It introduces an NLP-based framework for the automated extraction and classification of scientific statements to support semantic modelling for animal welfare assessment (SemMod-AWA);

2. It applies a positive-unlabeled learning strategy with adversarial sampling to address the lack of negative annotations;

Table 5.1: Composition of the statements across six domain-specific semantic models to assess animal welfare.

Semantic model	Species	Assessment domain	Num	Year	Pubs	Reference
SOWEL	Pregnant sows	Overall welfare of existing and new housing and management systems (system level)	90	2001	20	Bracke et al. (2002)
RICHPIG	Weaned and growing-finishing pigs	Welfare value of enrichment materials	679	2004	55	Bracke (2008)
FOWEL	Laying hens	Overall welfare of existing and new housing and management systems (system level)	296	2006	223	De Mol et al. (2006)
COWEL	Dairy cows	Overall welfare of existing and new housing and management systems (system level)	2,342	2009	476	Ursinus et al. (2009)
SWIM	Salmon	Overall welfare of existing and new housing and management systems (cage level)	330	2013	272	Stien et al. (2013)
ANyWEL	Cows, pigs, laying hens and broilers	Welfare potential of existing and new housing systems (system level; assuming good management practices)	442	2024	113	Schneider et al. (2025)

3. It provides a comparative evaluation of three modeling paradigms and is validated through expert review of unseen literature.

5.2 Materials and Methods

5.2.1 Data preparation

5.2.1.1 Positive samples collection

We utilized six different semantic models specifically developed for the assessment of animal welfare. These were SOWEL, FOWEL, COWEL, RICHPIG, SWIM, and ANYWEL, each targeting a particular animal species on a specific research domain. During the development of these models, the statements were curated from scientific publications and manually annotated by experts. We compiled these annotated statements into a unified positive sample corpus, denoted as $\mathcal{P}$, containing 4179 expert-validated instances (Table 5.1).

5.2.1.2 Adversarial sampling for negative sample generation

Constructing negative samples posed a greater challenge than positive samples, as only the latter were explicitly annotated. To address this, we employed an adversarial sampling strategy designed to extract high-confidence negative examples from unlabeled text within the same domain. Adversarial sampling operates under the principle of identifying maximally dissimilar instances relative to the positive class within the unlabeled data. This approach facilitates the learning of discriminative feature representations by exposing the model to oppositional instances near the decision boundary. Specifically, negative samples were drawn exclusively from the same set of reference publications used to construct the corresponding positive samples, thereby ensuring consistency in linguistic style and domain-specific context and minimizing potential domain shift.

The sample selection pipeline consists of three stages, as illustrated in Figure 5.2. In the first stage, we employed MinerU (Wang et al., 2024b) to process the reference documents (PDFs) and extract serialized sentences. These raw sentences were then subjected to text normalization to remove non-propositional or noisy content, such as DOIs, citation strings, and very short fragments (fewer than 10 tokens) - the sentence cleaning step in the figure.

In the second stage, we computed the cosine between each candidate sample and all positive samples. Each sample was encoded using the pre-trained MiniLM-L12 model (Sentence-Transformers, n.d.) with embedding dimension $d = 384$. For each candidate sample s_j, its embedding $\mathbf{e}_j$ undergoes pairwise cosine similarity computation against all positive samples in $\mathcal{P}$. The maximum similarity score for s_j is calculated as:

$$\mathrm{sim}_{\max}(s_j) = \max_{i \in \mathcal{P}} \cos(\mathbf{e}_j, \mathbf{e}_i) = \max_{i \in \mathcal{P}} \left(\frac{\mathbf{e}_j \cdot \mathbf{e}_i}{\|\mathbf{e}_j\| \|\mathbf{e}_i\|} \right) \tag{5.1}$$

where $\mathcal{P} \triangleq \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ denotes the set of positive sample embeddings. Here, $\|\mathbf{e}\| \triangleq \sqrt{\sum_{k=1}^{d} e_k^2}$ represents the ℓ_2-norm, and $\mathbf{a} \cdot \mathbf{b} \triangleq \sum_{k=1}^{d} a_k b_k$ specifies the vector dot product operation. Candidate sentences exhibiting minimal $\mathrm{sim}_{\max}(s_j)$ values are selected as high-confidence negatives, as their feature representations exhibit maximal divergence from the positive class.

Each candidate sample yields a vector of similarity scores with all positive samples. We retain only the maximum similarity score per candidate, which reflects its closest proximity to any known positive.

In the third stage, two selection strategies were implemented based on the computed

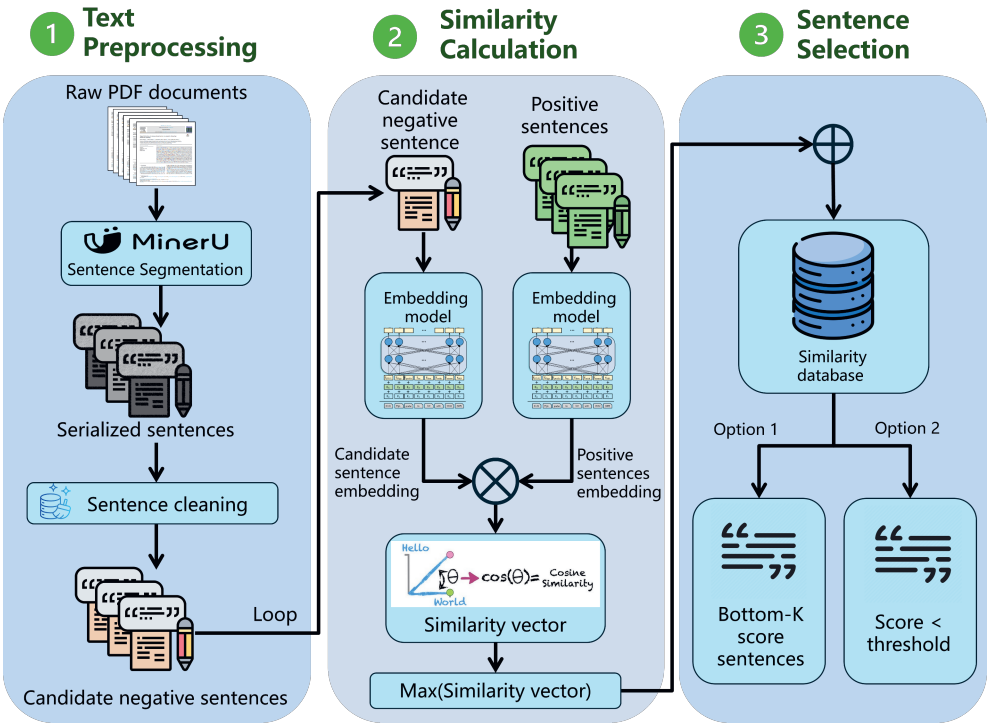


Figure 5.2: Negative sample selection pipeline via adversarial sampling. The negative sampling pipeline involves three stages: Sentence extraction and filtering via MinerU, cosine similarity-based divergence scoring using MiniLM embeddings, and bottom-K selection of semantically distant candidates as high-confidence negatives.

similarity scores. The bottom-K method selects a fixed number of candidates with the lowest similarity scores, allowing for direct control over dataset size. Alternatively, the threshold-based method selects all candidates below a predefined similarity cutoff, favoring confidence over count. For controlled dataset construction, we adopted the bottom-K strategy in our experiments.

5.2.1.3 Dataset construction

Table 5.2 provides an overview of the final dataset composition. For each semantic-model sub-corpus, a labeled subset was constructed by pairing manually annotated positive statements with an equal number of adversarially sampled negative instances, resulting in class-balanced subsets (e.g., $\mathcal{D}_{\text{FOWEL}}$). These six labeled subsets were subsequently merged to form the complete dataset used for model training and evaluation. The complete dataset $\mathcal{D}_{\text{ALL}}$ is defined as the union of all labeled subsets:

Table 5.2: Composition of the final dataset by semantic-model subcorpus, each containing both positive statements and an equal number of adversarially sampled negatives from the same source publications.

Set	Subset	Positive	Negative	Total	Year
Training	$\mathcal{D}_{\text{FOWEL}}$	296	296	592	2006
	$\mathcal{D}_{\text{COWEL}}$	2,342	2,342	4,684	2009
	Subtotal	2,638	2,638	5,276	–
Test	$\mathcal{D}_{\text{SOWEL}}$	90	90	180	2001
	$\mathcal{D}_{\text{RICHPIG}}$	679	679	1,358	2004
	$\mathcal{D}_{\text{SWIM}}$	330	330	660	2013
	$\mathcal{D}_{\text{ANYWEL}}$	442	442	884	2024
	Subtotal	1,541	1,541	3,082	–
Total	$\mathcal{D}_{\text{ALL}}$	4,179	4,179	8,358	–

$$\mathcal{D}_{\text{ALL}} = \bigcup_{i=1}^{6} \mathcal{D}_i = \mathcal{D}_{\text{FOWEL}} \cup \mathcal{D}_{\text{COWEL}} \cup \mathcal{D}_{\text{SOWEL}} \cup \mathcal{D}_{\text{RICHPIG}} \cup \mathcal{D}_{\text{SWIM}} \cup \mathcal{D}_{\text{ANYWEL}} \tag{5.2}$$

To promote domain-level generalisation and ensure robustness, we strictly partitioned datasets by source model to avoid overlap between training and test sets. Each labeled subset, therefore, consists of positive and negative statements originating from the same set of publications, enabling both semantic traceability and structural consistency across labels.

In total, the final dataset contains 8,358 instances, with 63.1% assigned to the training set and 36.9% to the test set. The test set comprises exclusively those labeled subsets drawn from semantic models not present in the training data, thereby establishing a robust framework for evaluating out-of-domain generalization.

5.2.2 LLM-based few-shot learning classification

To facilitate binary classification tasks using LLMs, we developed a prompt-based framework utilizing DeepSeek V2 (Liu et al., 2024), Gemma3 (Team et al., 2025), Qwen3 (Yang et al., 2025), and Phi-4 (Abdin et al., 2024) as the foundational models (Table 5.3). These LLMs, trained on extensive textual corpora, demonstrate strong capabilities in natural language understanding and generation, enabling them to perform complex tasks, such as information extraction and classification, without

Table 5.3: Information of LLMs used for the classification of welfare-related statements.

Model	Parameters	Context Window	Release	Publisher	Model Type	Reference
Qwen3	14B	40k	2025	Alibaba	Dense	(Yang et al., 2025)
DeepSeek-V2	16B	4k	2024	DeepSeek	MoE	(Liu et al., 2024)
Gemma3	12B	128k	2025	Google	Dense	(Team et al., 2025)
Phi-4	14B	16k	2024	Microsoft	Dense	(Abdin et al., 2024)

requiring additional fine-tuning. Our proposed prompt framework, depicted in Figure 5.3, adopts a few-shot learning approach and consists of five key components: (1) Role definition: Defines the model's role as an "animal welfare statement classifier" and clearly delineates the scope of the classification task; (2) Classification criteria: Introduces three required components for a valid welfare-related statement (attribute, comparison, and response) (3) Demonstration with examples: Presents two positive examples accompanied by detailed breakdown of the three required components to guide model reasoning; (4) Few-shot Learning and improved generalization: Provides six labeled instances to calibrate decision boundaries; (5) Instruction&Output formatting: Utilizes a standardized placeholder (text) to structure input prompts and constrains the output to a binary format ("yes" / "no") to reduce response variability and enhance interpretability.

5.2.3 Two-stage classification

The two-stage classification framework entails first encoding input samples using pre-trained embedding models, followed by applying machine learning classifiers to perform label prediction based on the resulting representations. Unlike LLM-based few-shot learning approaches, which operate in an end-to-end inference paradigm, the two-stage approach requires a training phase. However, the training process occurs solely at the second stage, where the classifier is trained while the embedding model remains fixed.

5.2.3.1 Feature extraction

As shown in Table 5.4, we employed four pre-trained Transformer-based embedding models, BERT, RoBERTa, E5, and BGE, as fixed feature extractors to generate dense semantic representations of input text. Owing to their pre-training on an extensive volume of corpus, those models possess strong capabilities in capturing semantic and contextual nuances. To preserve their pre-trained knowledge and ensure computational efficiency, all models are utilized in a frozen configuration, meaning their parameters remain unchanged during downstream training.

To maintain methodological consistency across models, we adopt a unified feature

Role Definition	You are a text classifier. Your task is to determine if the given text belongs to an animal welfare related scientific statement.
Classification Criteria	An animal welfare related scientific statement typically describes the relationship between an animal's welfare and the external environmental circumstances, including management practices. Scientific statements must meet the following criteria: 1. They must clearly define the attributes that affect welfare, such as environmental attributes, management attributes, etc. 2. They must compare different levels or intensities of these attributes 3. They must describe the animal's response or preference to different levels of the attributes
Demonstration with Examples	Example 1: "Pigs with straw have the opportunity to express more different behavioral elements compared with pigs with no straw and they actually did express more diverse behavior" - Attribute: Straw - Levels: With straw vs. without straw - Response: Expressing more (or less) different behavioral elements Example 2: "Hens reared without perches have difficulty using perches owing to low muscle strength, a lack of motor skills and the inability to keep balance, or they have impaired spatial skills, which are necessary for moving around in three-dimensional space" - Attribute: Reared with or without perches - Levels: Reared with perches vs. reared without perches - Response: Different physical and spatial abilities
Few-shot Learning	Few-shot examples: 1. "To prevent leg injuries, the inner radius fence in a race should have a 60 cm high solid panel at the bottom." (yes) 2. "The activity of each pig was assessed in an open field test during Week 11 of treatment." (no) 3. "This estimate is normally referred to for stocking and legislation purposes." (no) 4. "The oxygen levels in salmon sea cages are negatively correlated with stocking density." (yes) 5. "A number of studies have reported reduced feather pecking as a result of adding some type of enrichment to the pen or cage." (yes) 6. "Temperament scores evaluated for each breed group also showed that increased temperament score resulted in decreased average daily gains." (yes)
Instruction & Output Formatting	Now classify this text: Text: {text} Answer with only 'yes' or 'no' without any other text.

Figure 5.3: LLM prompt design for classifying animal welfare statements. The prompt consists of role definition, structured criteria, example demonstrations, and binary output formatting. The examples in the prompt are derived from (Porter et al., 1999; Jonsson et al., 2003; Channing et al., 2001).

extraction protocol based on attention-mask-aware mean pooling over the token embeddings from the final encoder layer. Specifically, after tokenization and encoding, we compute the mean of the token embeddings along the sequence dimension, weighing each token's contribution by the corresponding attention mask. This ex-

Table 5.4: Information of embedding models used for feature extraction.

Model	Key Features	Layers	Hidden Size	Params	Vocab. Size	Reference
BERT	Bidirectional, [CLS], MLM+NSP	12	768	110M	30,522	(Devlin et al., 2019)
RoBERTa	Dynamic masking, no NSP	12	768	125M	50,265	(Liu et al., 2019)
E5	Contrastive learning	12	768	110M	30,000	(Wang et al., 2022b)
BGE	English text embeddings	12	768	110M	32,000	(Xiao et al., 2024)

Note: BERT: Bidirectional Encoder Representations from Transformers; RoBERTa: Robustly optimized BERT approach; E5: Embeddings from bidirEctional Encoder rEpresentations; BGE: BAAI General Embedding; [CLS]: Classification token; MLM: Masked Language Modeling; NSP: Next Sentence Prediction; Params: Model parameters (M = millions); Vocab. Size: Number of unique tokens in the vocabulary of the tokenizer.

cludes padding tokens from influencing the final representation. The resulting 768-dimensional sentence embeddings from all four models are used as standardized input features for downstream classification tasks, ensuring consistency and comparability across embedding methods.

5.2.3.2 ML classifiers training with cross-validation

To classify these static feature representations, we employed a diverse set of machine learning (ML) classifiers, including Support Vector Machines (SVM) (Boser et al., 1992), XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), Random Forest (RF) (Breiman, 2001), Logistic Regression (LR) (Cox, 1958), k-Nearest Neighbors (KNN) (Guo et al., 2003), Naive Bayes (NB) (Maron, 1961), and neural networks (NN) (McCulloch & Pitts, 1943). The training process was conducted using stratified 5-fold cross-validation (StratifiedKFold) to ensure robust performance estimation while preserving class distribution. In each iteration, 80% of the data (four folds) was used for training, while the remaining 20% (one fold) served as the validation set. This process was repeated five times, ensuring that all data points were used for both training and validation. To further optimize classifier performance, we applied hyperparameter tuning using random search cross-validation (RandomSearchCV) over predefined parameter spaces (see Table 5.5). Accuracy was used as the primary evaluation metric, while precision, recall, and F1-score were monitored to maintain balanced performance across all classes. The entire training pipeline, including cross-validation and hyperparameter optimization, was implemented using the scikit-learn framework.

5.2.4 End-to-end classification

In contrast to the two-stage classification approach, an alternative strategy involves fine-tuning pre-trained language models directly on the target dataset in an end-to-end manner. This paradigm eliminates the need for intermediate feature extraction

Table 5.5: Hyperparameter Optimization Configurations for each ML classifier.

Classifier	Parameters and Search Space
SVM	C: {0.1, 1.0, 5.0}, kernel: {linear, rbf}, gamma: {scale, auto}
Random Forest	n_estimators: {100, 200}, max_depth: {10, 20}, min_samples_split: {2, 6}
XGBoost	n_estimators: {100, 200}, max_depth: {3, 6}, learning_rate: {0.01, 0.1}, subsample: {0.8, 1.0}, colsample_bytree: {0.8, 1.0}
LightGBM	n_estimators: {100, 200}, max_depth: {3, 6}, learning_rate: {0.01, 0.1}, num_leaves: {20, 40}, subsample: {0.8, 1.0}
K-NN	n_neighbors: {3, 7, 11}, weights: {uniform, distance}, metric: {euclidean, cosine}
Naive Bayes	var_smoothing: {1e-9, 1e-7}
Logistic Regression	C: {0.1, 1.0, 10.0}, penalty: l2 , solver: saga

by jointly optimizing representation within a unified training process.

We continue to utilize the same four Transformer-based models, BERT, RoBERTa, E5, and BGE, as the backbone encoders. Each input sentence is first tokenized using the corresponding model-specific tokenizer, with a standardized configuration: a maximum sequence length of 512 tokens, where longer texts are truncated and shorter ones are padded. The tokenized input is then passed through the encoder to generate contextualized token embeddings, each represented as a 768-dimensional vector. To enable classification, we attach a task-specific multi-layer classification head to each model. For BERT, RoBERTa, and BGE, we use the [CLS] token embedding from the final encoder layer as the aggregated sentence representation. For E5, we instead

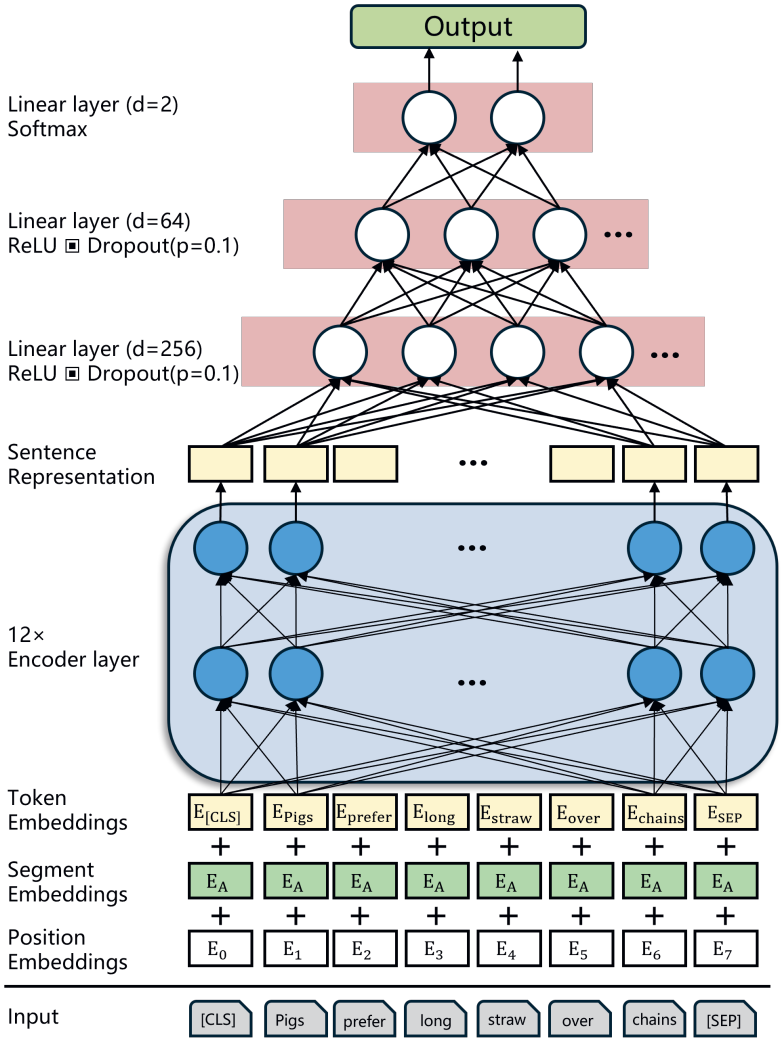


Figure 5.4: Architecture of the classification framework shared across all models. "Sentence Representation" refers to the aggregated embedding obtained from the encoder: Using the [CLS] token for BERT, RoBERTa, and BGE, and mean pooling for E5. This representation is fed into a multi-layer classifier for downstream prediction.

apply mean pooling over all token embeddings, following its original architecture. The resulting representation is first passed through a dropout layer to reduce overfitting, and then through a three-layer feedforward neural network: a linear transformation from 768 to 256 dimensions with ReLU activation and dropout, followed by a 256 to 64 dimensional layer with identical structure, and concluding with a 64 to 2 dimensional output layer. This architecture outputs a two-dimensional logit vector, corresponding to the predicted probability distribution over the binary classification labels.

Table 5.6: Statistics of the $\mathcal{D}_{\text{ABS}}$ dataset, constructed from 180 abstracts using alternative selection criteria.

Subset	Positive	Negative	Total
Train	123	1198	1321
Validation	30	301	331
Test	34	379	413

During fine-tuning, we employ the AdamW optimizer, a variant of Adam that decouples weight decay from the gradient update to ensure stable and efficient convergence. All the models were trained using the cross-entropy loss function, which is widely used for classification tasks. The cross-entropy loss is computed as follows:

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{5.3}$$

where N is the number of samples, $y_i \in \{0, 1\}$ is the ground truth label of the i-th sample, and $\hat{y}_i \in (0, 1)$ is the predicted probability for the positive class after sigmoid activation.

5.2.5 Expert validation

To further assess the robustness and adaptability of our framework in a related but distinct application domain, a domain expert constructed a new expert-annotated dataset, $\mathcal{D}_{\text{ABS}}$, derived from the 180 abstracts (of 182 scientific articles). These articles were selected as part of a literature search on novel animal-welfare measures, particularly in the emerging area of positive welfare. As summarized in Table 5.6, this dataset serves as an independent benchmark, distinguished from the original training set by its distinct annotation guideline.

Unlike the training data, where only positive instances were labeled, $\mathcal{D}_{\text{ABS}}$ includes explicit annotations for both positive and negative statements. This required deeper domain expertise and greater reliance on logical inference. Many positive statements in this dataset express welfare relevance implicitly or through loosely structured language, resulting in narrower semantic margins and increased classification difficulty.

We fine-tuned all models identified by $\mathcal{D}_{\text{ALL}}$ on this dataset. This setup enables a focused evaluation of the model's practical utility and transferability under more complex annotation standards, ambiguous and evolving semantic conditions.

5.2.6 Robustness to negative statements

To rigorously evaluate the model's robustness to variations in negative sample selection strategies, we conducted a sensitivity analysis on the similarity threshold used during adversarial sampling. The experiment utilized the same set of positive training instances as in the $\mathcal{D}_{\text{ALL}}$. By systematically varying the similarity threshold from 0.3 to 0.9 (option 2 in the sentence selection procedure, see Figure 5.2), we constructed a series of training datasets that differed markedly in both the quantity and quality of negative samples. Each dataset represented a distinct training environment, potentially contaminated by varying degrees of label noise. For each of these environments, an independent model was fine-tuned and subsequently evaluated on the same held-out test set from the $\mathcal{D}_{\text{ALL}}$. By analyzing the variation in F1 scores across these training conditions, we were able to quantitatively assess the model's resilience to bias and label noise induced by uncertainty in the negative sampling process.

5.2.7 Experiment environment and evaluation metrics

All the models were run on a Linux-based Pop!_OS operating system, equipped with an A5500 GPU with 16GB of memory, an Intel i9-13950HX CPU, and 32GB of RAM.

All three classification approaches were evaluated on the test set to assess their generalization ability. The evaluation utilized four standard classification metrics: accuracy, precision, recall, and F1-score. Accuracy reflects the overall proportion of correctly classified samples. Precision measures the proportion of all positive predictions that are truly positive. Recall measures the proportion of all actual positive cases that are correctly predicted as positive. F1-score represents the harmonic mean of precision and recall, providing a balanced assessment in the presence of class imbalance.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.4}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.5}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.6}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.7}$$

where TP (true positives) refers to the number of instances that are actually positive and are correctly predicted as positive by the model. FP (false positives) denotes the number of instances that are actually negative but are incorrectly classified as positive. TN (true negatives) represents the number of negative instances that are correctly identified as negative, and FN (false negatives) corresponds to the number of positive instances that are incorrectly predicted as negative.

5.2.8 Interactive tool for animal-welfare statement extraction

To operationalise the proposed framework, we built an interactive, web-based application (Fig. 5.5). Users can upload PDF articles, which are parsed by the MinerU document-analysis engine to recover clean text and layout. The tool supports both the end-to-end and the two-stage models described above to identify candidate statements.

Predicted positive statements are highlighted in situ within the document together with their confidence scores. Results can be filtered and exported with a one-click action to facilitate downstream analysis.

The application is built with Streamlit (Inc., 2019) and uses PyTorch with HuggingFace Transformers for sentence-level inference. The design supports batch processing and is suitable for large-scale literature screening.

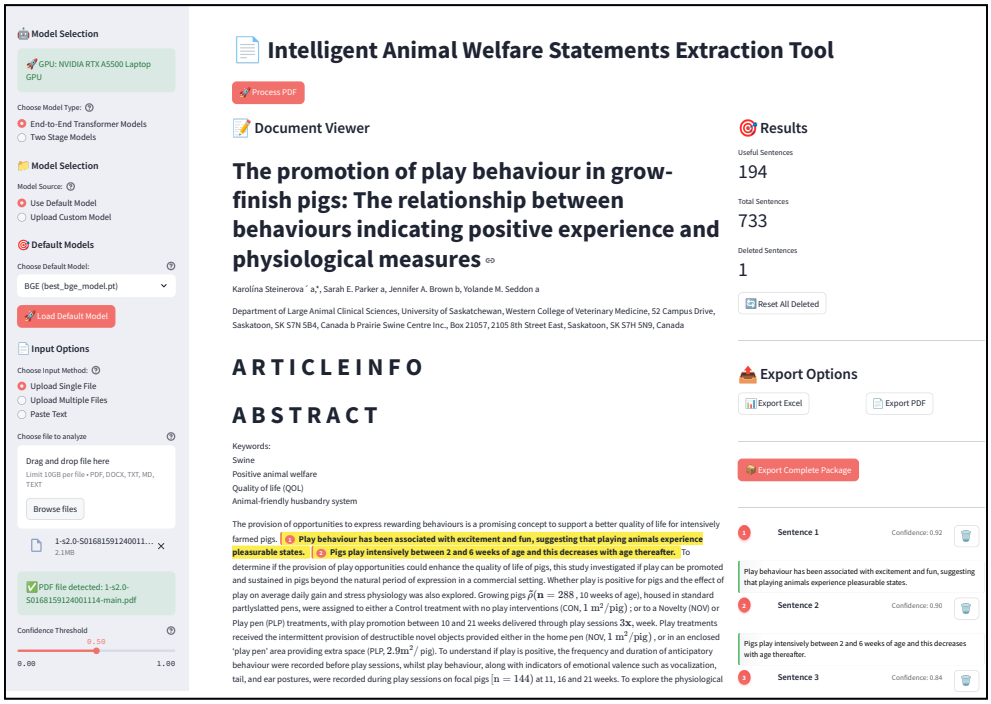


Figure 5.5: Interactive, GPU-accelerated tool for extracting scientific animal-welfare statements from unstructured documents. The left pane provides model selection (end-to-end or two-stage), GPU/status indicators, and input options for single or multiple files or direct paste. The central viewer highlights sentences predicted as positive within the document, while the right panel lists extracted sentences with confidence scores and quick actions; the header offers one-click export to Excel/PDF and a summary of useful, total, and deleted sentences.

5.3 Results

5.3.1 Few-shot learning classification

Table 5.7 presents a comparative evaluation of LLMs on $\mathcal{D}_{\text{ALL}}$ dataset under a few-shot learning setting. Overall, all models demonstrated good precision, with Gemma3 achieving the highest precision of 92.1%. Both DeepSeek-V2 and Phi-4 followed closely, with precision of 89.4% and 89.8%, respectively. In contrast, Qwen3 exhibited the lowest precision of 81.8%.

However, the recall across all models was generally lower, suggesting that a relatively large portion of positive samples were misclassified as negative. Among the models, Phi-4 achieved the highest recall (75.9%), and Qwen3 the lowest (46.9%). This lower recall reduced the F1-score (59.3%) and the accuracy (68.2%) of Qwen3, which were

Table 5.7: Performance comparison of few-shot learning methods on the $\mathcal{D}_{\text{ALL}}$.

Model	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
DeepSeek-V2	89.4	64.6	75.0	78.4
Gemma3	92.1	69.2	79.0	81.6
Qwen3	81.8	46.9	59.3	68.2
Phi-4	89.8	75.9	82.2	83.6

Table 5.8: Prediction examples from various LLMs. In Sample 1, the welfare impact of isolation versus social interaction on a cow is shown (isolation of this animal leads to stress). Sample 2 illustrates the preference of a laying hen for nest selection.

ID	Ground truth	DeepSeek-V2	Gemma3	Qwen3	Phi-4
Sample 1	1	1	1	0	1
Sample 2	1	1	0	0	0

Full statements:
Sample 1: However, cows isolated from other cows at calving may be stressed by being completely left alone (EFSA, 2009).
Sample 2: A further reason for inconsistency in nest choice could be that laying hens are gregarious in egg laying and choose to lay their eggs in the company of other hens rather than alone (Stämpfli et al., 2012).

the lowest among all evaluated models.

Overall, Phi-4 outperformed the other models by achieving the highest F1-score (82.2%) and accuracy (83.6%), reflecting a more balanced trade-off between precision and recall. Although Gemma3 led in precision, its lower recall resulted in slightly reduced overall performance in terms of F1-score and accuracy compared to Phi-4. Table 5.8 presents prediction examples from various LLMs. Sample 1 shows the welfare impact of isolation versus social interaction on a cow (isolation leads to stress), and only Qwen3 incorrectly classifies the sample. Sample 2 illustrates the preference of a laying hen for nest (egg-laying site) selection. In this case, Gemma3, Qwen3, and Phi-4 all misclassify the input as a negative example.

5.3.2 Two-stage classification

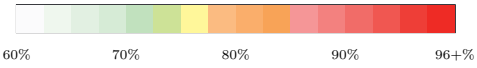
Table 5.9 presents the performance of various feature extraction models in combination with different classifiers. Overall, most configurations exhibited strong precision, typically exceeding 85.0%, although none achieved a recall above 90.0%. The highest precision, 94.7%, was attained by the BERT–SVM combination. In terms of overall performance, BERT combined with LR delivered the best results, achieving a precision of 94.3%, recall of 88.7%, accuracy of 91.7%, and an F1-score of 91.4%.

As shown in Table 5.9 and Figure 5.6, the SVM yielded the strongest classification

Table 5.9: Performance comparison of different embeddings and classifiers.

	Classifier									
Embedding	KNN	LGB	LR	NB	NN	RF	SVM	XGB	Mean	SD
Precision										
BERT	89.4%	92.5%	94.3%	90.1%	91.7%	91.4%	94.7%	92.6%	92.1%	1.7%
BGE	83.8%	92.2%	91.8%	86.5%	91.0%	90.5%	93.5%	91.6%	90.1%	3.1%
E5	89.1%	92.6%	94.2%	90.7%	93.7%	92.8%	93.0%	93.6%	**92.5%**	1.6%
RoBERTa	91.0%	92.4%	92.3%	87.2%	93.1%	90.4%	94.1%	91.4%	91.5%	2.0%
Mean	88.3%	92.4%	93.2%	88.6%	92.4%	91.3%	**93.8%**	92.3%	-	-
SD	2.7%	0.1%	1.1%	1.8%	1.1%	1.0%	0.6%	0.9%	-	-
Recall										
BERT	82.5%	83.2%	88.7%	72.7%	86.5%	76.9%	83.8%	82.2%	82.1%	4.8%
BGE	83.1%	68.8%	71.7%	68.7%	77.7%	66.8%	72.4%	70.3%	72.4%	5.1%
E5	83.1%	74.0%	77.3%	71.8%	77.3%	73.7%	81.4%	73.5%	76.5%	3.8%
RoBERTa	87.9%	87.6%	87.3%	80.1%	84.7%	85.1%	86.4%	87.2%	**85.8%**	2.4%
Mean	**84.2%**	78.4%	81.3%	73.3%	81.6%	75.6%	81.0%	78.3%	-	-
SD	2.2%	7.4%	7.1%	4.2%	4.1%	6.6%	5.3%	6.7%	-	-
Accuracy										
BERT	86.4%	88.2%	91.7%	82.4%	89.3%	84.8%	89.5%	87.8%	87.5%	2.7%
BGE	83.6%	81.5%	82.6%	79.0%	85.0%	79.9%	83.7%	81.9%	82.2%	1.9%
E5	86.5%	84.1%	86.3%	82.3%	86.1%	84.0%	87.7%	84.2%	85.1%	1.7%
RoBERTa	89.6%	90.2%	90.0%	84.2%	89.2%	88.0%	90.5%	89.5%	**88.9%**	1.9%
Mean	86.5%	86.0%	87.6%	82.0%	87.4%	84.2%	**87.9%**	85.9%	-	-
SD	2.1%	3.4%	3.5%	1.9%	1.9%	2.9%	2.6%	3.0%	-	-
F1-score										
BERT	85.9%	87.6%	91.4%	80.5%	89.0%	83.5%	88.9%	87.1%	86.7%	3.2%
BGE	83.5%	78.8%	80.5%	76.6%	83.9%	76.9%	81.6%	79.5%	80.2%	2.6%
E5	86.0%	82.3%	85.0%	80.2%	84.7%	82.1%	86.9%	82.3%	83.7%	2.1%
RoBERTa	89.4%	89.9%	89.7%	83.5%	88.7%	87.7%	90.1%	89.2%	**88.5%**	2.0%
Mean	86.2%	84.7%	86.7%	80.2%	86.6%	82.5%	**86.9%**	84.5%	-	-
SD	2.1%	4.4%	4.3%	2.4%	2.3%	3.9%	3.3%	3.8%	-	-

Note: SD stands for standard deviation.



performance among all models. Its mean precision was 93.8%, and its recall was 81.0%. Although this recall was not the highest, SVM achieved the highest accuracy and F1-score, at 87.9% (SD = 2.6%) and 86.9% (SD = 3.3%), respectively. Both LR and NN yielded comparable results. Their mean precision exceeded 92.0%, mean recall was above 81.0%, mean accuracy was above 87.0%, and mean F1-score was above 86.0%. KNN achieved the highest recall, with a mean of 84.2% (SD = 2.2%) among all classifiers, but its precision was the lowest, at 88.3% (SD = 2.7%). The LGB model showed the most stable precision, ranging from 92.2% to 92.6% (SD = 0.1%). Its recall, however, was more variable and somewhat lower, with a mean of 78.4% (SD = 7.4%). NB performed the worst of all models. Its precision was 88.6% (SD = 1.8%), slightly higher than KNN, and its mean recall was 73.3% (SD = 4.2%),

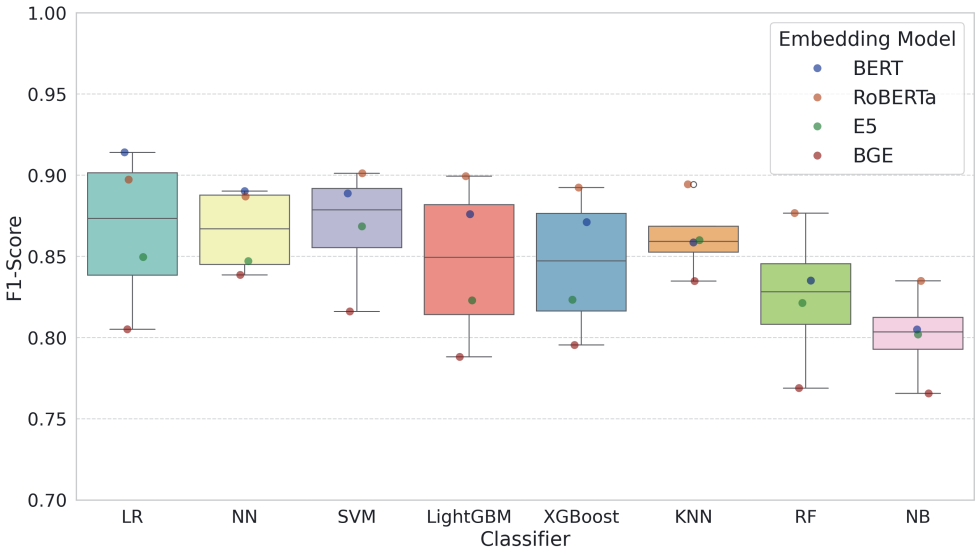


Figure 5.6: F1-score distribution of various classifiers on the test set of $\mathcal{D}_{\text{ALL}}$ under different embedding feature extractors.

the lowest value observed. As a result, NB also had the lowest accuracy and F1-score, at 82.0% (SD = 1.9%) and 80.2% (SD = 2.4%), respectively.

Regarding the embedding methods, as presented in Table 5.9 and Figure 5.7, all embedding models achieved a mean precision exceeding 90%, with mean recall values ranging from 72.4% to 85.8%. Overall, the feature extraction approach based on RoBERTa outperformed the other models. Specifically, RoBERTa achieved the highest mean recall of 85.8% (SD = 2.4%). Although its precision was 91.5% (SD = 2.0%), slightly lower than that of E5 and BERT, it yielded the highest mean accuracy (mean = 88.9%, SD = 1.9%) and F1-score (mean = 88.5%, SD = 2.0%). BERT ranked second, delivering competitive results with a mean accuracy of 87.5% (SD = 2.7%) and an F1-score of 86.7% (SD = 3.2%). E5 achieved the highest precision at 92.5% (SD = 1.6%). However, its recall was only 76.5% (SD = 3.8%), resulting in lower accuracy and F1-scores compared to RoBERTa and BERT. BGE exhibited the weakest performance across all metrics, with a mean precision of 90.1% (SD = 3.1%), a mean recall of 72.4% (SD = 5.1%), and mean accuracy and F1-scores of 82.2% (SD = 1.9%) and 80.2% (SD = 2.6%), respectively.

5.3.3 End-to-end classification

Compared to LLM-based few-shot learning methods and two-stage approaches, the end-to-end method consistently demonstrated superior performance in overall F1 and accuracy. Table 5.11 presents the performance comparison of four pre-trained lan-

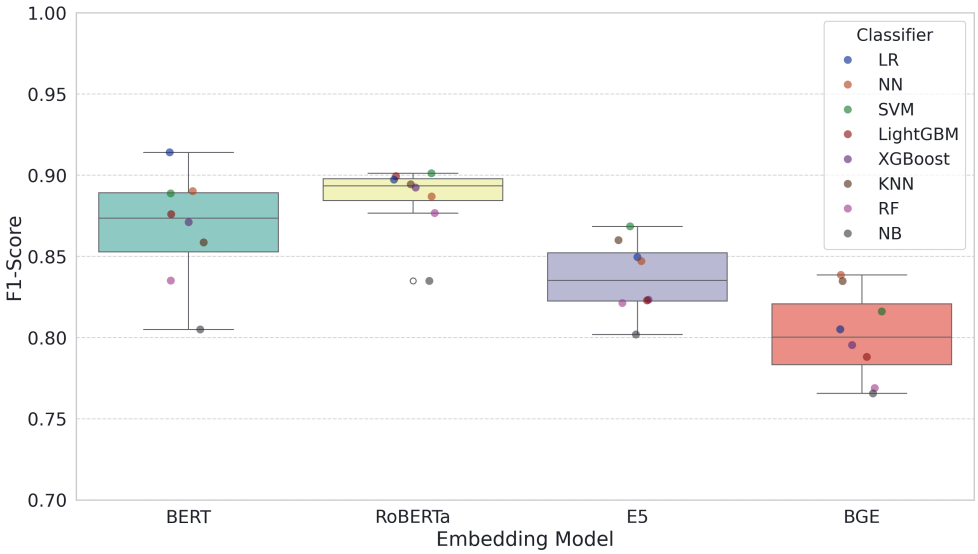


Figure 5.7: F1-score distribution of various embedding feature extractors across different classifiers on the test set of $\mathcal{D}_{\text{ALL}}$

guage models fine-tuned on the training set. Notably, all models achieved strong results, with accuracy exceeding 92.0%, precision above 92.0%, recall surpassing 90.0%, and F1-scores above 92.0%. Table 5.10 presents representative predictions made by each model on the dataset. RoBERTa and E5 achieved correct classifications on samples 1, 2, and 3. In contrast, all models misclassified the negative sample 4 as positive.

Among the evaluated models, E5 exhibited the best overall performance, achieving the highest accuracy (94.3%), precision (96.0%), and F1-score (94.2%). However, its recall (92.4%) was slightly lower than that of BERT, which attained the highest recall at 95.5%. Despite this strength, BERT's lower precision resulted in a slightly reduced F1-score. RoBERTa demonstrated precision comparable to E5 (96.0%), but its recall was the lowest among all models (90.1%), leading to a slightly lower F1-score. BGE showed consistently competitive performance across all metrics but remained marginally behind E5 in every aspect.

Table 5.12 presents the performance of the models on each subset. All models achieved superior results on the $\mathcal{D}_{\text{ANYWEL}}$ subset, with precision at least 98.0%, recall above 95.0%, and both F1-score and accuracy at least 97.0%. BERT, RoBERTa, and BGE exhibited the lowest performance on the $\mathcal{D}_{\text{RICHPIG}}$ subset. Their precision values were 87.4%, 94.8%, and 90.4%, respectively, while their recall values were 94.0%, 86.5%, and 90.0%. In contrast, E5 performed worst on the $\mathcal{D}_{\text{SOWEL}}$ subset, with a precision

Table 5.10: Prediction examples from end-to-end models. Samples 1, 2, and 3 are positive cases, each reporting findings on how environmental or management factors affect animal welfare. In contrast, Sample 4 does not describe any specific empirical findings.

ID	Ground truth	BERT	RoBERTa	E5	BGE
Sample 1	1	1	1	1	1
Sample 2	1	1	1	1	1
Sample 3	1	0	1	1	0
Sample 4	0	1	1	1	1

Full statements:

Sample 1: However, cows isolated from other cows at calving may be stressed by being completely left alone (EFSA, 2009, p. 133).

Sample 2: A further reason for inconsistency in nest choice could be that laying hens are gregarious in egg laying and choose to lay their eggs in the company of other hens rather than alone (Stämpfli et al., 2012, p. 588).

Sample 3: The earth trough was of great interest to the weaned piglets. New earth was used by all piglets before feeding (when presented at the same time) (Wood-Gush & Beilharz, 1983).

Sample 4: Social status and resistance to disease in house mice (Mus musculus) status-related modulation of hormonal responses in relation to immunity costs in different social and physical environments (Barnard et al., 1996).

Table 5.11: Performance comparison of fine-tuned models on the test set.

Model	Accuracy	Precision	Recall	F1-Score
BERT	93.6%	92.1%	**95.5%**	93.8%
RoBERTa	93.2%	96.0%	90.1%	92.9%
E5	**94.3%**	**96.0%**	92.4%	**94.2%**
BGE	92.9%	93.6%	92.2%	92.9%

of 92.8% and a recall of 85.6%. Its accuracy and F1-score were 89.4% and 89.0%, respectively. For the $\mathcal{D}_{\text{SWIM}}$ subset, which was the only subset related to fish welfare, all models yielded satisfactory results. Precision ranged from 94.2% to 95.2%, recall from 88.5% to 93.9%, accuracy from 91.5% to 94.4%, and F1-score from 91.3% to 94.4%.

These results suggest that E5 was the most balanced and robust model for classifying scientific statements that can be used in SemMod-AWA.

Table 5.12: Performance comparison of fine-tuned models on different subsets of the dataset.

	Subset	Accuracy	Precision	Recall	F1-Score
BERT	$\mathcal{D}_{\text{ANYWEL}}$	98.4%	98.0%	99.0%	98.4%
	$\mathcal{D}_{\text{RICHPIG}}$	90.2%	87.4%	94.0%	90.6%
	$\mathcal{D}_{\text{SOWEL}}$	93.3%	90.6%	96.7%	93.6%
	$\mathcal{D}_{\text{SWIM}}$	94.4%	94.8%	93.9%	94.4%
RoBERTa	$\mathcal{D}_{\text{ANYWEL}}$	97.1%	98.4%	95.7%	97.0%
	$\mathcal{D}_{\text{RICHPIG}}$	90.9%	94.8%	86.5%	90.5%
	$\mathcal{D}_{\text{SOWEL}}$	91.1%	95.1%	86.7%	90.7%
	$\mathcal{D}_{\text{SWIM}}$	93.2%	95.2%	90.9%	93.0%
E5	$\mathcal{D}_{\text{ANYWEL}}$	97.4%	98.8%	95.9%	97.4%
	$\mathcal{D}_{\text{RICHPIG}}$	93.7%	95.1%	92.1%	93.6%
	$\mathcal{D}_{\text{SOWEL}}$	89.4%	92.8%	85.6%	89.0%
	$\mathcal{D}_{\text{SWIM}}$	92.7%	94.9%	90.3%	92.6%
BGE	$\mathcal{D}_{\text{ANYWEL}}$	98.2%	98.6%	97.7%	98.2%
	$\mathcal{D}_{\text{RICHPIG}}$	90.2%	90.4%	90.0%	90.2%
	$\mathcal{D}_{\text{SOWEL}}$	92.8%	91.4%	94.4%	92.9%
	$\mathcal{D}_{\text{SWIM}}$	91.5%	94.2%	88.5%	91.3%

5.3.4 Robustness to negative statements

Table 5.13 summarizes the number of negative samples in the training set generated under different adversarial sampling thresholds. When the threshold is set to 0.3, 0.4, or 0.5, the number of negative samples is smaller than that of positive samples. Once the similarity threshold exceeds 0.5, the number of negative samples surpasses the positives.

Figure 5.8 illustrates the performance of the E5 model trained on datasets constructed with varying similarity thresholds, evaluated on the same test set. The results indicate that the F1-score increased steadily from 66.7% to 92.4% as the similarity threshold increased from 0.3 to 0.6, reaching its peak at 0.6. Notably, at a similarity threshold of 0.3, only 11 negative samples were included, resulting in the lowest F1-score of 66.7%. However, beyond the 0.6 threshold, model performance began to decline, with the F1-score dropping to 81.8% at a similarity threshold of 0.9.

Table 5.13: Number of samples in the training set under different adversarial sampling thresholds.

Set	Threshold						
	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Positive	2638	2638	2638	2638	2638	2638	2638
Negative	11	178	1460	5053	11767	20383	22672

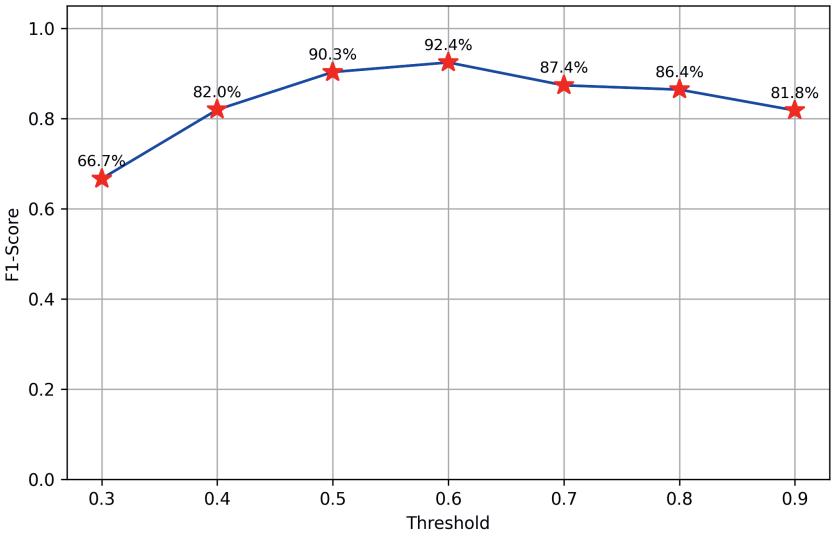


Figure 5.8: Performance evaluation of E5 trained using different thresholds in adversarial negative sampling.

5.3.5 Expert Validation

We evaluated the end-to-end classification performance on the test set of $\mathcal{D}_{\text{ABS}}$, with the results summarized in Table 5.14.

The BGE model consistently outperformed the other models across evaluation metrics. It achieved the highest accuracy (93.7%), precision (59.1%), recall (76.5%), and F1-score (66.7%), demonstrating the most balanced overall performance. RoBERTa and E5 models exhibited moderate performance, with F1-scores of 55.7% and 54.9%, respectively. Both models surpassed BERT in all metrics except for recall but remained substantially inferior to BGE, particularly in terms of precision and overall harmonic mean. Although BERT's recall (70.6%) was relatively high, its low precision (41.4%) compromised the F1-score, resulting in the lowest value (52.2%) among

Table 5.14: Comparison of end-to-end models on the $\mathcal{D}_{\text{ABS}}$ test set.

Model	Accuracy	Precision	Recall	F1-Score
BERT	89.3%	41.4%	70.6%	52.2%
RoBERTa	91.5%	48.9%	64.7%	55.7%
E5	90.1%	43.9%	73.5%	54.9%
BGE	**93.7%**	**59.1%**	**76.5%**	**66.7%**

the models.

5.4 Discussion

5.4.1 Comparison between statement classification approaches

We employed three strategies for sentence classification, and the end-to-end fine-tuning approach demonstrated the best performance across all evaluation metrics.

5.4.1.1 Performance analysis of few-shot learning approach

The prompt-based LLM approach achieved relatively high precision (all above 80%), but generally low recall (except for Phi-4, all below 70%). This indicates that while LLMs perform well on clear-cut cases, they tend to overlook ambiguous or low-signal instances, implicitly relying on high-confidence thresholds for prediction. This bias may stem from two main sources. First, there are limitations in the prompt design paradigms. Structured prompts often emphasize prototypical representations of concepts, biasing models toward surface-level pattern matching over deeper semantic understanding (Cheng et al., 2024). This issue is compounded by the small pool of few-shot examples, which cannot capture the full range of features of linguistic variation, thereby limiting generalization to weak signals and borderline cases (Ma et al., 2023; Amatriain, 2024). Furthermore, token-length constraints can restrict the model's ability to process the full context of non-standard expressions, resulting in a narrow and less robust decision boundary (Liu et al., 2025b). Second, the models have inherent constraints. As LLMs are primarily pre-trained on general-domain corpora, their grasp of specialized knowledge is limited, making it challenging to define nuanced downstream tasks solely through prompts (Sahoo et al., 2024). Moreover, our use of locally deployable models (12B-16B parameters) is a contributing factor. Although these models are powerful, their capacity for domain-specific generalization may be constrained compared to larger-scale counterparts (Hsieh et al., 2023).

5.4.1.2 Performance analysis of two-stage classification approach

The two-stage approach outperformed the LLM prompt method, with most model combinations achieving precision above 90% and recall above 70%. This may be because the two-stage approach decouples feature extraction and classification, allowing the model to make full use of the training data through supervised learning. By separating these two components, the machine learning classifier can be effectively trained to identify the optimal decision boundary in the feature space. This supervised process enables the model to better distinguish challenging or ambiguous cases based on their semantic features.

Among the evaluated embedding models, RoBERTa outperformed its counterparts, including BERT and the more recent BGE and E5. This may be attributed to the architectural and training advantages of RoBERta. RoBERTa employs dynamic masking, a much larger training corpus (160GB vs. BERT's 16GB), longer training duration, and a byte-level tokenizer. This enables it to capture richer linguistic features and generate embeddings with enhanced semantic and contextual information (Liu et al., 2019). In contrast, BGE and E5 are built on contrastive learning, which optimizes embeddings by pulling semantically similar samples closer and pushing dissimilar ones apart in the vector space (Xiao et al., 2024; Wang et al., 2024d). This approach focuses only on relative distances between samples, rather than their absolute positions, making it less effective for directly modeling class boundaries in classification tasks (Le-Khac et al., 2020).

Regarding machine learning classifiers, SVM and LR achieved the best performance, whereas RF and NB performed relatively poorly. The success of SVM can be attributed to the maximum-margin principle and kernel-based methods, which are particularly well-suited to the high-dimensional, semantically rich, and often linearly separable feature spaces produced by deep embeddings (Hofmann et al., 2008). When the feature exhibits mild nonlinearity, kernel functions (e.g., RBF) can implicitly map features to higher-dimensional spaces, and the well-structured embeddings further enhance the effectiveness of margin maximization and support vector selection (Liao & Couillet, 2019). LR, while simpler, benefits from the linear separability of embeddings. This allows the model to effectively map input features to class labels. Regularization techniques such as L1 or L2 play a crucial role in suppressing redundant dimensions and preventing overfitting (Salehi et al., 2019; Demir-Kavuk et al., 2011). In contrast, the high dimensionality of embeddings makes it difficult for RF to find effective splitting criteria in sparse dense feature spaces, leading to overfitting or underfitting (Wang et al., 2018). Similarly, NB's assumption of feature independence is fundamentally incompatible with the correlated dimensions of dense embeddings, resulting in inaccurate probability estimations and degraded classification performance (Blanquero

et al., 2021).

5.4.1.3 Performance analysis of end-to-end classification approach

Of three approaches, the end-to-end fine-tuned models achieved the highest and most stable performance, with all models attaining precision above 92% and recall exceeding 90%. This advantage likely stems from the fact that all model parameters are jointly optimized on the training data, allowing the embedding space to be directly adapted to the classification objective (Gururangan et al., 2020). In contrast, the two-stage approach relies on fixed pre-trained embeddings, which lack task-specific knowledge for optimal classification performance. The effectiveness of feature extraction relies on the similarity between the pre-training and the target task. If the domain gap is large, training classifiers alone often performs poorly and is not adequately adapted to downstream domain tasks (Peters et al., 2019).

An inversion in model performance was observed between the two-stage and end-to-end approaches. While RoBERTa performed best in the two-stage approach, the E5 achieved the highest accuracy (94.3%) and F1-score (94.2%) in the end-to-end setup, surpassing both RoBERTa and BERT. We hypothesize that this improvement stems from E5's contrastive learning objective during pre-training, which encourages semantically similar examples to be embedded closer together and dissimilar ones to be farther apart (Wang et al., 2024d). This process inherently imposes a degree of separability in the representation space. When fine-tuned in an end-to-end manner, E5 is able to effectively reshape and align its already well-structured embedding space with the new classification boundaries. This initialization advantage allows the model to converge more efficiently and accurately during training (Rethmeier & Augenstein, 2023). Additionally, the use of linear layers as the classification head places a stronger reliance on the quality of the underlying embeddings. In this context, the contrastive representations produced by E5 provide a beneficial starting point, requiring fewer transformations to achieve effective class separation compared to models trained with language modeling objectives.

In summary, among the three sentence classification approaches, the end-to-end fine-tuning method achieved the best overall performance across all metrics. The E5 model demonstrated the highest precision and overall F1-score, while the BERT model achieved the best recall.

5.4.1.4 Analysis of models' performance on different subsets

A salient observation is that all four end-to-end models performed better on the more recent $\mathcal{D}$ANYWEL subset than on the other test sets. We hypothesise two complementary reasons. First, $\mathcal{D}$ANYWEL is closer to the training distribution: it shares

species (e.g., laying hens and dairy cows) and exhibits lexical and semantic overlap with the training domain, enabling the models to reuse domain-specific patterns—particularly terminology about housing, feeding, and common welfare concerns in intensive systems. During fine-tuning, such patterns can crystallise into domain-specific "shortcuts" aligned with the target labels (Michel, 2021), improving accuracy when the test data reflect a similar distribution. Second, $\mathcal{D}_{\text{ANYWEL}}$ contains statements written under more formalised and specific guidelines than the older models and focuses on welfare potential (i.e., assuming good management practices). This standardisation and narrower scope likely reduce ambiguity, making relevant statements easier to identify.

The challenge of out-of-distribution generalisation becomes apparent on subsets such as $\mathcal{D}$SWIM, which contains welfare descriptions of farmed Atlantic salmon in marine environments. Despite this distribution shift, most models maintained high precision (>94%) and recall (>90%) on $\mathcal{D}$SWIM, with the notable exception of BGE's recall. This suggests that, beyond surface-level pattern matching, the models leverage more abstract, domain-invariant cues indicative of animal welfare. A plausible contributing factor is the relative immaturity of fish-welfare research: the SWIM semantic model was derived from more conservative and explicitly specified guidance, focusing on broadly accepted criteria. As a result, statements in $\mathcal{D}_{\text{SWIM}}$ may be more explicit and self-contained, reducing linguistic variability and making positive welfare-relevant statements easier to identify even under a distribution shift.

5.4.2 Robustness to negative statements

In our experimental investigation, we observed a non-monotonic relationship between the similarity threshold employed in adversarial negative sampling and the F1 score on the test set. Specifically, as the threshold increased, the F1 score initially improved before subsequently declining. This trend highlights an inherent trade-off between the quantity and quality of negative samples used during training. At lower similarity thresholds (e.g., ≤ 0.5), the selected negative samples tend to be semantically distant from the corresponding positive instances, thereby ensuring high sample purity and minimal label noise. However, this advantage is offset by a drastic reduction in the number of available negative samples. For instance, at a threshold of 0.3, the negative set comprises merely 11 samples, which introduces significant data sparsity. Under such conditions, the model encounters difficulty in learning a robust and generalizable decision boundary due to insufficient training signals. As the threshold increases to an intermediate range (e.g., around 0.6), a more favorable balance emerges. The number of negative samples becomes sufficiently large while maintaining relatively high semantic dissimilarity to the positive class. This combination of adequate sam-

ple size and preserved label quality facilitates richer supervision, thereby enhancing the model's capacity to construct a more discriminative and well-calibrated boundary. The observed peak in F1 performance within this range further supports this interpretation. Beyond this optimal interval, however, continued increases in the similarity threshold result in a marked deterioration in negative sample quality. As semantically similar (and potentially mislabeled) "hard negatives," which may in fact be latent positives, are incorporated into the training set, label noise escalates significantly. This phenomenon reduces the separability between positive and negative instances in the embedding space, ultimately impairing the model's discriminative performance.

The model has notable resilience to imperfections in the negative sampling process. Except at the extremely low threshold of 0.3, where performance degradation is pronounced due to insufficient training data, the model maintains an F1 score exceeding 80% across a broad spectrum of thresholds. This robustness indicates that the fine-tuned E5 model has acquired a stable and generalizable decision function, rather than overfitting to specific data artifacts.

5.4.3 Analysis of expert validation

Compared to the $\mathcal{D}_{\text{ALL}}$, model performance on the expert validation set of $\mathcal{D}_{\text{ABS}}$ exhibited a notable decline, with precision often falling below 60% and recall dropping below 80%. This discrepancy reflects a fundamental difference in task complexity and data distribution between the two datasets. The $\mathcal{D}_{\text{ABS}}$ includes a high proportion of semantically ambiguous and borderline samples, where correct classification requires not only surface-level pattern recognition but also in-depth domain knowledge and complex inferential reasoning. Indeed, these instances are challenging even for experienced human annotators.

The performance ranking of models on $\mathcal{D}_{\text{ABS}}$ diverged from that observed on the $\mathcal{D}_{\text{ALL}}$. Specifically, the BGE model outperformed others, including the previously stronger E5. This reversal may be attributed to differences in pre-training objectives. BGE employs a contrastive learning strategy that explicitly optimizes the separation of hard negative pairs, encouraging the embedding space to form sharper semantic boundaries through gradient-based margin maximization (Chen et al., 2024). This inductive bias enhances its discriminative capacity under low-resource or semantically ambiguous conditions. In contrast, E5 relies on a more generic contrastive loss, which is prone to suboptimal convergence in sparse or noisy distribution settings (Wang et al., 2022b).

Despite the observed decline in precision, the proposed framework remains practically

valuable for real-world information for application in SemMod-AWA to assess animal welfare. In knowledge discovery contexts, recall is often prioritized over precision to ensure exhaustive retrieval of candidate instances. Serving as a high-recall candidate generation engine, our models efficiently surface potentially relevant information at scale. The false positive filtering can then be managed by human experts, offering significant advantages over fully manual review. In addition, the models developed in this study have internalized a great deal of linguistic and conceptual knowledge in animal-related fields, making them a pre-trained backbone for future applications. When encountering a new assessment domain or changing selection criteria, these models can be quickly adapted by making a small amount of fine-tuning on a limited amount of annotated data.

5.4.4 Limitations and future directions

While our proposed method demonstrates significant efficacy in identifying relevant scientific statements to assess animal welfare through semantic modelling, its primary limitation lies in the inherently dynamic nature of the classification criteria. Different experts possess unique qualifications and associated norms, modeling goals may vary and evolve, and an individual modeler's understanding can shift over time—a phenomenon known as "concept drift." This makes it challenging to contrast a single, universally applicable model. This challenge motivated us to validate our model on a new, demanding dataset curated under different, state-of-the-art standards. Nevertheless, when applied to new, complex scenarios with fine-grained distinctions, a pre-trained model may exhibit reduced precision and recall. Future work therefore focus on evolving our tool into a dynamic, adaptive system. A key insight for this task is that for SemMod-AWA modelers, recall is often more critical than precision, as false positives can more easily be discarded in subsequent steps, while missed statements (false negatives) may result in an information loss. Therefore, we propose transitioning towards a Human-in-the-Loop (HITL) (Wu et al., 2022) or active learning framework (Rouzegar & Makrehchi, 2024). In this paradigm, the model's predictions would be reviewed by the expert, and any corrections—especially the misclassified hard examples—would be fed back to incrementally fine-tune the model. By integrating this continuous feedback loop, the system would not only improve its general performance but also dynamically adapt to the specific, evolving standards of the user. This transforms the tool from a static classifier into a collaborative, personalized knowledge extraction partner that becomes more accurate and aligned with its user's needs through ongoing use.

Secondly, our current model primarily operates at the sentence level, whereas animal welfare information in unstructured text documents can often be expressed across

entire paragraphs or sections. This requires an ability to synthesize multi-sentence information to determine if it collectively meets the criteria for an animal welfare statement for SemMod-AWA, a capability the current system lacks. Our proposed solution for future work is to develop a multi-step statement extraction pipeline. This will leverage prompt engineering with LLMs to first summarize lengthy paragraphs, thereby condensing information and increasing the semantic density of shorter text units. Our existing statement extraction model will then be applied to these summaries, aiming to significantly improve the retrieval of welfare-related statements embedded within longer narratives, with careful validation to ensure the summarization preserves essential information. Moreover, in scientific texts, essential contextual information may appear in earlier sections—for instance, abbreviations introduced at the outset or methodological details in the Materials and Methods—that are necessary to determine whether a finding is welfare-relevant. Incorporating mechanisms to capture and integrate such dispersed context will therefore be an important extension of future work.

A third avenue concerns *species specification*. For each positive statement, the system should also identify the applicable species (e.g., pigs, chickens) and animal categories (e.g., pregnant sows, weaned piglets). We envisage a species-aware tagging module that combines lightweight NER with normalisation to controlled vocabularies, or a multi-label classifier jointly trained to predict statement relevance and species/category tags. This would reduce ambiguity, enable species-specific retrieval and analytics, and support cross-study synthesis.

5.5 Conclusion

This study presents an NLP-based framework for the automated classification and extraction of scientific statements that can be used to assess animal welfare from unstructured textual sources. To address the challenge of limited negative samples, a contrastive sampling strategy was introduced, which enabled the generation of high-quality negative instances for model training. This contributed to the development of a robust end-to-end classifier with strong generalization capacity.

Among the evaluated approaches, the end-to-end fine-tuning method achieved the highest precision (92.1%–96.0%) and recall (90.1%–95.5%), outperforming few-shot learning with large language models and two-stage classification pipelines. The E5 model achieved the best precision (96.0%), while BERT delivered the highest recall (95.5%). Moreover, the model demonstrated consistent performance across datasets with varying noise levels, confirming its resilience to ambiguous or weakly labeled samples.

On a challenging expert-annotated dataset where semantic differences between positive and negative cases were minimal, the BGE-based end-to-end model achieved superior results (precision: 59.1%; recall: 76.5%), indicating its suitability for nuanced classification tasks.

Overall, this research contributes a scalable and adaptable solution to reduce reliance on expert labeling in animal welfare data processing. The framework and accompanying user interface offer practical tools for supporting welfare assessment from large-scale textual resources. Future research may explore the deployment of this system in multilingual or cross-domain settings to further enhance its applicability in real-world agricultural contexts.

Chapter 6

S-MADSS: An agentic framework for evidence-Grounded, compliance-oriented decision support in Atlantic salmon farming

This chapter has been submitted as:

Zhang, C., Gansel, L. C., Bracke, M., & da Silva Torres, R. (2026). S-MADSS: An agentic framework for evidence-grounded, compliance-oriented decision support in Atlantic salmon farming. Under review at *Smart Agricultural Technology*.

Abstract

Atlantic salmon farming requires operational decisions that must satisfy regulatory requirements, protect fish welfare, and remain feasible under farm conditions. Existing digital tools mainly support isolated functions, such as lice forecasting or feeding optimization, and rarely provide traceable decision support across a full operational workflow. To address this gap, we propose S-MADSS, an evidence-grounded decision-support system for Atlantic salmon farming. The system supports two main functions: (i) evidence-grounded question answering (QA) for informational queries and (ii) compliance- and risk-oriented decision support for Atlantic salmon sea-cage operations. The system combines a traceable retrieval-augmented generation (RAG) pipeline with a multi-agent auditing workflow. The evidence pipeline integrates BM25 retrieval, dense vector retrieval, cross-encoder reranking, and local context expansion to construct supporting evidence from a domain knowledge base of regulations, manuals, and scientific literature. On top of this evidence layer, specialized agents perform answer generation, review, action proposal, and multi-perspective risk auditing from regulatory, welfare, and feasibility perspectives. We evaluate S-MADSS through four complementary settings: retrieval benchmarking, evidence-grounded QA, multi-hop evidence integration, and scenario-based risk identification, followed by a real-farm workflow demonstration. Hybrid retrieval with reranking achieved the best retrieval performance, with a mean Hit@k of 0.910 and Hit@1 improving from 0.478 to 0.752 over the BM25-only baseline. Local context expansion improved evidence coverage and answer correctness in multi-hop QA. In risk identification, the evidence-grounded auditing workflow achieved a maximum risk recall of 0.950, substantially outperforming retrieval-free LLM-based identification (0.637). A real-farm case further showed that monitoring data can be translated into compliance-oriented recommendations with explicit assumptions, risks, and supporting evidence. These results indicate that S-MADSS extends retrieval-augmented generation from information access to structured, traceable, and compliance-oriented decision support in salmon farming, and highlights the potential of evidence-grounded agentic workflows for regulation-sensitive agricultural domains beyond salmon farming. The code and agent prompts

used in this study are available at `https://github.com/jay-zc/S-MADSS.git` (As of March 2026).

6.1 Introduction

Aquaculture has become an increasingly important component of global food production, with the salmon farming industry as one of its most valuable sectors (FAO, 2024). Over time, the gradual refinement of legal frameworks, regulatory policies, and operational standards contributed to the sustainable development of the salmon farming industry. The daily management of the sea cages is governed by a variety of standards, licenses, policies, welfare codes, and management manuals (Luthman et al., 2019). These cover various aspects, including parasite and disease monitoring, treatment procedures, handling, and environmental surveillance.

In practice, although technology providers and farm operators have developed a range of digital systems for documentation, analysis, planning, and decision support, many operational decisions still rely heavily on farmer experience and manual consultation of extensive documentation. This creates a practical challenge: relevant constraints are distributed across long and heterogeneous sources, while regulations, technical recommendations, and reporting requirements continue to evolve. As a result, it is increasingly difficult to ensure that day-to-day decisions remain fully aligned with current compliance obligations and wider farm objectives, such as sustainability goals. Operational mistakes can have substantial economic, environmental, and welfare consequences. For example, in 2024, Masoval carried out a salmon-lice treatment despite the poor health condition of the fish, leading to around 10,000 fish deaths.[1] Incidents of this kind illustrate the need for decision support that can surface operational thresholds and their supporting evidence at the point of action.

Therefore, decision support systems (DSSs) may play an important role in improving regulatory compliance and operational safety in aquaculture. Various studies have explored digital tools for specific aquaculture tasks (Budaev et al., 2025; Ángel Cobo et al., 2019). For instance, Lakseluskartet[2] can be used to simulate the spatial distribution of salmon lice on the Norwegian coast, providing early alerts for the potential outbreak of salmon-lice infestation. Beyond risk assessment, other optimization-based approaches have also been investigated. For instance, (Budaev et al., 2025) proposed a digital twin framework to simulate stomach and gut digesta transit and feed in-

[1] `https://ilaks.no/masoval-ilagt-bot-pa-naer-to-millioner-kroner-spesiell-og-risikofylt-situasjon/` (As of March 2026).

[2] `https://www.hi.no/forskning/marine-data-forskningsdata/lakseluskart/html/lakseluskart.html`

take in rainbow trout, supporting feeding strategy optimization. Similarly, (Ángel Cobo et al., 2019) developed a DSS that integrates a bio-economic model encompassing the economic, environmental, biological, and technical factors to maximize the economic performance of seabream reared in floating cages. Although such tools are valuable, they primarily support isolated functions such as lice forecasting, feeding optimization, or bio-economic planning. They rarely provide compliance-oriented guidance that jointly considers regulatory requirements, fish welfare, environmental constraints, and operational feasibility within a single decision workflow.

Recent developments in large language models (LLMs) enable more comprehensive decision-support systems by improving QA, knowledge synthesis, and contextual reasoning. However, directly deploying LLMs in compliance-oriented settings remains risky. Models may hallucinate plausible but incorrect statements and may violate the latest policies when their training data are outdated (Ji et al., 2023; Bang et al., 2025). Retrieval-augmented generation (RAG) mitigates these issues by grounding generation in an external knowledge base and constraining outputs to verified sources (Wu et al., 2025; Arslan et al., 2024). By continuously updating the knowledge base with legal, welfare, and technical documents, a RAG-based system can improve both the reliability and the regulatory currency of decision support. However, evidence retrieval alone is not sufficient for operational decision support. In practice, farm decisions often require the joint consideration of multiple perspectives, including legal compliance, fish welfare, technical feasibility, and site-specific farm conditions. This makes single-pass QA insufficient for scenarios in which the system must identify risks, compare possible actions, and state assumptions explicitly. Agentic LLM systems provide a promising way to structure this broader process. Instead of treating the model as a single generator, agentic systems decompose a task into coordinated reasoning steps, invoke external tools when needed, and allow specialized roles to review intermediate outputs (Luo et al., 2025a). For compliance-oriented decision support, this design is especially attractive because it separates evidence construction from downstream judgment, and makes it possible to assess a proposed action from multiple perspectives before presenting a recommendation.

Based on this motivation, we propose a salmon multi-agent decision support system (S-MADSS), an evidence-grounded and compliance-oriented decision-support system for Atlantic salmon farming. S-MADSS is designed to support two task types: (i) evidence-grounded QA for information-seeking queries and (ii) compliance- and risk-oriented decision support for operational scenarios. By combining a domain knowledge base with optional farm-specific contextualization within a traceable, role-based multi-agent workflow, S-MADSS moves beyond single-function tools toward decision support that jointly considers regulatory, welfare, and feasibility constraints. This

study presents a proof-of-concept for evidence-grounded, compliance-oriented decision support in aquaculture and demonstrates the initial feasibility of a traceable workflow for salmon-farming applications.

The main contributions of this work are as follows:

1. We propose S-MADSS, a compliance-oriented decision-support system framework implemented for Atlantic salmon farming that unifies evidence-grounded question answering and compliance- and risk-aware operational decision support within a single workflow.

2. We design a traceable evidence-construction pipeline for compliance-oriented aquaculture tasks by combining hybrid retrieval, reciprocal rank fusion, cross-encoder reranking, and local context expansion, enabling explicit linking between system outputs and supporting evidence chunks.

3. We formulate compliance-oriented farm decision support as a multi-perspective auditing process in which specialized agents assess candidate actions from regulatory, welfare, and feasibility perspectives before producing a final recommendation.

4. We conduct a component-to-workflow evaluation, including single-hop grounded QA, multi-hop evidence integration, scenario-based risk identification, and a real-farm workflow demonstration, to assess retrieval quality, grounded answer quality, evidence coverage, and risk-detection capability.

6.2 Materials and methods

6.2.1 Concept of general MADSS framework

The multi-agent decision support system (MADSS) is a layered agentic framework for compliance-oriented decision support in agriculture and aquaculture, designed for settings where operational decisions depend on both stable domain knowledge and case-specific contextual information. As shown in Fig. 6.1, MADSS is organized into four layers. From top to bottom, these layers are: i) The workflow layer. It defines task-level coordination logic, including task scope and decision boundaries; ii) The agent layer, which contains role-based reasoning components that determine how bounded autonomy is exercised within this scope; iii) The tool layer, which provides executable modules for the AI agents, such as environment monitoring, information search, risk identification, etc; iv) The resource layer. It organizes both stable resources (e.g., regulations, manuals, and scientific literature) and contextual inputs

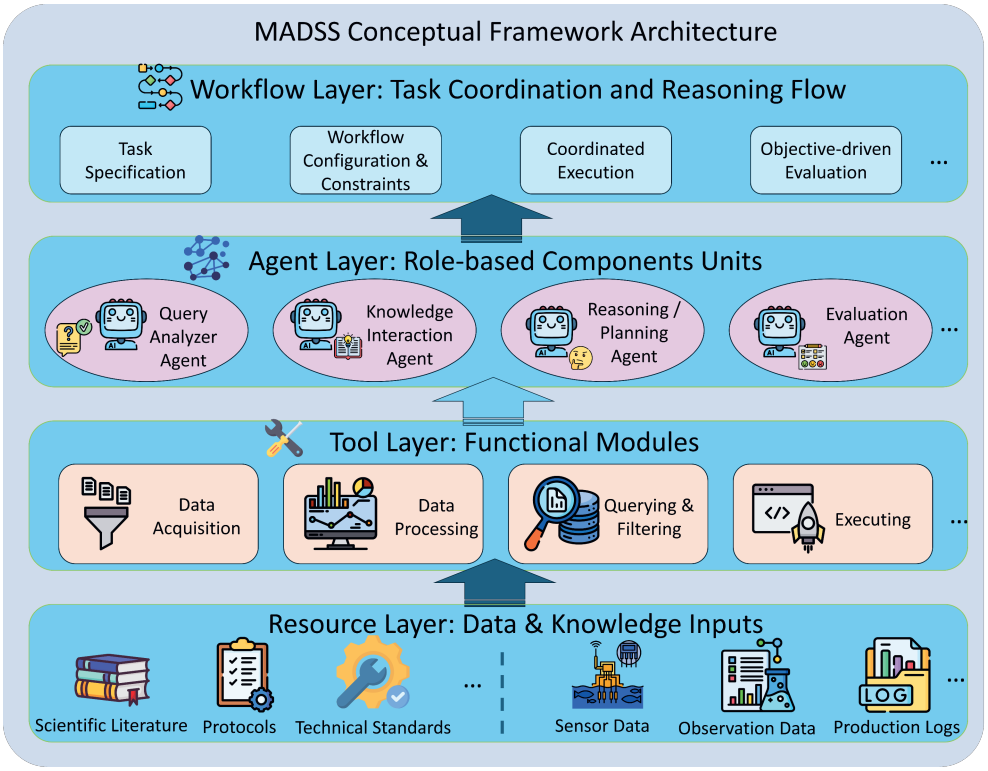


Figure 6.1: Conceptual framework architecture of MADSS. The framework is organized into four layers: workflow, agent, tool, and resource. The workflow layer defines task coordination, the agent layer supports role-based reasoning, the tool layer provides functional modules, and the resource layer contains both stable knowledge sources and contextual inputs. Arrows indicate the main directions of control and information flow across layers.

(e.g., farm records and observations), thereby defining the information available for grounding and decision support.

6.2.2 System instantiation: salmon farming (S-MADSS)

In this study, MADSS is instantiated as S-MADSS for Atlantic salmon sea-cage farming. The system supports two task types: evidence-grounded QA for informational queries and compliance- and risk-oriented decision support for operational scenarios. The detailed design of this instantiation is described in the following layer-specific sections (workflow, agent, tool, and resource layers).

6.2.3 Workflow layer

As illustrated in Fig. 6.2, S-MADSS executes a routed workflow for two task types: evidence-grounded QA and compliance-oriented decision support. A query routing

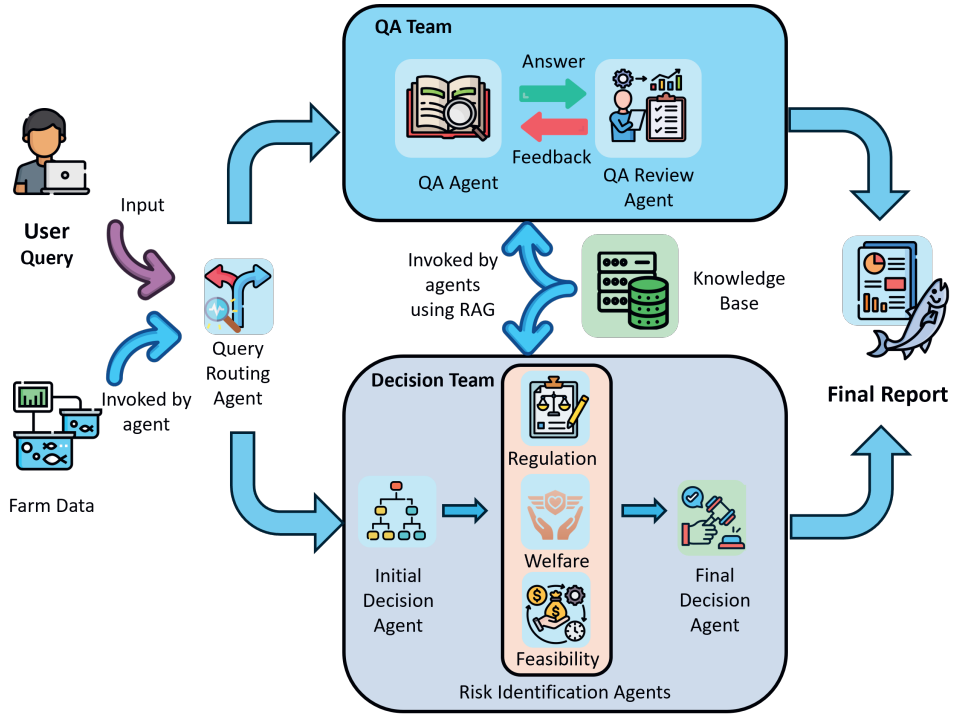


Figure 6.2: Overall workflow of S-MADSS. User queries and optional farm-specific data are processed by a query routing agent, which routes the request to either the QA team for evidence-grounded QA or the decision team for compliance- and risk-oriented decision support. The QA team performs answer generation and review, while the decision team generates candidate actions, conducts regulation-, welfare-, and feasibility-oriented risk assessment, and produces a final recommendation for action. Both pathways are supported by the knowledge base and lead to a traceable final report.

agent first determines whether the input should be handled by the QA pathway or the decision-support pathway, and whether farm-specific contextualization is required. When site-specific information is needed, a farm-environment report tool constructs a structured farm-context representation F for the target site.

All workflow components interact through a shared environment state:

$$S = \{q, Q, E, F, A, M\}, \tag{6.1}$$

where q is the user query; $Q = \{q_1, q_2, \ldots, q_n\}$ is the set of sub-queries; $E = \{(e_i, \mathrm{id}_i, \mathrm{meta}_i)\}_{i=1}^{n_E}$ is the set of retrieved evidence chunks from the knowledge base K, where each chunk corresponds to a sentence-preserving text segment derived from source documents, together with their identifiers and metadata; F is the optional farm-context representation; A is the set of candidate actions generated in the decision-support pathway; and M stores inter-agent messages and feedback his-

Table 6.1: Example of query decomposition into sub-queries for evidence retrieval.

Query	Generated sub-queries
What treatment options are available for salmon lice in Atlantic salmon farming?	q_1: What are the major treatment categories for salmon lice? q_2: What medicinal treatment options are available? q_3: What non-medicinal treatment options are available? q_4: What conditions or indications are associated with each treatment type? q_5: What key regulatory, welfare, or operational constraints apply to these treatments?

tories. Table 6.1 provides a concrete example of query decomposition for evidence retrieval.

State update and traceability At each workflow step, routing decisions, retrieved evidence, intermediate outputs, and review feedback are written to S. This state records how the final answer or recommendation is derived from the query, contextual inputs, and evidence. If the available evidence is judged to be missing or insufficient for the current step, the workflow triggers an additional retrieval step, updates E, and then resumes generation or evaluation.

QA pathway For informational queries, the workflow first generates sub-queries Q conditioned on (q, F) and retrieves evidence $E \subset K$. A preliminary answer is then generated as:

$$y_0 = f_{\text{QA_init}}(q, E, F), \tag{6.2}$$

where $f_{\text{QA_init}}$ denotes the initial evidence-grounded answer function, and y_0 denotes the base answer.

The answer is then reviewed and iteratively refined:

$$y_{t+1} = f_{\text{QA_refine}}(y_t, q, E, F, r_t), \qquad t = 0, 1, \ldots, T_{\max} - 1, \tag{6.3}$$

where $r_t \in M$ denotes structured review feedback. The refinement loop stops when the answer is judged adequate or when the iteration limit $T_{\max}$ is reached.

Decision-support pathway For decision-support queries, the workflow first retrieves task-relevant evidence $E \subset K$ and then constructs a candidate action

set:

$$A = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}, \tag{6.4}$$

where each candidate action $\alpha \in A$ is represented as:

$$\alpha = (act, P, C, E_{\text{exp}}, R_{\text{risk}}, E_{\text{id}}), \tag{6.5}$$

with *act* denoting the action type, P the required parameters, C the preconditions, E_{exp} the expected effects, R_{risk} the anticipated risks, and E_{id} the supporting evidence chunk identifiers.

Each candidate is then audited from regulatory (*reg*), welfare (*wlf*), and feasibility (*fea*) perspectives. The corresponding audit outputs are stored as:

$$M_{\text{audit}} = \{r^{(\text{reg})}, r^{(\text{wlf})}, r^{(\text{fea})}\}, \tag{6.6}$$

where $r^{(\text{reg})}$, $r^{(\text{wlf})}$, and $r^{(\text{fea})}$ denote the audit results from the respective perspectives.

Finally, the workflow aggregates the candidate actions and audit outputs to produce a recommendation:

$$y^* = f_{\text{final}}(A, M_{\text{audit}}, E, F), \tag{6.7}$$

where y^* denotes the final recommendation, the final output includes the recommended action, explicit assumptions and preconditions, identified risks and mitigation measures, and traceable evidence links to the supporting chunks in E.

6.2.4 Agent layer

S-MADSS is implemented as an 8-role-based system of agents that interact through the shared state S. Each agent has a bounded responsibility within the workflow and communicates through structured outputs written to S.

6.2.4.1 Query routing agent

The query routing agent serves as the system entry point. Given a user query q, it determines (i) whether the task should be handled as evidence-grounded QA or as decision support and (ii) whether farm-specific context F is required. Its output is a routing decision and, when needed, a context request for downstream processing.

6.2.4.2 QA team

The QA team consists of a QA agent and a QA review agent.

QA agent The QA agent is also responsible for constructing evidence-grounded responses. Given (q, F), it formulates sub-queries, invokes the retrieval pipeline, and drafts an answer grounded in the retrieved evidence. Its output includes both the response text and the referenced evidence chunk identifiers.

QA review agent The QA review agent evaluates the drafted answer with respect to three criteria: (i) coverage of the constraints implied by the query, (ii) consistency between claims and retrieved evidence, and (iii) usability of the response. It returns structured feedback $r_t \in M$ for bounded revision. When support is judged insufficient or weak, it can trigger additional retrieval before the next revision step.

6.2.4.3 Decision team

The decision team consists of an initial decision agent, three specialist risk-identification agents, and a final decision agent.

Initial decision agent Conditioned on (q, F, E), the initial decision agent proposes a small set of candidate actions. For each option, it specifies the action type, operational parameters, key preconditions, expected effects, anticipated risks, and the identifiers of the supporting evidence chunks.

Risk-identification agents Three specialist agents, each operating on its own domain (regulation, welfare, and feasibility), assess each candidate from complementary perspectives, generating the structured audit outputs $M_{\text{audit}} = \{r^{(\text{reg})}, r^{(\text{wlf})}, r^{(\text{fea})}\} \subset M$. For each proposed option, they identify potential risk points R_{risk}, including regulatory conflicts, welfare concerns, and feasibility-related constraints. If the currently available evidence does not sufficiently support a relevant threshold, condition, or exception, the agent can request an additional retrieval step from the knowledge base K.

Final decision agent The final decision agent aggregates the candidate set A, and the specialist audits M_{audit} to select or refine the final recommendation y^*. The output y^* prioritizes traceability and implementability by explicitly stating key assumptions and preconditions and attaching evidence chunk identifiers to each critical constraint or threshold referenced in the recommendation.

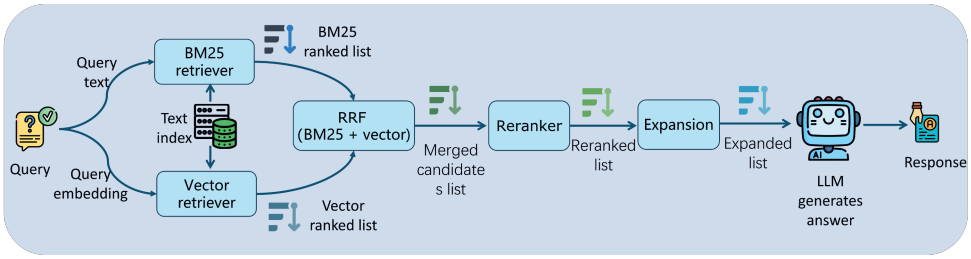


Figure 6.3: Hybrid retrieval pipeline. BM25 and vector retrieval results are fused, reranked with a cross-encoder, expanded with neighboring chunks, and passed to the generator together with evidence chunk identifiers.

6.2.5 Tool layer

6.2.5.1 RAG

As shown in Fig. 6.3, RAG is the core tool-layer component for evidence construction in S-MADSS. Operating over the domain knowledge base (Sec. 6.2.6), it provides traceable, context-aware evidence for both evidence-grounded QA and compliance- and risk-oriented decision support. The pipeline consists of five stages: hybrid retrieval, rank fusion, cross-encoder reranking, local context expansion, and evidence-grounded generation. The tool returns generated outputs together with the identifiers and metadata of the evidence chunks used.

To improve evidence coverage over heterogeneous long-form documents, we combine sparse lexical retrieval and vector retrieval, which provide complementary signals for exact constraint matching and semantic similarity. Sparse retrieval is implemented with BM25 (Robertson & Zaragoza, 2009), which favors exact matching of regulatory terms, thresholds, and numeric expressions. The BM25 score for a query q and chunk d is

$$\mathrm{Score}_{\mathrm{BM25}}(q,d) = \sum_{t \in q} \mathrm{IDF}(t)\, \frac{f(t,d)\,(k_1+1)}{f(t,d) + k_1 \left(1 - b + b \frac{|d|}{\mathrm{Avg}(|d|)}\right)}, \tag{6.8}$$

$$\mathrm{IDF}(t) = \log\left(\frac{N - n(t) + 0.5}{n(t) + 0.5}\right), \tag{6.9}$$

where $f(t,d)$ is the frequency of token t in chunk d, $|d|$ is the chunk length, $\mathrm{Avg}(|d|)$ is the average chunk length, k_1 and b are hyperparameters, and IDF is the inverse document frequency measuring the rarity of a term across the corpus. In parallel, vector retrieval embeds the query and document chunks into a shared semantic space and scores each chunk by cosine similarity:

$$\mathrm{Sim}(q,d) = \frac{\vec{q} \cdot \vec{v}_d}{\|\vec{q}\| \|\vec{v}_d\|}, \tag{6.10}$$

where $\vec{q}$ and $\vec{v}_d$ denote the embedding vectors of the query and chunk, respectively. This improves retrieval robustness when the query and source text use different surface forms.

Because sparse and vector scores are not directly comparable, their ranked lists are merged using Reciprocal Rank Fusion (RRF):

$$S_{\mathrm{RRF}}(q,d) = \sum_{m\in\{\mathrm{bm25,vector}\}} \frac{1}{k + r_m(d)}, \tag{6.11}$$

where $r_m(d)$ denotes the rank of chunk d under retrieval method m, and k is a smoothing constant. We then apply a cross-encoder reranker to the top-K candidates to refine relevance estimation through joint query-chunk encoding. In this study, we use `bge-reranker-v2-m3` as the cross-encoder reranker due to its strong reranking performance and robustness on long-form text (de Souza P. Moreira et al., 2024; Chen et al., 2024).

To reduce the fragmentation introduced by chunking, we apply local context expansion after reranking (Zhao et al., 2024a). For each selected chunk d, neighboring chunks from the same document are added within a symmetric window of size w:

$$E_w(d) = \{d_{-w}, \ldots, d_{-1}, d, d_{+1}, \ldots, d_{+w}\}, \tag{6.12}$$

where $E_w(d)$ denotes the locally expanded evidence set centered on d. Finally, the generator produces an answer conditioned on the selected evidence chunks and their metadata. By retaining explicit chunk identifiers throughout this process, the pipeline supports traceability by linking output claims back to their supporting evidence.

6.2.5.2 Farm-environment report tool

The farm-environment report tool provides site-specific contextual information for S-MADSS by retrieving farm-level environmental and operational records and converting them into a structured farm-context report. The retrieved records are standardized and indexed by locality identifier, after which an LLM summarizes salient historical patterns (e.g., lice dynamics and prior interventions) into a compact context representation. This report is used to condition downstream evidence retrieval and decision support when site-specific contextualization is required. An example of the report from the tool is shown in Fig. 6.4. This tool transforms plain tabular data into reports that include detailed treatment summaries and trend analyses from the past few weeks.

6.2.6 Resource layer

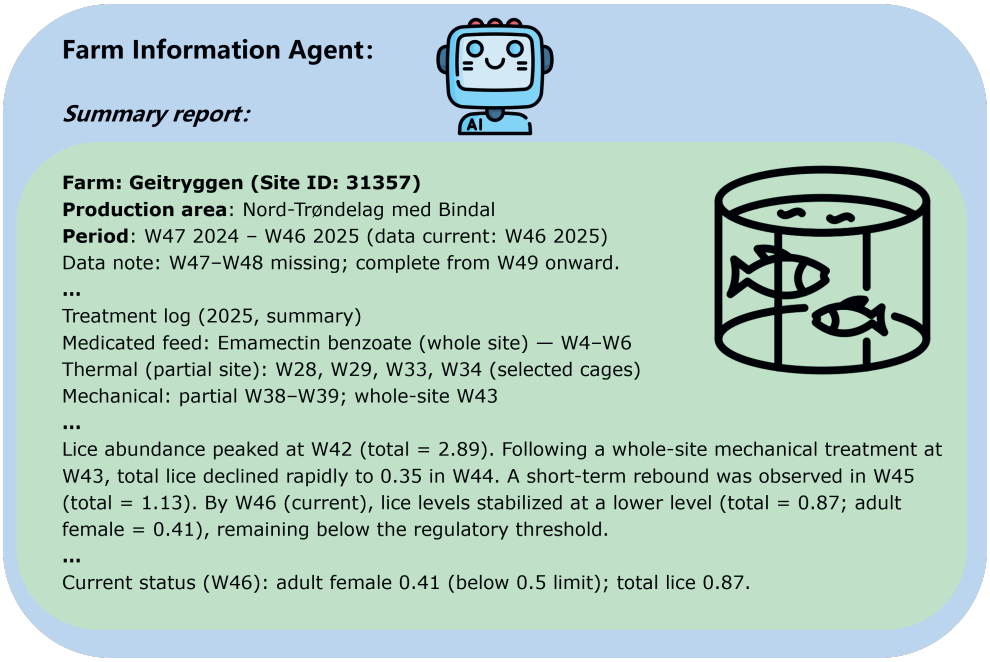


Figure 6.4: Farm-environment report tool. Farm records are retrieved from the Norwegian Fish Health platform, normalized into a structured representation, indexed by locality identifier, and summarized into a farm-context report for downstream use. W[number] denotes the calendar week number.

S-MADSS relies on two resource types: (i) a domain knowledge base used for evidence retrieval and constraint grounding, and (ii) farm-level operational data used for contextualization in case-specific queries.

Knowledge base To support evidence-driven reasoning, we constructed a domain-specific knowledge base as the central repository of regulatory and technical information (Fig. 6.5). The corpus comprises three categories: (1) regulatory documents covering legal compliance, environmental protection, and welfare requirements; (2) industry operation manuals and technical guidelines; and (3) peer-reviewed scientific publications and technical reports. Regulatory documents were primarily collected from official legal and governmental websites relevant to Norwegian aquaculture (e.g., Lovdata[3] and related authority websites). Complementary sectoral and background materials were gathered from public institutional platforms such as FAO,[4] while peer-reviewed publications were retrieved from Scopus using keywords related to salmon aquaculture management, including salmon farming and salmon lice treatment. All

[3] `https://lovdata.no/` (As of March 2026).

[4] `https://www.fao.org/home/en/` (As of March 2026).

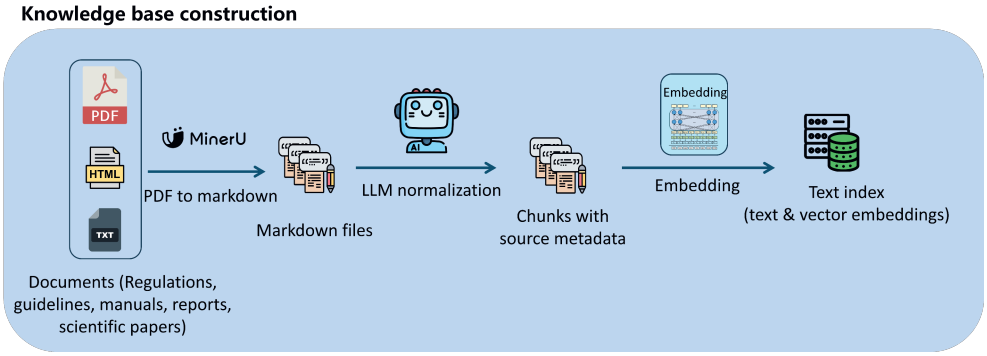


Figure 6.5: Knowledge-base construction pipeline. Documents are converted into structured Markdown, English-normalized text, segmented into sentence-preserving chunks, and embedded for vector retrieval. The final index stores chunk text together with identifiers and metadata for traceability.

collected materials were organized by document type, resulting in seven regulatory documents, 40 manuals/guidelines, and 380 research articles. The repository containing the full document list will be made publicly available upon acceptance.

For downstream processing, PDF files were converted into structured Markdown using MinerU (Niu et al., 2025). The corpus is bilingual, with regulatory documents predominantly in Norwegian and scientific publications largely in English. We applied a normalization step that includes cross-lingual translation to produce an English-normalized version for retrieval and generation using `MIMOv2-flash` (Team et al., 2026) due to its multilingual support and cost-effectiveness. The original text is preserved alongside the normalized text to maintain auditability and enable back-checking when needed.

The normalized text was segmented into sentence-preserving chunks using a recursive, token-based strategy that preserves sentence boundaries. Chunk anchors were placed at approximately 800 tokens, which has been commonly used in research (Wang et al., 2025c; Lima, 2025) and adjusted to the nearest sentence boundary to reduce fragmentation (Fig. 6.6). Each chunk was embedded using `Qwen 3-Embedding-0.6B` (Zhang et al., 2025d) for vector retrieval and semantic matching, which is a resource-efficient embedding model with strong retrieval performance (Roth et al., 2025).

To support transparent evidence attribution, each chunk is stored with metadata that includes the source document identifier, document type (regulation/manual/literature), original language, authorship, and publication date. These fields are referenced during retrieval and response generation to provide traceable, evidence-grounded outputs.

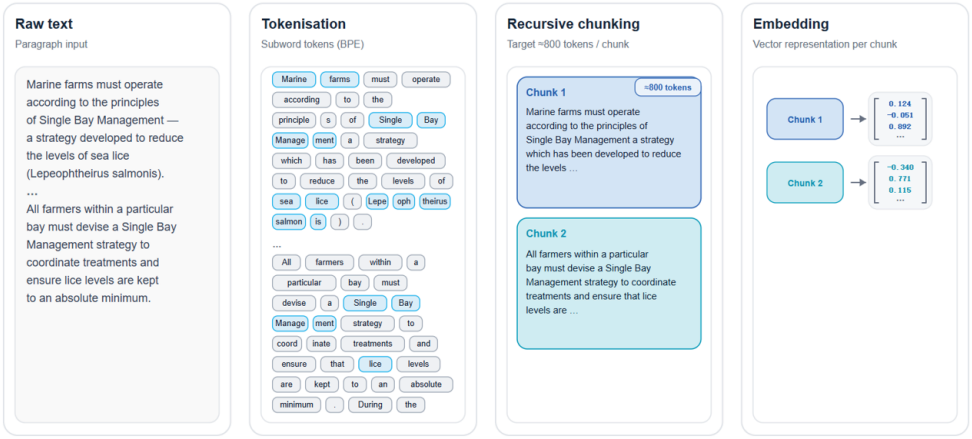


Figure 6.6: Sentence-preserving chunking and embedding pipeline. The figure shows how text is processed for retrieval. From left to right, the text is represented as raw text, split into word and punctuation tokens during tokenization, recursively grouped into coherent chunks with a target chunk size of approximately 800 tokens, and adjusted to the nearest sentence boundary to reduce fragmentation. Each resulting chunk is then converted into a separate embedding vector for vector retrieval and semantic matching.

Farm-level operational data In addition to static knowledge, S-MADSS can incorporate farm-level operational context. In this study, farm data were sourced from the Norwegian Fish Health platform,[5] which provides publicly available aquaculture information about salmon farms. The data include locality identifiers, geographic coordinates, and management area descriptors, as well as weekly records of salmon lice levels and reported treatment histories. When used, these records are summarized into a compact farm-context representation that conditions evidence retrieval and response synthesis in case-specific queries.

6.3 Evaluation

We evaluate S-MADSS using a component-to-workflow strategy covering three core capabilities: evidence retrieval, evidence-grounded answer generation, and scenario-based risk identification, followed by a real-farm case study for workflow demonstration. Because these capabilities impose different reasoning and evidence requirements, we construct separate datasets and use task-specific metrics for each evaluation setting.

[5] `https://www.barentswatch.no` (As of March 2026).

6.3.1 Research questions

The evaluation is designed to address the following research questions:

1. Which retrieval configuration provides the strongest evidence foundation for compliance-oriented QA in S-MADSS? This question assesses how various retrieval strategies impact the retrieval of relevant evidence.

2. How do different LLM backbones use retrieved evidence to generate evidence-grounded answers? This question examines how model choice influences the integration and verbalization of retrieved evidence.

3. Does local context expansion improve evidence integration for multi-hop (multi-step reasoning) questions in S-MADSS? This question examines whether local context expansion facilitates the recovery of fragmented evidence distributed across multiple segments.

4. How does evidence coverage affect risk identification in the decision-support workflow of S-MADSS? This question examines how retrieval completeness influences risk detection and aggregation in decision scenarios.

5. Does S-MADSS support an end-to-end, traceable decision-support workflow in a real-farm setting? This question assesses system-level feasibility through a case-based demonstration.

6.3.2 Dataset construction

Because the five research questions (RQ1-5) target different system capabilities, no single dataset is sufficient for all evaluations. As summarized in Table 6.2, we therefore construct separate datasets to isolate the capability of interest in each setting. The single-hop QA dataset is used to evaluate retrieval quality (RQ1) and evidence-grounded generation under controlled evidence conditions (RQ2). The multi-hop QA dataset is used to test whether local context expansion improves evidence integration across fragmented supporting chunks (RQ3). The risk identification dataset is used to evaluate decision-support behavior in recall-sensitive regulatory scenarios (RQ4). In addition, we include a real-world farm case study to demonstrate the feasibility and traceability of the end-to-end workflow (RQ5). To reduce obvious annotation noise, the first author manually inspected the generated samples and removed samples with unclear evidence linkage, inconsistent constraint descriptions, or incomplete traceability. A representative example from the risk identification dataset is provided in Table 6.3, and additional examples are included in the Appendix (Tables C.1 and C.2).

Table 6.2: Summary of evaluation datasets.

Dataset	Purpose	Source	Samples	Evaluation Aspect
Single-hop QA	Retrieval and answer grounding	All documents	230	Retrieval & correctness
Multi-hop QA	Context expansion evaluation	All documents	202	Retrieval & correctness
Risk identification dataset	Risk identification	Regulatory documents	100	Risk recall
Case study	System demonstration	Fish Health Platform	1 site	Qualitative analysis

(i) Single-hop QA We construct 230 single-hop QA pairs from the domain knowledge base, where each question and reference answer is strictly grounded in a single sampled chunk. This dataset is used to evaluate head retrieval accuracy (Hit@k) and answer correctness under different numbers of retrieved evidence chunks.

(ii) Multi-hop QA We construct 202 multi-hop QA pairs that require integrating evidence across multiple supporting chunks. For each sample, the minimal supporting chunk set is identified and verified against the source documents. This dataset is used to evaluate evidence coverage (Recall@k) and the effect of local context expansion under fragmented evidence.

(iii) Risk identification dataset We construct 100 scenario-based risk identification samples from regulatory documents. For each sampled chunk, we generate a management scenario and introduce deliberate violations of the stated constraints (e.g., treatment intervals, distance restrictions, documentation requirements), which are annotated as gold risk points. This dataset is used to evaluate Risk Recall.

(iv) Case study We conduct a single-site case demonstration using publicly-available lice monitoring and treatment records from the Norwegian Fish Health platform to illustrate workflow feasibility. The output of the S-MADSS is provided in C.2.

6.3.3 Experimental setup

Different LLMs were assigned to different agents and knowledge construction in the system to balance generation quality, orchestration efficiency, and evaluation consistency.

RAG-based evidence generation For evidence-grounded answer generation, we evaluated four instruction-tuned LLM backbones: `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024), `Gemma 3-12B`, `Gemma`

[6] `https://faolex.fao.org/docs/pdf/nor82122.pdf` (As of March 2026).

Table 6.3: Example from the risk identification dataset illustrating regulatory non-compliance risk identification.

Scenario	Current condition: locality 20345 has an ongoing feed-based lice treatment with a chitin synthesis inhibitor eight weeks after the previous chitin-inhibitor treatment. The site is located 600 meters from a registered shrimp fishing field. Operation of medicated-feed lice treatment is in progress. What are the risks in this scenario?
Gold risk points	1. The site is administering a chitin synthesis inhibitor only 8 weeks after the previous course, which is less than the required minimum interval of six months between such treatments at the same locality. 2. The treatment is being carried out at a locality that is only 600 meters from a registered shrimp fishing field, which is closer than the required 1000-meter minimum distance for using chitin synthesis inhibitors.
Potential consequence (not scored)	These deviations increase the risk of unacceptable environmental effects on nearby shrimp resources and represent non-compliance with lice-treatment distance and frequency restrictions.
Source chunk	*Chitin synthesis inhibitors administered through feed may not be used on the same locality until at least six months have passed since the last such treatment, irrespective of which inhibitor was used. Chitin synthesis inhibitors may not be used on localities that are closer than 1000 meters to shrimp fishing fields, as shown in the Directorate of Fisheries' online map tool.*[6]

`3-27B` (Team et al., 2025), and `Qwen 3 (30B)` (Yang et al., 2025). These models were used to generate responses conditioned on retrieved evidence chunks and to support comparisons of grounded answer quality under the same retrieval setting.

Agent execution Workflow-level agent components, including query routing, coordination, and risk identification, were implemented using `MIMOv2-flash` (Team et al., 2026). This model was selected for lightweight and stable structured generation in the multi-agent workflow.

Evaluation via LLM-as-Judge Factual correctness and evidence consistency are assessed using an LLM-as-Judge protocol implemented with `DeepSeek-v3.2` (DeepSeek-AI et al., 2025). The judge model assessed generated outputs against either reference answers or retrieved evidence, depending on

the evaluation task. Prior work indicates that strong reasoning-oriented LLMs can provide reliable and human-aligned judgments for factual and evidence-grounded evaluation tasks (Badshah & Sajjad, 2025; Wang et al., 2025b; Zhang et al., 2025a).

Dataset construction The evaluation datasets were constructed using `DeepSeek-v3.2` under controlled prompting conditions. The generated candidates were filtered for basic validity criteria, including the presence of traceable evidence identifiers, internally consistent constraint descriptions, and sufficiently specific scenario formulations for evaluation.

6.3.4 Evaluation metrics

We evaluate the system using six metrics for different experiments: Hit@k, Recall@k, correctness, faithfulness, risk recall, and evidence chunk recall.

Hit@k Hit@k measures whether the system retrieves the gold-standard evidence chunk within the top-k ranked results. Each evidence chunk in the knowledge base is associated with a unique identifier. Let the retrieval evaluation set contain N samples. For sample i, the system returns a ranked list of retrieved chunk identifiers $\mathcal{R}_i^{(k)} = \{r_{i1}, \ldots, r_{ik}\}$, and the gold evidence chunk identifier is denoted by g_i.

A retrieval is considered successful if the gold chunk identifier appears in the top-k retrieved results. Hit@k is defined as:

$$\text{Hit@k} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[g_i \in \mathcal{R}_i^{(k)}\right]. \tag{6.13}$$

where $\mathbf{1}[\cdot]$ is the indicator function, which equals 1 if the condition is true and 0 otherwise.

Recall@k For multi-hop questions, multiple evidence chunks may be required to fully support the correct answer. Let the multi-hop evaluation set contain N samples. For sample i, the set of gold evidence chunk identifiers is denoted by G_i, and the top-k retrieved chunk identifiers are $\mathcal{R}_i^{(k)}$.

Recall@k measures the proportion of gold evidence chunks that are successfully retrieved within the top-k results:

$$\text{Recall@k} = \frac{1}{N} \sum_{i=1}^{N} \frac{|G_i \cap \mathcal{R}_i^{(k)}|}{|G_i|}. \tag{6.14}$$

Table 6.4: Ground-truth-based correctness evaluation rubric.

Score	Category	Criteria
5	Correct	All factual statements supported by ground truth; no missing key facts; no incorrect claims.
4	Mostly Correct	All key facts correct; minor details missing or slightly imprecise; no contradictions.
3	Partially Correct	At least one key fact correct; at least one key fact missing, incorrect, or ambiguous.
2	Mostly Incorrect	Most key facts incorrect or missing; only minor facts correct; may include unsupported claims.
1	Incorrect	No key facts correct or direct contradiction with ground truth.

LLM-as-Judge protocol Correctness and faithfulness were evaluated using an LLM-as-Judge protocol implemented with `DeepSeek-v3.2` (DeepSeek-AI et al., 2025). For both metrics, the judge assigns a score on a five-point rubric, which is then linearly rescaled to $[0, 1]$ for analysis. Correctness is assessed against the reference answer, whereas faithfulness is assessed against the retrieved evidence.

Correctness Correctness measures whether the generated answer is factually consistent with the ground-truth answer (Gao et al., 2025). For each sample, the judge compares the query, reference answer, and generated answer, focusing on factual accuracy, completeness of key elements, and the absence of contradictions. The raw 1–5 judgment (Table 6.4) is rescaled to $c_i \in [0, 1]$. The overall correctness is computed as:

$$\text{Correctness} = \frac{1}{N} \sum_{i=1}^{N} c_i. \tag{6.15}$$

Faithfulness Faithfulness measures whether the generated answer is factually supported by the retrieved chunks (Tamber et al., 2025). For each sample, the judge compares the query, retrieved chunks, and generated answer, focusing on whether all factual claims are supported by the retrieved evidence and whether any claim contradicts that evidence. The raw 1–5 judgment (Table 6.5) is rescaled to $f_i \in [0, 1]$. The overall faithfulness score is computed as:

Table 6.5: Faithfulness evaluation rubric (1–5).

Score	Category	Criteria
5	Fully Faithful	All factual claims explicitly supported by retrieved chunks.
4	Mostly Faithful	All key claims supported; minor details weakly supported or loosely inferred.
3	Partially Faithful	Some claims supported; at least one key claim unsupported or ambiguous.
2	Low Faithfulness	Most claims unsupported; relies on world knowledge or speculation.
1	Not Faithful (Hallucinated)	Claims unsupported or contradict retrieved chunks.

$$\text{Faithfulness} = \frac{1}{N}\sum_{i=1}^{N} f_i. \tag{6.16}$$

Risk recall For risk evaluation, let the risk dataset contain N scenarios. For scenario i, the gold risk-point set is denoted by $\mathcal{T}_i$, and the predicted risk-point set is denoted by $\hat{\mathcal{T}}_i$. Risk points are matched under semantic equivalence, where different surface expressions describing the same underlying risk are treated as identical. Risk recall is calculated as:

$$\text{Risk Recall} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|\hat{\mathcal{T}}_i \cap \mathcal{T}_i\right|}{|\mathcal{T}_i|}. \tag{6.17}$$

Evidence chunk recall In the risk-identification setting, evidence chunk recall quantifies the coverage of constraint-bearing gold evidence available to the risk-assessment agents. For each scenario i, let G_i^{risk} denote the annotated gold evidence chunk set and let $\hat{G}_i$ denote the set of retrieved evidence chunks. Evidence chunk recall is defined as

$$\text{Evidence Chunk Recall} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|G_i^{\text{risk}} \cap \hat{G}_i\right|}{\left|G_i^{\text{risk}}\right|}, \tag{6.18}$$

6.4 Results

To evaluate S-MADSS as an end-to-end decision-support system rather than a standalone QA model, we organize the experiments as a progressive capability chain. Specifically, we examine whether the system can (i) retrieve relevant evidence, (ii) generate evidence-grounded answers, (iii) integrate fragmented evidence, (iv) identify risks under decision scenarios, and (v) support a traceable real-farm workflow. Each experiment therefore builds on the previous one, moving from evidence access to risk-aware decision support.

6.4.1 Evidence pipeline for grounded QA

6.4.1.1 Retrieval performance under different retrieval–reranking configurations

This experiment assesses which retrieval configuration provides the most reliable evidence foundation for compliance-oriented grounded QA. Using the single-hop QA dataset, we compare BM25, vector retrieval, and a hybrid retriever (BM25 + vector), each evaluated with and without cross-encoder reranking under different reranking depths. Table 6.6 reports Hit@k results and the mean score across the evaluated k values. Among all settings, the hybrid retriever with Top-100 reranking achieves the highest mean Hit@k score (0.910), outperforming the best reranked BM25 configuration (mean: 0.890) and the best reranked vector configuration (mean: 0.819). It also attains the best head and tail performance, with Hit@1 = 0.752 and Hit@20 = 0.974.

Without reranking, the BM25 retriever consistently outperformed the vector-based retriever across all Hit@k levels (e.g., Hit@1: 0.478 vs. 0.248; Hit@20: 0.878 vs. 0.774). The hybrid retriever underperformed BM25 at smaller k values (Hit@1 = 0.378) but surpassed it as k increased, reaching 0.865 at Hit@10 and 0.917 at Hit@20.

Cross-encoder reranking improves all retrieval strategies in both early-rank performance and mean Hit@k. Under Top-100 reranking, the mean score increases from 0.592 to 0.819 for vector retrieval, from 0.762 to 0.890 for BM25, and from 0.762 to 0.910 for the hybrid retriever. At the top rank, Hit@1 increases from 0.248 to 0.657 (vector), from 0.478 to 0.748 (BM25), and from 0.378 to 0.752 (hybrid). Under the same reranking setting, the hybrid configuration attains the highest mean score and top-rank accuracy among the compared methods.

Table 6.6: Hit@k comparison of retrieval strategies under different reranking configurations. The best results for each metric are highlighted in bold.

Retrieval configuration	Hit@1	Hit@5	Hit@10	Hit@15	Hit@20	Mean
Vector Only	0.248	0.539	0.657	0.744	0.774	0.592
Vector with rerank (Top-20)	0.557	0.722	0.757	0.765	0.774	0.715
Vector with rerank (Top-50)	0.630	0.787	0.826	0.830	0.844	0.783
Vector with rerank (Top-100)	0.657	0.817	0.865	0.874	0.883	0.819
BM25 Only	0.478	0.757	0.835	0.861	0.878	0.762
BM25 with rerank (Top-20)	0.709	0.844	0.878	0.878	0.878	0.837
BM25 with rerank (Top-50)	0.735	0.874	0.904	0.935	0.944	0.878
BM25 with rerank (Top-100)	0.748	0.891	0.922	0.935	0.952	0.890
Hybrid	0.378	0.748	0.865	0.900	0.917	0.762
Hybrid with rerank (Top-20)	0.735	0.883	0.913	0.917	0.917	0.873
Hybrid with rerank (Top-50)	0.735	0.891	**0.952**	0.957	0.965	0.900
Hybrid with rerank (Top-100)	**0.752**	**0.909**	0.948	**0.965**	**0.974**	**0.910**

6.4.1.2 Effect of LLM backbones on evidence-grounded answer generation

In this experiment, we fix the retrieval configuration to the best-performing setting from the previous section (hybrid retrieval with Top-100 reranking) and evaluate the evidence-grounded generation capability of different LLMs. For each query, the model receives the top-m reranked chunks as input.

As shown in Fig. 6.7, all models benefit substantially from retrieved evidence. The largest gain occurs when moving from $m = 0$ to a small amount of retrieved chunks, after which performance improves more gradually. For example, Qwen 3 (30B) increases from 0.463 at $m = 0$ to 0.684 at $m = 1$, reaches 0.785 at $m = 5$, and then remains relatively stable up to $m = 20$ (0.808). Across the evaluated settings, Qwen 3 (30B) achieves the highest overall correctness. The same pattern is observed for the other backbones. Llama 3.1 (8B) increases from 0.334 at $m = 0$ to 0.657 at $m = 1$ and peaks at 0.760 at $m = 5$. Gemma 3 (12B) also improves with added evidence and reaches 0.761 at $m = 20$. Gemma 3 (27B) shows steady gains up to $m = 10$ and attains 0.774 at $m = 20$.

Although all models benefit from retrieved evidence, their behavior is not uniform at the sample level. Relative to Qwen 3 (30B), Llama 3.1 (8B) achieves higher Correctness on 46 samples (20.0%), and Gemma 3 (27B) does so on 48 samples (20.9%) (Fig 6.8). Table 6.7 provides a representative case where the retrieved evidence contains an explicit numerical guideline (15 mg/L TSS), but only Gemma 3 (27B) verbalizes the numeric limit as the recommended operational threshold in the final

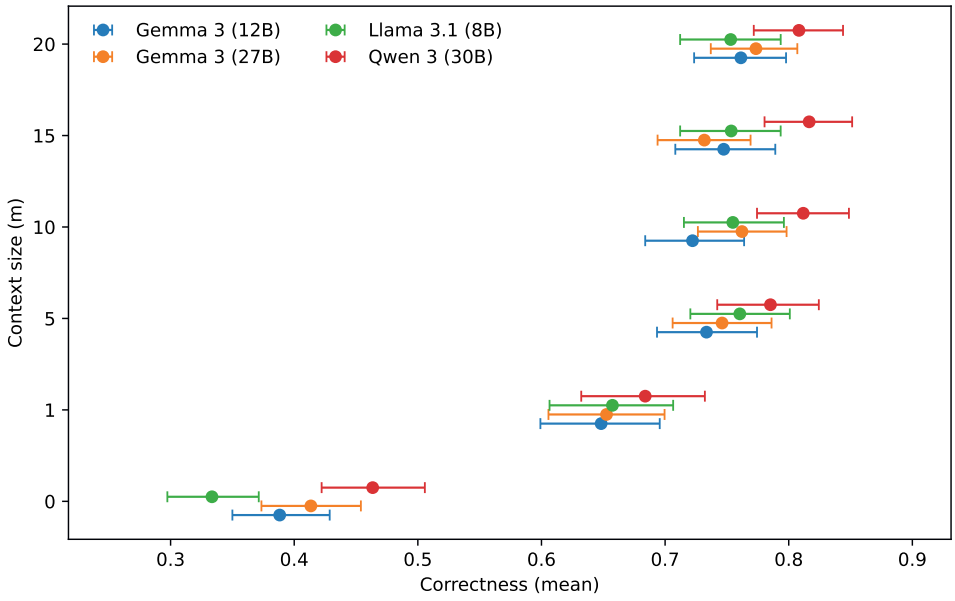


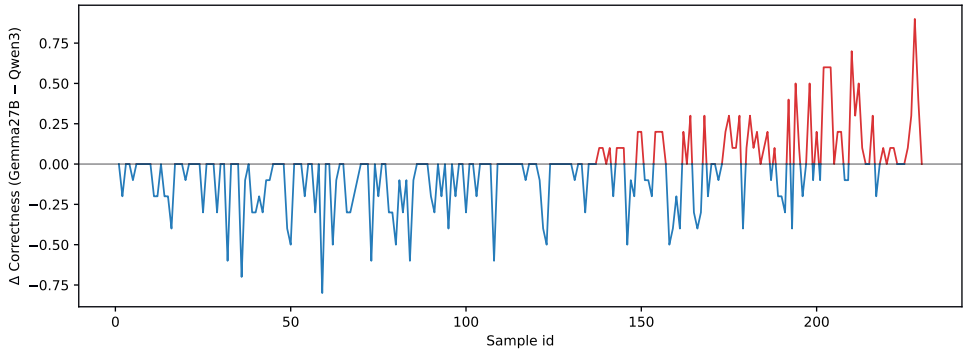
Figure 6.7: Correctness across LLMs under different numbers of context size (m). Points indicate the mean Correctness over all test samples, and horizontal error bars show the 95% confidence intervals (bootstrap).

answer.

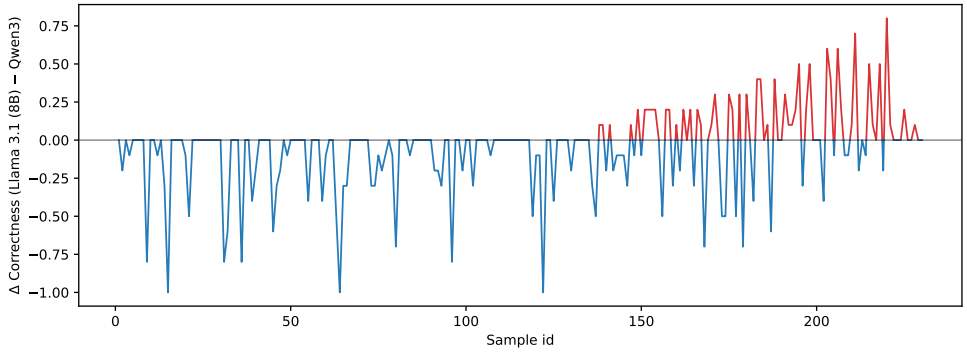
6.4.2 Effect of local context expansion on multi-hop evidence integration

Building on the best-performing configuration identified in the previous section (hybrid retrieval with Top-100 reranking and Qwen 3 (30B) for generation), we evaluate whether local context expansion improves evidence integration for multi-hop QA. We compare the base-chunk baseline (w=0) with symmetric expansion windows (±1, ±2, and ±3). Fig. 6.9 and Fig. 6.10 report retrieval and generation performance under these settings.

Retrieval coverage improves consistently with expansion. Both Hit@k and Recall@k increase as the window widens, with the largest gains appearing in early ranks and diminishing at larger k. For instance, Hit@1 rises from 0.540 (base) to 0.619 (±1) and continues to improve under wider windows; Hit@10 increases from 0.728 (base) to 0.871 (±2) and 0.881 (±3). At the tail, improvements are smaller but still positive (Hit@20: 0.896 to 0.906). Recall@k follows the same direction: Recall@1 increases from 0.307 (base) to 0.451 (±1) and reaches 0.524 under ±3, while Recall@10 increases from 0.568 (base) to 0.752 (±2) and 0.764 (±3), and Recall@20 reaches 0.803 with



(a) Sample-level ΔCorrectness between Gemma 3 (27B) and Qwen 3 (30B). Samples are sorted in descending order by Qwen 3 performance (m=15). ΔCorrectness is computed as Gemma 3 (m=20) minus Qwen 3 (m=15); positive values indicate samples where Gemma 3 outperforms Qwen 3.



(b) Sample-level ΔCorrectness between Llama 3.1 (8B) and Qwen 3 (30B). Samples are sorted in descending order by Qwen 3 performance (m=15). ΔCorrectness is computed as Llama 3.1 (m=5) minus Qwen 3 (m=15); positive values indicate samples where Llama 3.1 outperforms Qwen 3.

Figure 6.8: Sample-level ΔCorrectness against Qwen 3 (30B) across two baselines. In both panels, samples are sorted by Qwen 3 performance (m=15).

±3.

The generation results show a different pattern. Correctness improves under moderate expansion and reaches a shared maximum of 0.856 at two configurations: ±1 with k=15 and ±2 with k=10. By contrast, faithfulness is generally highest in the base setting when more retrieved chunks are provided, and decreases as the expansion window becomes wider. At k=10, for example, faithfulness decreases from 0.892 in the base setting to 0.873 under ±1, 0.866 under ±2, and 0.786 under ±3.

Overall, local context expansion improves retrieval coverage for multi-hop questions, while moderate expansion (±1–±2) provides the strongest downstream answer cor-

Table 6.7: The retriever obtained evidence containing the suggested operational upper bound for total suspended solids (TSS, 15 mg/L), but only Gemma 3 verbalized the numeric guideline in the final answer.

Query	What is the recommended operational limit for abrasive particles in water to protect fish's respiratory health?
Gold answer	An upper limit of 15 mg/L for total suspended solids (TSS) has been suggested for operational consideration to protect salmon from gill damage and compromised oxygen transfer.
Top retrieved chunks by RAG	(1) "Suspended solids in the water column can contribute to increased chemical and biological oxygen demand...No universally accepted threshold for acceptable TSS exists for salmon production, *but an upper limit of 15 mg/L has been suggested for operational consideration*" (2) "Measurement of total gas pressure is used to assess both the total gas pressure and the amount and saturation of dissolved nitrogen gas in the water..." ...
Qwen 3 (30B) *Correctness: 0.4*	States that no specific operational limit is identified for abrasive particles, although it acknowledges the risk to gill health. Mentions the 15 mg/L threshold but fails to assert it as the recommended limit.
Gemma 3 (27B) *Correctness: 1.0*	Correctly identifies the 15 mg/L limit for total suspended solids as the operational guideline to minimize gill damage.

rectness.

6.4.3 Risk identification under different evidence coverage settings

To evaluate decision-support behavior beyond QA, we examine how the three specialized risk-assessment agents (regulation, welfare, and feasibility) identify risks under different evidence-coverage settings. All agents share the same RAG pipeline (hybrid retrieval with Top-100 reranking and Qwen 3 (30B) for generation), while the local context expansion strategy (base, ± 1, ± 2, and ± 3) is varied to control evidence coverage. As a retrieval-free baseline, we also evaluate direct LLM-based risk identification without retrieved evidence.

The results are summarized in Fig. 6.11. Overall, the evidence-grounded workflow substantially outperforms the retrieval-free baseline, improving Risk Recall from 0.637 to a maximum of 0.950. Increasing the retrieval context from k=15 to k=20 consis-

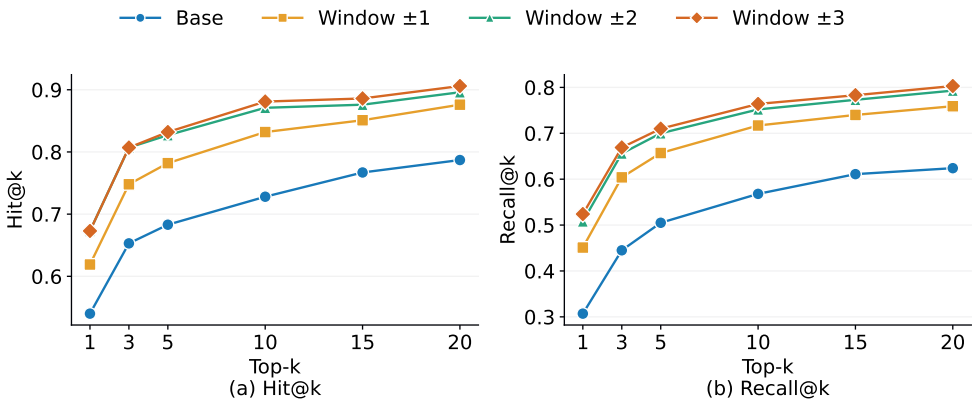


Figure 6.9: Retrieval performance for multi-hop QA under different local context expansion windows. The base setting ($w = 0$) is compared with symmetric expansion windows (±1, ±2, and ±3) using Hit@k and Recall@k.

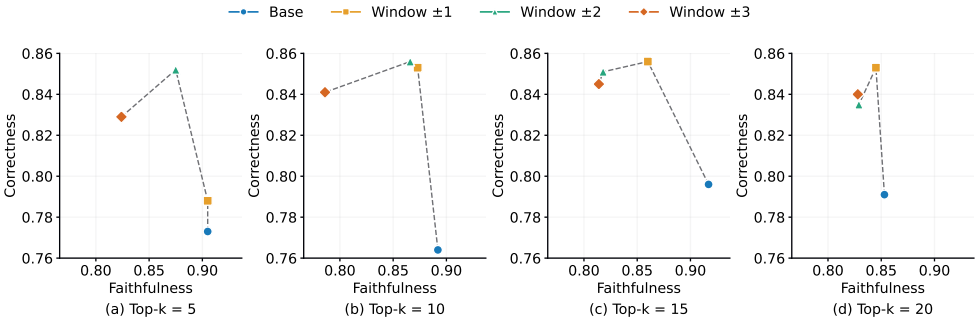
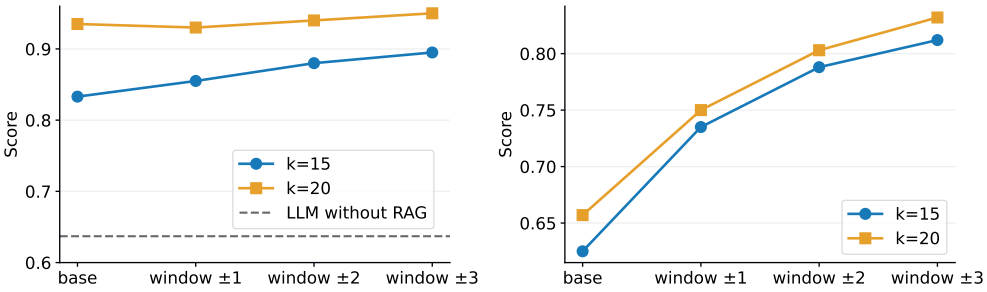


Figure 6.10: Generation performance for multi-hop QA under different local context expansion windows in terms of correctness and faithfulness across retrieval budgets and expansion settings.

tently improves Risk Recall across the evaluated settings.

Retrieved-evidence recall also increases with broader context expansion. The highest Risk Recall is obtained under the ±3 setting with k=20, which reaches 0.950 together with the highest retrieved-evidence recall (0.832). Compared with the multi-hop QA setting, the best risk-identification performance is obtained under broader expansion and a larger number of retrieved evidence chunks.

In summary, both larger number of retrieved chunks and local context expansion improve risk identification, and the best observed performance is obtained at k=20 with broader expansion (±3).



(a) Risk recall

(b) Retrieved evidence recall

Figure 6.11: Risk identification performance using regulation, welfare, and feasibility of risk-identification agents under different RAG configurations.

Table 6.8: Key farm indicators for the case study.

Indicator	Value	Status
Adult female lice (AF)	0.17	Near regulatory limit
Mobile lice	0.26	Moderate level
Stationary lice	0.00	Stable
Sea temperature (°C)	7.73	Slightly increasing
Regulatory AF limit	0.2	Threshold

6.4.4 Real-world workflow demonstration

We present a single-site farm case from Barentswatch[7] to illustrate the end-to-end workflow of S-MADSS under realistic operational conditions. In this case, the user query concerns whether treatment for salmon lice should be conducted in Week 18. This case study serves as a qualitative demonstration of workflow feasibility and output traceability. Using publicly available salmon-lice monitoring and treatment records from the Norwegian Fish Health platform, the system transforms farm observations into a structured farm-context report and generates an evidence-grounded recommendation with explicit assumptions, monitoring triggers, contingency options, and evidence-linked constraints. The more detailed system output is provided in C.2.

Table 6.8 summarizes the key indicators relevant to the decision. These indicators are automatically derived by the farm-environment report tool, which aggregates weekly observations (lice counts, temperature, and regulatory limits) into structured descriptors such as compliance status, temporal trends, and risk-relevant signals for downstream decision support.

[7] `https://www.barentswatch.no` (As of March 2026).

Based on these inputs, the system first analyzes the current compliance status and temporal lice trends, identifying a sustained upward trajectory in adult female lice. During the analysis, the system identifies that the observation period falls within the spring regulatory window, in which a stricter threshold of 0.2 adult female lice applies. This seasonal constraint indicates a reduced tolerance margin and increases the risk of near-term non-compliance compared with the standard regulatory limit.

Given the query regarding treatment planning for Week 18, the system generates an evidence-grounded recommendation for proactive intervention. The recommended action is to conduct a freshwater bath within the next 7 days (week 18–19) to reduce adult female lice and maintain counts below the regulatory threshold. In addition to the primary recommendation, the system provides operational implementation guidance, monitoring requirements with trigger thresholds for post-treatment verification, and a contingency option (hydrogen peroxide bath) if the treatment effect proves insufficient. The case output also includes an explicit risk evaluation, in which the proposed action is audited from regulatory, welfare, and operational-feasibility perspectives, and the main residual risks and mitigation needs are stated alongside the recommendation. The system response links key decision elements to retrieved evidence and highlights potential compliance, operational, and welfare risks implied by the current scenario.

6.5 Discussion

This study presents a compliance-oriented decision-support workflow for Atlantic salmon sea-cage farming. Rather than optimizing a single module, the framework integrates evidence retrieval, evidence-grounded answer generation, local context expansion, and agent-based risk auditing into a unified and traceable pipeline. The results show that these components work together to support constraint-grounded reasoning across both QA and scenario-based decision tasks. By linking outputs to retrievable document chunks, S-MADSS improves transparency in how regulatory requirements, welfare considerations, and operational constraints are interpreted. The case study further suggests that incorporating farm-level contextual information can make recommendations more operationally grounded and audit-ready. Taken together, these findings suggest that S-MADSS constitutes a viable proof-of-concept for evidence-grounded and compliance-oriented decision support in aquaculture.

6.5.1 Analysis of retrieval performance and reranking

The results suggest that evidence retrieval for compliance-oriented aquaculture tasks depends on multiple forms of relevance rather than a single matching criterion. In

this setting, relevant evidence is often expressed through constraint-bearing language such as thresholds, treatment conditions, chemical names, and procedural requirements, while similar content may also be paraphrased across regulations, manuals, and scientific literature. As a result, retrieval quality depends not only on topical similarity, but also on whether the system can recover the exact constraints that govern operational decisions.

This helps explain why BM25 performs strongly in the initial retrieval stage. Many task-critical statements in this setting depend on exact lexical matching, because small wording differences may alter regulatory interpretation. BM25 is based on term-frequency and inverse-document-frequency weighting, which favors exact matches on informative and relatively rare terms while accounting for document length (Robertson & Zaragoza, 2009). This makes it well suited to retrieving chunks that contain constraint-bearing terms, threshold values, and numeric details (Kamalloo et al., 2024). However, the evidence space is also lexically heterogeneous, and similar requirements may be expressed differently across regulations, manuals, and scientific literature (Zhao et al., 2025). Vector-based retrieval complements this behavior by mapping queries and documents into a shared embedding space, allowing semantically related evidence to be retrieved even when the surface wording differs (Karpukhin et al., 2020).

The advantage of hybrid retrieval is therefore not only higher aggregate performance, but also better alignment with the structure of the knowledge base. It combines lexical precision for constraint matching with semantic recall across heterogeneous formulations (Cormack et al., 2009; Mandikal & Mooney, 2024). Cross-encoder reranking improves this pipeline through a different mechanism. Whereas first-stage retrievers score documents largely from independently computed query and document representations, a cross-encoder jointly encodes each query-chunk pair and models fine-grained token-level interactions (Nogueira & Cho, 2019). This allows it to distinguish chunks that are merely topically similar from those that actually satisfy the query's applicability conditions. In our setting, this is especially important for identifying exceptions, scope restrictions, threshold–unit combinations, and other compositional constraints that often determine whether a piece of evidence is valid for downstream use. This additional precision comes at a computational cost, because cross-encoders cannot precompute document representations independently of the query; each query-chunk pair must be processed jointly at inference time. For that reason, reranking is most effective as a second-stage component applied to a limited candidate set returned by the initial retrievers. In this architecture, the first stage maximizes coverage across different forms of relevance, while the second stage enforces finer-grained applicability. This refinement is especially important for evidence-grounded reasoning, because

omitted or misaligned constraints can propagate into downstream interpretation and decision errors. These findings therefore answer RQ1 by showing that hybrid retrieval with reranking provides the most reliable evidence foundation for S-MADSS.

6.5.2 Evidence summarization: impact of LLM backbones

Across the four LLM backbones, the performance of evidence-constrained generation can be explained by two interacting factors: external evidence availability and internal evidence interpretation. The former concerns whether retrieved chunks contain sufficient task-relevant constraints; the latter concerns how the model selects, integrates, and verbalizes that evidence.

From the perspective of external evidence availability, the strong quality gain obtained from even a small number of retrieved chunks suggests that the main bottleneck is access to relevant evidence rather than language modeling capacity alone. A limited amount of highly relevant evidence is often sufficient to shift generation from prior-driven inference to retrieval-grounded reasoning, producing the largest marginal improvement (Kozhipuram et al., 2025). However, these gains plateau as more relevant chunks are added. Larger contexts therefore do not produce linear improvements, because additional evidence can introduce redundancy, inconsistency, and scope misalignment. Under such conditions, performance depends more on evidence density and ranking quality than on context volume alone (Vladika & Matthes, 2025).

Among the evaluated backbones, Qwen 3 (30B) achieved the best overall performance. However, model scale alone does not determine effectiveness. Sample-level analysis shows that Gemma3 (27B) and even Llama 3.1 (8B) outperform Qwen3 (30B) on some queries. This variation likely reflects differences in training data, alignment, and optimization objectives, which can affect generation behavior (Han et al., 2025; Wu et al., 2024). Prior work also shows that instruction tuning and alignment can influence assertion strength and decision preferences in LLMs (Shaikh et al., 2024). When retrieved evidence is only indirectly related to the target concept or when conceptual boundaries are ambiguous, models may apply different standards of extrapolation. More conservative models may remain closer to the explicit scope of the evidence and weaken assertions when direct support is limited. In contrast, models trained on broader instruction-following or summarization data may be more willing to abstract and synthesize related evidence into explicit numerical statements or consolidated conclusions (Wu et al., 2024; Hall et al., 2026). Therefore, when retrieval results are the same, output differences often arise from how models define the scope of applicable evidence rather than from retrieval failure. Table 6.7 illustrates this pattern.

From a system-design perspective, these findings suggest that stronger backbones remain preferable when high-quality grounded summarization is required, while smaller models may still be useful in cost-sensitive or supporting roles. More broadly, the results indicate that improving grounded generation depends on both retrieval quality and backbone choice: retrieval determines whether the relevant evidence is available, whereas the backbone determines how effectively that evidence is converted into a usable answer. Overall, this answers RQ2 by showing that retrieved evidence substantially improves grounded answering across backbones, and that Qwen 3 (30B) provides the strongest overall performance, although backbone-specific variation remains.

6.5.3 Impact of local context expansion

The results on multi-hop questions show that local context expansion improves both retrieval and generation. As the window increases from the base setting to ±3, Hit@k and Recall@k consistently rise, indicating that adjacent chunks often contain complementary information required for multi-hop reasoning (Fig. 6.9). Without expansion, segmentation boundaries may separate logically connected evidence. Local expansion, therefore, functions as a structural repair mechanism that partially restores discourse continuity. Notably, even a ±1 window yields substantial gains, suggesting that critical supporting evidence typically co-occurs within a limited local neighborhood. However, window size introduces a trade-off between correctness and faithfulness. Correctness improves with expansion and peaks in the ±1–±2 range, while faithfulness declines as the window grows, particularly at ±3. This pattern indicates that expansion increases evidence coverage but reduces information density by introducing weaker or background content (Vladika & Matthes, 2025). As competing information increases, key evidence becomes less salient, leading to lower alignment scores. Thus, retrieval coverage does not translate monotonically into generation quality, as performance is constrained by both evidence completeness and attention allocation capacity.

From a system perspective, local expansion can be viewed as a minimally sufficient repair strategy. A ±1 window provides substantial gains, ±2 yields diminishing returns, and ±3 introduces noticeable faithfulness degradation. Moderate expansion (±1 or ±2), combined with control over the number of evidence chunks or deduplication, offers a better balance between coverage and alignment stability. These results answer RQ3 by showing that local context expansion improves multi-hop evidence integration, with moderate expansion providing the best trade-off.

6.5.4 Risk identification

The risk identification experiments demonstrate that the decision module within the S-MADSS exhibits stable risk detection capability. During decision generation, the regulation, welfare, and feasibility agents invoke the RAG pipeline to retrieve relevant clauses and identify potential risk factors based on the retrieved evidence. This indicates that the system does not merely produce decision recommendations, but also provides explicit, evidence-grounded risk assessments. As a result, identified risks are traceable to specific textual sources, enhancing the structural transparency of the decision process.

In terms of metric relationships, retrieved evidence recall and risk recall are positively correlated overall, but not linearly aligned. When increasing the retrieval depth from $k = 15$ to $k = 20$, evidence recall improves only marginally, whereas risk recall increases substantially. This discrepancy may be explained by semantic redundancy in the dataset. Since the corpus is constructed from regulation-related documents within a knowledge base, similar or equivalent risk descriptions may appear across different files. Therefore, even if a specific annotated chunk is not retrieved, the agent may still detect the corresponding risk when semantically equivalent formulations are present. Moreover, once partial risks are identified, the agents may adjust their focus or retrieval emphasis, introducing a degree of dynamic refinement in the identification process. Consequently, risk recall reflects the system's ability to integrate and aggregate risk signals at the semantic level, rather than solely its ability to retrieve particular annotated text segments.

Compared with multi-hop QA, risk identification exhibits a different optimization profile. QA tasks emphasize precise alignment between answers and specific evidence, whereas risk detection primarily concerns avoiding omission of critical constraints and is therefore more recall-sensitive.

From a decision-support perspective, the agent-based, evidence-grounded risk identification mechanism enables the system to present decision recommendations together with explicitly supported risk assessments, helping decision-makers to evaluate trade-offs based on documented constraints. Moreover, by incorporating organization-specific regulations or internal policies, the framework can adapt risk evaluation to institutional preferences, providing a customizable and evidence-grounded approach to decision support. This answers RQ4 by showing that risk identification in S-MADSS depends strongly on evidence coverage and benefits substantially from the evidence-grounded auditing workflow.

6.5.5 Case study implications for practical decision support

The case study complements the offline QA and risk-identification experiments by illustrating how the proposed components interact within a realistic decision-support workflow. Rather than serving as a statistical validation of system performance, the case demonstration shows that farm-level monitoring records can be transformed into a structured contextual representation, linked to retrieved evidence, and incorporated into a recommendation that explicitly states assumptions, risks, and evidence references. The example demonstrated the operational coherence of the workflow.

At the same time, the system's recommendation of a tarpaulin-based freshwater treatment should be understood as one feasible option rather than a universally optimal format. In practice, the final choice still depends on the farm's specific conditions, existing infrastructure, operational constraints, and internal practices. More context-sensitive decision support would therefore benefit from extending the knowledge base beyond public documents to include farm-specific protocols, internal guidelines, and management preferences. This addresses RQ5 by demonstrating that S-MADSS can support a traceable end-to-end workflow in a realistic farm setting.

6.5.6 Limitations and future work

Despite the demonstrated effectiveness of S-MADSS in evidence retrieval, evidence-grounded generation, and risk identification, three primary limitations warrant further investigation.

First, the system depends on the quality, timeliness, and consistency of external data sources. Missing records, delayed reporting, outdated regulations, and inconsistencies across sources may reduce recommendation reliability. In particular, conflicts may arise between sources of different authority. In addition, regulatory updates are often scattered across revisions and related documents, making current requirements difficult to identify. Future work should improve evidence prioritization, especially by assigning greater weight to legally binding and official sources, and should support automated detection of regulatory changes.

Second, the current knowledge base is limited by the scope and detail of available resources. Public materials often provide only general guidance and may not offer the level of operational detail needed for farm-level compliance support. In practice, copyright, licensing, and access restrictions may further limit knowledge base construction. At the same time, internal company documents, such as standard procedures and compliance guidelines, could improve decision quality if they are securely integrated. In addition, the knowledge base in this study was constructed using a task-oriented collection strategy, focusing on building a flexible evidence reposi-

tory for decision support from heterogeneous sources such as regulations, manuals, and scientific literature, rather than following a formal systematic-review protocol. Future research should therefore explore structured integration of both public and organization-specific knowledge sources.

Third, S-MADSS should be understood as a decision-support framework rather than an autonomous decision-making system. Its current implementation is still a proof of concept, not a fully mature operational DSS. Although explicit evidence citation improves transparency and traceability, regulatory interpretation may still involve contextual ambiguity and cannot replace expert judgment in complex cases. Moreover, while the present study shows architectural feasibility and promising performance under controlled conditions, broader validation across diverse farms, regulatory settings, and longer operational periods is still needed. Future work should therefore focus on human-in-the-loop validation, continuous knowledge base maintenance, clearer governance boundaries, and improved agent coordination, tool routing, and evidence selection.

At the system level, the present study should be viewed as an initial framework rather than a mature operational decision-support system. Although instantiated for Atlantic salmon farming, the framework was designed with extensibility in mind. Future development could connect it to richer farm-level data streams and analytical modules, such as high-frequency sensor measurements, computer-vision-based salmon-lice detection (Zhang et al., 2025b, 2024a; Zheng et al., 2024), salmon lice forecasting platforms, and welfare-oriented assessment tools (Zhang et al., 2026), thereby providing more timely and multi-perspective inputs for downstream decision support. In this sense, the value of the framework lies not only in the current salmon-farming application, but also in its extensibility as a general design pattern. With an updated domain knowledge base and task-specific redefinition of agent roles, the framework can be adapted to concrete decision-support problems in other agricultural and aquaculture settings, such as livestock or plant management (Neik et al., 2025). Although such transfer would still require domain-specific empirical validation, the present study provides an initial foundation for that broader development.

6.6 Conclusion

We presented S-MADSS, an evidence-grounded and compliance-oriented decision-support system for Atlantic salmon sea-cage farming, implemented as an instantiation of the broader MADSS framework for agriculture and aquaculture. This work addresses the lack of unified, evidence-grounded workflows for compliance-oriented decision support in Atlantic salmon farming. By combining a domain knowledge base

with farm-level contextual information, S-MADSS extends RAG from information access to traceable, compliance-oriented decision workflows.

Across the evaluation, hybrid retrieval with cross-encoder reranking provided the strongest evidence foundation, and retrieved evidence substantially improved grounded answer correctness across all tested LLM backbones, though model scale alone did not determine generation quality. Local context expansion improved evidence coverage and downstream answer quality by reducing the effect of chunk fragmentation, while moderate expansion provides a better balance between evidence completeness and precision. This evidence-grounded design also translated into stronger risk detection within the decision-support workflow, and a real-farm case study demonstrated that monitoring records can be converted into traceable, evidence-linked recommendations.

S-MADSS is intended to support, rather than replace, expert decision-making, and its practical reliability depends on the quality and timeliness of external knowledge and farm data. Future work should focus on broader field validation, automated updating of regulatory knowledge, and adaptation of the framework to other aquaculture and agriculture production contexts.

Chapter 7

General discussion

7.1 Summary of findings

This thesis investigates how AI-based methods can support more responsible and sustainable salmon farming management. Following the seeing–knowing–acting logic introduced in Section 1.1, the work is organized around three interrelated dimensions: farm-state perception (seeing), animal welfare assessment (knowing), and decision support (acting). These dimensions are addressed through three representative bottlenecks: detecting early planktonic salmon lice in seawater, extracting scientific statements related to animal welfare, and integrating legal, regulatory, and farm information into a modular AI-agent-based decision-support framework.

RQ1: How can AI-based methods support the automated detection of planktonic-stage salmon louse larvae in seawater under conditions of low target density, complex visual backgrounds, and limited labeled data?

RQ1 is addressed in Chapters 2, 3, and 4, which progressively move from feasibility assessment to robustness under data scarcity. Chapter 2 establishes the feasibility of automated detection by developing a rapid seawater image acquisition system, constructing an object detection dataset for planktonic-stage salmon lice, and comparing multiple detection models. The key finding is that salmon louse larvae can be detected from seawater images with high accuracy under controlled imaging conditions, particularly using RT-DETR, which achieved a precision of 97.2% and a recall of 97.8%. This shows that AI-based image detection can provide a practical and low-cost route for rapid screening of planktonic salmon lice.

Building on this feasibility study, Chapter 3 examines whether synthetic data could improve detection under more complex seawater backgrounds. The main finding is that combining synthetic data with real data improves model performance, particularly in complex seawater samples with more impurities and variable backgrounds. For example, YOLOv8n's F1-score increased from 70.6% to 87.7% after incorporating synthetic training data. A field test on real-world seawater samples further validated this feasibility, where models trained with synthetic data improved recall by 11.0%–87.0%. Importantly, the number of impurities added to synthetic images has a stronger influence on model performance than other synthesis parameters, and within the tested range, adding more impurities generally leads to better performance. This suggests that exposing the model to more complex and diverse visual backgrounds is important for improving generalization to realistic seawater samples.

Chapter 4 further examines how synthetic data could be used more effectively under data-scarce conditions by proposing Salmon Lice Detection based on Online Synthetic

Training (SDOST), an online synthetic-data training framework. SDOST introduces proximal feature replay and fast gradient sign method (FGSM)-based adversarial regularization to reduce the distributional gap between synthetic and real data, mitigate overfitting to limited real samples, and improve robustness to appearance perturbations and domain shift. These modules enable the model to make more effective use of synthetic samples during training and lead to better detection performance than fixed offline synthesis, especially in few-shot settings. Compared with offline-only training, SDOST improved F1 by 4.3%–17.7%, while yielding substantially larger gains of 13.8%–166.4% over source-only training.

Together, Chapters 2–4 demonstrate that automated planktonic lice detection is not only feasible, but can also be made progressively more robust and data-efficient. This robustness matters in practice, since natural plankton communities vary considerably across time and season, and a reliable detection model must generalize across this visual complexity. Synthetic data supports this generalization while also allowing models to be built faster, without depending solely on real data. These findings lay a methodological basis for the early detection of sparse, small, and hard-to-observe aquaculture targets more broadly.

RQ2: How can AI-based methods support the automatic extraction of welfare-related scientific statements for animal welfare modelling?

RQ2 is addressed in Chapter 5. This chapter targets the evidence bottleneck in welfare model construction. Rather than proposing a fixed salmon welfare model, it develops an NLP-based method for identifying welfare-related scientific statements from the literature. A cross-domain dataset is constructed by integrating outputs from semantic models across multiple species, including dairy cows, sows, and salmon. Because naturally occurring negative samples are limited, adversarial sampling is used to select semantically distant sentences as negative examples. Three strategies are compared: LLM-based few-shot learning, two-stage classification, and end-to-end fine-tuning of pretrained Transformer models. The key finding is that end-to-end fine-tuning achieved the best overall performance, with a precision of 96.0% and a recall of 92.4% on the main test set. On a more challenging expert evaluation dataset, where the distinction between welfare-related and non-welfare statements is more subtle, the model maintained a recall of 76.5%, though precision dropped to 59.1%, reflecting the increased difficulty of this dataset. An interactive visualization system is developed to support inspection and use of the extracted statements.

In response to RQ2, Chapter 5 shows that NLP-based methods can transform welfare-related statements scattered across unstructured literature into traceable evidence units for downstream semantic modelling. The method does not replace expert judge-

ment; rather, it functions as an evidence-screening component that can reduce manual literature screening workload and support more systematic construction and updating of welfare models.

RQ3: How can legal, regulatory, and other management-relevant information be integrated into a flexible and modular AI-based decision-support approach for salmon farming?

RQ3 is addressed in Chapter 6. This chapter proposes the MADSS framework, which consists of a workflow layer, a resource layer, a tool layer, and an agent layer. An implementation of this framework, S-MADSS, is developed to explore decision support aligned with legal and policy requirements. The system organizes different decision perspectives as specialized agents to allow regulatory compliance, risk assessment, farm-information analysis, and decision generation. Legal and policy documents, knowledge bases, and farm information are connected through function calls, enabling each agent to retrieve and integrate information according to its assigned role. A key finding of Chapter 6 is that modular agent roles and collaborative workflows can make heterogeneous decision evidence more explicit and traceable. The context retrieval strategy with local context expansion further supports this process by incorporating adjacent text after relevant knowledge fragments are retrieved, improving knowledge recall. In risk identification, the evidence-grounded auditing workflow achieved a maximum risk recall of 95.0%, compared with 63.7% for retrieval-free LLM-based identification. Through a case study on salmon lice treatment recommendations, the chapter demonstrates that the proposed framework can coordinate multiple perspectives, including treatment feasibility, welfare concerns, and regulatory compliance, within a unified workflow.

In response to RQ3, Chapter 6 shows that a multi-agent architecture can provide a feasible design pathway for structured, traceable, and evidence-based decision support in salmon farming. The contribution is mainly architectural: it demonstrates how legal, regulatory, welfare-related, and farm-specific information can be organized within a modular reasoning framework. However, the decision quality of this framework in real management contexts still requires further systematic evaluation.

Together, these three components show how different AI methods can support salmon farming management: detecting difficult-to-observe farm states, transforming dispersed welfare knowledge into evidence units, and organizing regulatory, welfare, and farm-specific information into a risk-aware decision-support workflow.

7.2 Reflections

7.2.1 Salmon lice detection

The task for planktonic stage lice detection is essentially a challenging object detection problem: salmon lice are very sparse, occupy only a small image area, have limited visible features, and are surrounded by many visually similar non-target objects, such as plankton, particles, and impurities. Synthetic data are effective mainly because they expand the diversity of target–background–non-target combinations, rather than simply enlarging dataset size. In this thesis, synthetic images are generated by extracting salmon lice, impurities, and backgrounds from real images and recombining them into new scenes. This instance-based synthesis preserves important real visual properties, including morphology, transparency, texture, and imaging noise, while increasing scene and condition variation. However, synthetic data alone cannot fully replace real data, as pasted instances may still introduce boundary artefacts, texture mismatch, or unnatural spatial relationships, causing a degree of domain shift between synthetic and real images. This is consistent with the broader synthetic-to-real gap reported in other domains (Dvornik et al., 2019), where instance-level realism does not guarantee scene-level realism. This helps explain why combining synthetic data with a small amount of real data is more effective than using synthetic data alone: real samples anchor the model to the deployment domain, while synthetic samples expand the range of training variations.

The performance of RT-DETR dropped from approximately 97.5% F1-score on clean data (Chapter 2) to approximately 81.7% on complex seawater data without synthetic data (Chapter 3), indicating that insufficient background diversity can reduce generalization in real deployment scenarios. The orthogonal experiments in Chapter 3 further support this interpretation, as the number of impurities has the strongest effect on model performance, while the number of salmon lice is less influential. The primary explanation is that more impurities introduce greater background diversity, which helps the detector generalize to complex seawater samples. A secondary, complementary explanation concerns how impurities interact with the pasting process itself. Because synthetic images are constructed by pasting salmon lice and impurities onto new backgrounds, the model could in principle rely on pasted boundaries as a shortcut cue rather than on genuine lice morphology to distinguish targets (Geirhos et al., 2020). Introducing a larger number of pasted impurity instances may weaken this shortcut: once impurities outnumber lice targets, pasted objects are no longer systematically associated with positive labels, which discourages reliance on paste artefacts and instead encourages the model to learn more stable target–background distinctions.

Given the practical constraint of limited real data, this thesis emphasizes expanding data diversity through synthesis rather than scaling model capacity. Offline synthesis increases dataset size, but its combinations remain fixed and may still be memorized, particularly by larger models. Online synthesis addresses this by dynamically generating new target, impurity, and background arrangements during training, making each batch more diverse and reducing overfitting to repeated patterns. This also likely explains its advantage under few-shot settings.

Within this online framework, proximal feature replay and fast gradient sign method (FGSM)-based adversarial regularization target two distinct aspects of the synthetic-real gap. Proximal feature replay introduces samples that are closer to real images in feature space, compensating for part of the distributional gap between diverse synthetic samples and real complex seawater samples. This feature-space compensation can help form a smoother decision boundary around real samples. FGSM-based adversarial regularization instead targets robustness to fragile, artefact-related cues. Because synthetic images may contain paste artefacts or other local irregularities, the detector may otherwise learn unstable patterns tied to these artefacts. FGSM introduces small loss-increasing perturbations, forcing the model to maintain predictions under harder inputs, which reduces dependence on such cues and encourages reliance on more stable lice morphology and target-background relationships (Goodfellow et al., 2014).

Salmon lice counts are already used as a regulatory indicator, but current practice relies on counting parasitic-stage lice from sampled fish (Thorvaldsen et al., 2019). The methods developed in this thesis could serve as a complementary early-stage indicator, supporting broader multi-stage monitoring. In farm management, early detection lets farmers act before infestations escalate, while in research it provides a basis for finer-grained ecological and epidemiological analysis of lice distribution and dispersal. The current work does not yet achieve fully automated monitoring, but combined with human-assisted seawater pre-processing and post-counting, it can still improve lice monitoring efficiency and accuracy. Follow-up work is currently applying the system developed in this thesis to characterize lice distribution across sea areas, depths, and times.

The value of this approach extends beyond salmon lice detection itself. This instance-based synthetic data approach is transferable to other aquaculture perception tasks sharing similar constraints: monitoring of plankton, other ectoparasites, larvae, or suspended biological particles, where weak target features, scarce labelled data, and abundant non-target interference are common.

7.2.2 Welfare modelling

The extraction of scientific statements for animal welfare modelling is a semantic evidence-screening task. Its main difficulty lies in semantic ambiguity and domain dependence. In many cases, welfare-related and non-welfare-related statements are not always clearly separated in the feature space and may share similar keywords, species names, experimental terms, or welfare-related expressions while differing in whether they provide usable evidence for building a welfare model.

Because this boundary is inherently ambiguous rather than clear-cut, recall is particularly important, since missing potentially relevant evidence may be more harmful to welfare model construction than introducing some false-positive candidates. A related consideration concerns the construction of negative samples. Since welfare knowledge usually appears as positive scientific statements, naturally labelled negative samples are limited. Sentences that appear superficially welfare-related but should not be included are useful, as they test whether the model has learned genuine semantic distinctions rather than shallow cues such as keywords, species names, or topical similarity.

Among the methods examined in this thesis, end-to-end fine-tuning of pretrained language models performed better than LLM-based few-shot methods and two-stage methods under the current experimental conditions. This suggests that direct supervised optimization can be more effective when the target boundary is narrow, domain-specific, and difficult to describe completely through prompts. Fine-tuning adjusts the model parameters using labelled examples and can better adapt to the specific inclusion criteria used in this thesis. By contrast, few-shot LLM performance can be affected by many factors, such as parameter scale, architecture, prompt design, example selection, and the model's implicit interpretation of the task. These results should not be interpreted as a general limitation of LLM-based methods. Rather, they show that, for the LLMs and prompts available at the time of this study, few-shot prompting was less stable than end-to-end fine-tuning for this specific semantic screening task. As LLMs continue to improve, this balance may shift. What is likely to remain unchanged is that welfare modelling can benefit from AI-based methods, whether through fine-tuned classifiers, LLMs, or future hybrid approaches.

Taken together, these findings position scientific statement extraction as a front-end component of welfare model construction. It demonstrates that the ambiguity underlying welfare-related language can be handled well enough to meaningfully reduce manual screening effort while keeping every candidate statement traceable for expert review.

7.2.3 Decision support system

The value of LLMs and AI-agent systems in DSSs lies in organizing structured and traceable decision workflows. Salmon farming decisions require the integration of heterogeneous information, such as farm states, regulatory constraints, site-specific conditions, and animal welfare. If recommendations rely only on the internal parametric knowledge of a model, they are vulnerable to outdated information, missing context, and hallucination (Huang et al., 2025).

RAG is therefore essential because it turns open-ended generation into evidence-grounded generation. Its effectiveness, however, depends on retrieval and context construction. In a salmon-farming DSS, hybrid retrieval is particularly useful because the knowledge base contains both precise domain terminology and semantically varied descriptions (Cormack et al., 2009; Mandikal & Mooney, 2024). BM25 can capture exact terms such as species names, treatment methods, legal expressions, and regulatory thresholds, while vector-based retrieval can identify relevant passages expressed with different wording. Combining the two improves coverage across regulations, guidelines, scientific evidence, and farm-specific procedures. After retrieval, local context expansion can improve context completeness by including neighbouring chunks, since definitions, conditions, or procedural steps may appear around the matched chunk (Zhao et al., 2024a). The window size for expansion must be controlled: too narrow a window may omit necessary context, whereas too broad a window may introduce redundant or conflicting information and increase the reasoning burden of the LLM (Fang et al., 2024). A further distinction concerns the scope of the knowledge itself: public legal and policy documents provide necessary constraints, but decision-making in practice also depends on more localized knowledge, such as site-specific data and expert judgement, and a practical DSS must distinguish between these rather than treating all retrieved evidence as equally authoritative.

Beyond retrieval quality, coordinating multiple agents introduces a further design problem. In an agentic DSS, the central design problem is the coordination of knowledge, tools, and reasoning steps. A multi-agent architecture can decompose complex decisions into subtasks such as regulatory compliance, risk identification, farm-state analysis, welfare-impact assessment, operational feasibility checking, and recommendation generation (Krishnan, 2025). This allows decisions that depend on many interacting factors, such as risk level, treatment feasibility, and regulatory thresholds, to be reasoned about in a structured way. However, more agents do not necessarily mean higher validity. If retrieval is inaccurate, knowledge is incomplete, or inter-agent communication and verification are weak, a multi-agent structure may amplify errors through inconsistency, context loss, or faulty intermediate outputs (Cemri et al., 2025). This coordination problem is addressed in S-MADSS by assigning each

agent a clearly bounded role, such as regulatory compliance or risk assessment, and by grounding every agent's output in explicitly retrieved evidence rather than free recall, so that intermediate outputs remain traceable back to their source documents.

Nonetheless, because each agent relies on an LLM to interpret retrieved evidence and generate intermediate reasoning, errors such as misread context or unsupported inference can still occur at any step, and a multi-agent pipeline offers no inherent guarantee against this (Ji et al., 2023). Reducing reliance on long LLM reasoning chains, for instance by supplying agents with structured, high-confidence inputs rather than requiring them to reason everything from retrieved text, is one way to narrow this risk. As LLMs continue to improve, their reasoning reliability is likely to increase as well, further reducing such errors over time.

7.3 Limitations and future work

Although this thesis explores possible pathways through which AI can support responsible and sustainable salmon farming management across three dimensions—perception, welfare assessment, and decision support—several limitations remain. These limitations arise not only from the individual methods developed in this thesis, but also from broader challenges encountered when AI methods move from experimental settings to real aquaculture environments.

7.3.1 Perception: salmon lice detection

One limitation concerns the domain gap between synthetic and real data. Although the synthetic-data method proposed in Chapter 3 uses target-free seawater backgrounds to generate more target-containing samples and to approximate a broader real-world distribution, a domain gap between synthetic and real data remains. Mixed training with real data narrowed this gap, but two limitations remain: synthetic samples cannot fully eliminate the boundary artefacts and texture mismatch introduced by the pasting process, and the real data used for post-training are mainly collected from one region in Norway. These data therefore cannot fully represent variations caused by different geographical regions, seasons, plankton communities, water-quality conditions, and sampling devices.

A related limitation concerns the difficulty of evaluation itself, especially the construction of ground truth for real samples. It is commonly assumed that judgments made by domain experts can serve as ground truth. However, this study found that this assumption holds only when the sample size is very small. A 20-minute video contains approximately $20 \times 60 \times 25$ frames, and each frame may contain hundreds of planktonic organisms or impurities. Experts need several hours to inspect such a

video, and their missed detections proved higher than those of the models. The final reference standard used in this thesis is therefore constructed from the union of all model detections and expert detections rather than true ground truth. Consequently, the detection performance reported on real samples, such as approximately 87% of the salmon lice included in the reference standard, should be understood as a result relative to an imperfect and operationally constructed reference standard, rather than as an exact estimate of the true detection rate.

The relative importance of evaluation metrics should be interpreted in relation to the current stage of the system. Because salmon lice occur at very low densities in seawater, while other planktonic organisms and suspended particles are far more abundant, recall is currently particularly important. The system has not yet achieved fully automated final counting, and false-positive removal and final counting still require subsequent human verification. Missing lice larvae may therefore be more problematic than retaining some false-positive candidates.

A further limitation is that the system has yet to achieve full automation and large-scale deployment. From the hardware perspective, the current seawater sample pre-processing procedure relies on a high-throughput water pump combined with two-stage filtration. The filtration process is mainly performed manually, which would become a bottleneck in large-scale applications. An automatic pre-filtration device will therefore be needed. From the model perspective, Chapters 2 and 3 detect salmon lice using bounding boxes, but the collected data are continuous-frame videos. Frame-by-frame results need to be aggregated to avoid repeatedly counting the same individual. At present, the model saves and annotates frame-level candidates, after which cropped targets are manually counted. This has reduced the processing time of a 20-minute sample to a few minutes, but the workflow is still not fully automated. One feasible direction is to introduce a tracking method and assign a unique identifier (ID) to each target across frames. However, under real conditions, salmon lice move with the water flow, rotate, become occluded, and change appearance. Strong dynamics and background variation may make both detection and tracking unstable, causing ID drift or target loss, and consequently leading to repeated counting or missed counting.

To address these limitations, future work can improve the system in several directions. As data continues to accumulate, the representativeness of detection models can be improved by combining synthetic data with self-supervised or foundation-model pre-training, active learning, and failure-sample feedback mechanisms. For instance, large volumes of unlabelled seawater images are collected during routine sampling, most of which currently go unused beyond the manually annotated subset. These images could serve as pretraining data for contrastive methods, such as MoCo (He et al., 2020)

or I-JEPA (Assran et al., 2023), allowing the detection backbone to learn general representations of seawater scenes before fine-tuning on the comparatively small labelled dataset used in this thesis. Alternatively, foundation models, such as DINO (Caron et al., 2021), already shown to transfer well across natural and remote-sensing imagery, could be adapted to seawater imagery with only a lightweight detection head trained on top, further reducing reliance on labelled lice examples. Either route would complement the synthetic-data strategies developed in this thesis. A secondary classification model could also be added after detection to further screen candidate targets and reduce false positives, a direction additional preliminary tests have indicated to be feasible. Video temporal information could also be used so that the model considers motion, morphology, and trajectory continuity across neighbouring frames, rather than relying only on single-frame appearance. In addition, salmon lice have been reported to exhibit colours under fluorescence that differ from those of other planktonic organisms (Thompson et al., 2022). Future work could therefore explore imaging systems that combine fluorescence and visible-light imaging to enhance discriminative features. Finally, long-term testing in more regions, seasons, and farming environments is still required to evaluate the generalization ability and operational stability of the system under real deployment conditions.

Despite these limitations, this thesis provides an important foundation for automated salmon lice detection, demonstrating the feasibility of using machine learning to identify rare biological targets in complex seawater backgrounds and offering a methodological basis for future high-frequency, low-cost, and regional monitoring systems.

7.3.2 Welfare: evidence extraction for animal welfare assessment

The limitations of this work stem from a single underlying issue: real scientific literature is far more diverse than any dataset built from a limited set of species and semantic-modelling sources can capture. Although Chapter 5 constructs the dataset using data from dairy cows, sows, and salmon, and evaluates generalization through cross-domain testing, the distribution of real literature, spanning far more species, welfare topics, and writing styles, is considerably broader. This same gap also explains why semantic boundaries remain difficult to resolve: welfare-related and non-welfare-related statements do not fall into cleanly separated regions of feature space, but occupy a continuous, blurred spectrum that even domain experts must judge case by case carefully. This difficulty is reflected directly in the numbers: on the expert evaluation dataset, the model achieved a precision of only 59.1%, alongside a recall of 76.5%. This trade-off reflects the recall priority discussed above, but it also carries a practical cost: a substantial proportion of candidate statements will still require ex-

pert filtering before use. Even so, a lower-precision screening tool can still eliminate a large share of irrelevant statements, meaningfully reducing the workload of manual literature review.

Another limitation concerns the scope. Scientific statement extraction is only one component of complete animal welfare modelling. A complete model still requires further extraction of welfare attributes, environmental indicators, animal-based indicators, management factors, and their relationship strengths and applicable conditions from the identified statements. Therefore, these outputs still need to be interpreted, filtered, and integrated with domain knowledge. Future research should extend this work to the extraction of attributes, indicators, weighing categories, and relationships, as well as evidence grading and conflict detection, so that the system can not only identify relevant statements but also determine how these pieces of evidence support the structure and weighting of welfare models.

The choice of method may change as technology develops. In Chapter 5, the end-to-end fine-tuned model outperformed the few-shot LLM and two-stage methods, indicating that targeted fine-tuning is more effective under the data scale and task setting of this study. However, this does not mean that LLMs will not become advantageous in the future. As LLMs continue to improve in long-text reasoning and tool use, welfare evidence modelling may evolve from single-step classification into multi-step knowledge construction. For example, multiple LLM-based agents could be assigned to literature parsing, statement extraction, welfare attributes, and weighing categories extraction before final verification by experts.

Despite these limitations, this study provides a methodological foundation for animal welfare assessment in aquaculture. Aquatic animal welfare involves multiple dimensions, including health, behaviour, physiological stress, environment, and management. Its assessment depends on large amounts of scientific evidence scattered across different species and farming systems. By improving the efficiency, transparency, and traceability of welfare evidence acquisition, it supports the future development of updatable and evidence-driven integrated welfare assessment models.

7.3.3 Decision support in aquaculture

The current S-MADSS has not yet fully integrated the perception model and animal welfare evidence extraction method developed in the preceding chapters. Although the system architecture supports the incorporation of such modules through external functions, knowledge bases, and submodules, this thesis has not implemented a complete end-to-end workflow from salmon lice detection and welfare assessment to decision recommendations. Achieving full integration would require deploying sam-

pling and detection equipment in real farming environments, continuously collecting field data, and converting detection results into inputs that can be used by the DSS. However, such deployment conditions and continuous field data are difficult to obtain in the short term.

Second, the practical decision-making capability of S-MADSS is constrained by the scope and quality of available knowledge and data resources. Although RAG with external knowledge bases reduces reliance on the model's internal parametric knowledge and improves traceability, they also make system performance dependent on the coverage, timeliness, and scenario-specificity of the knowledge provided (Wang et al., 2024e). Public sources usually provide only general principles and rarely include operational details essential for real decision-making, such as company-specific operating manuals, treatment procedures, historical implementation logs, and abnormal cases (Zhou et al., 2025). As a result, the current system remains limited in generating highly actionable and context-specific recommendations. Therefore, future S-MADSS design should incorporate more operational data and historical cases under appropriate data governance and privacy-protection mechanisms, while further clarifying human-in-the-loop mechanisms, including recommendation review procedures, responsibility boundaries, error-correction mechanisms, and stakeholder roles.

Third, the current system still requires more testing in real-world scenarios. This thesis mainly uses a case study to demonstrate the feasibility of S-MADSS for salmon lice treatment recommendations. However, real management involves more complex situations. Future work should therefore evaluate the system across a broader range of decision scenarios, including the accuracy, interpretability, stability, user trust, and compatibility of its recommendations with actual management workflows. Before such systematic evaluation is completed, the conclusions of this thesis should be limited to demonstrating the feasibility of the framework, rather than making general claims about its decision quality or risk-control effectiveness.

Despite these limitations, S-MADSS demonstrates how multi-agent structures, external knowledge bases, and function calling can be combined into a unified evidence-based decision-making workflow, a design concept for aquaculture DSSs. Future research may focus on building more domain-specific agent systems with broader integration of sensors, data-analysis modules, and predictive models. For example, sensor data could be combined with time-series models to predict environmental or health-related changes (Cerqueira et al., 2024), while vision-language models could describe fish health status from visual information (Zhang et al., 2024b). These outputs could then support decision-making and early-warning functions, enabling agent-based DSSs to integrate existing research achievements into more practical and adaptive management systems.

7.4 Societal impact

This thesis explores AI-supported salmon farming through farm-state perception, animal welfare assessment, and decision support. These contributions do not by themselves solve the sustainability challenges of salmon farming, but they can reduce management risks caused by insufficient information in seeing, knowing, and acting. This section discusses the societal relevance of this research at two levels. First, at the level of the aquaculture production system itself, contributions are assessed through the three pillars of sustainability: economic, environmental, and social (Asche et al., 2018). Second, at the level of AI as a technology, we consider the broader ethical, legal, and societal (ELSA) implications introduced by its use (Wang et al., 2025a).

7.4.1 Contributions to sustainable salmon farming

The global demand for nutritious food rises and aquatic foods are becoming increasingly important for food security (Troell et al., 2023; FAO, 2025). Within the three pillars of sustainability introduced above, this thesis contributes to food production and industry stability, animal welfare and ethical responsibility, and environmental impact.

7.4.1.1 Economic sustainability: production and industry stability

From the perspective of economic sustainability, digitalized, information-rich, and intelligent aquaculture can support more efficient resource use, more stable production quality, and reduced production risks, ultimately leading to higher production volumes (Føre et al., 2018). By combining perception, welfare evidence extraction, and decision support, this thesis contributes to the development of precision aquaculture and moves one step closer to the broader vision of blue transformation (FAO, 2022). Compared with farm management based primarily on experience, more precise perception and more evidence-based decision support can facilitate more context-specific decisions on lice control, feeding, and disease management. Timely detection of planktonic salmon lice may facilitate early management interventions, reducing farm-level economic losses (Rabe et al., 2024).

7.4.1.2 Social sustainability: animal welfare

From the perspective of social sustainability, animal welfare is an important ethical dimension of sustainable development and has received increasing public attention (Keeling et al., 2019). The work in Chapter 5 contributes to more systematic use of welfare research. Its significance does not lie in producing more food, but in respecting animal ethics and animal life, and in reducing suffering due to farm-

ing (Bracke et al., 2023). Decision support based on animal welfare and regulatory compliance can further reduce risks of health damage or mortality caused by inappropriate environmental conditions or management actions.

7.4.1.3 Environmental sustainability

From the perspective of environmental sustainability, aquaculture generally has lower emissions than terrestrial livestock production (Gephart et al., 2021), but it can still affect surrounding wild environments through chemical dispersion, parasite transmission, and other ecological interactions (Taranger et al., 2015). The salmon lice detection work in this thesis helps address a monitoring gap for early stages of salmon lice. Earlier and more reliable information about lice levels in seawater can support more targeted interventions, potentially reducing the frequency of large-scale chemical or mechanical delousing operations. This matters because frequent chemical treatments accelerate the development of lice resistance, reducing long-term treatment effectiveness, while also affecting non-target crustaceans in the same environment (Urbina et al., 2019). Beyond farm boundaries, the same monitoring approach could be applied in coastal habitats and wild salmon migration routes, providing spatially resolved information on exposure pressure to support environmental risk assessment. Such information could ultimately inform management decisions that reduce salmon lice impacts on wild salmonid populations, whose decline or collapse can have significant consequences for the broader ecosystem (Samways, 2025).

At the same time, it is necessary to acknowledge that AI methods also have environmental costs (Xiao et al., 2025). In this thesis, the inference energy consumption of edge-deployed salmon lice detection models is limited. The more substantial computational and carbon costs are mainly associated with the training and use of LLMs and multi-agent decision-support systems. A more balanced assessment is therefore required. AI may reduce treatment frequency and associated environmental impacts through more precise management, but it also introduces emissions through computation. This trade-off should be explicitly considered rather than assuming that the use of AI automatically leads to "greener" solutions.

7.4.2 Implications of introducing AI into aquaculture: Ethical, Legal, and Societal Aspects (ELSA)

Beyond technical performance, the ELSA of AI must also be considered. This is particularly relevant because the outputs of this research are not merely algorithmic demonstrations, but tools that may influence future aquaculture management, welfare assessment, and regulatory processes.

7.4.2.1 Ethical aspects: reliability and human oversight

In terms of reliability, detection models may produce false negatives and false positives, welfare assessment models may be affected by biases in training data, and LLM-based decision-support systems may generate recommendations that are difficult to verify. This thesis uses synthetic data to mitigate insufficient and imbalanced training data, and external knowledge bases to reduce hallucination in LLM-based systems. However, these methods can only reduce uncertainty rather than eliminate it. Therefore, such systems should clearly present their uncertainty, scope of applicability, and evidence sources, so that outputs can be understood, questioned, and reviewed. This requirement becomes particularly pressing if monitoring of salmon lice density or welfare assessment is incorporated into future regulatory requirements: where automatic counts may influence treatment obligations or regulatory consequences, they should not be treated as unquestionable facts, and space should be retained for human review.

For this reason, this thesis positions these systems as assistive tools and emphasizes human-in-the-loop oversight. However, human involvement alone does not guarantee effective judgment. When recommendations appear authoritative, users may formally retain the right to review while substantively deferring to the system, a problem often referred to as automation bias (Alon-Barkat & Busuioc, 2023). The effectiveness of human-in-the-loop design, therefore, depends on whether the system is designed to be questioned and corrected. This includes presenting confidence levels and evidence sources, allowing recommendations to be traced and overturned, and clarifying responsibility in advance when errors occur. Especially for sensitive operations, such as medication use, mechanical treatment, and stocking density adjustment, final decisions should remain the responsibility of qualified professionals. Looking forward, this requirement may also need to be formalized at the regulatory level: until AI systems can reliably achieve full autonomous decision-making and their developers are prepared to assume corresponding liability, regulation should continue to require a qualified human to retain final decision-making authority, as a safeguard for responsible farm management.

7.4.2.2 Legal aspects: data, compliance and responsibility

From a legal perspective, AI-supported aquaculture may involve issues of data ownership, protection of commercially sensitive information, copyright, regulatory compliance, and responsibility allocation. Farm data may include sensitive information, such as production volume, mortality, disease status, treatment strategies, and equipment conditions. The use of LLMs may also raise questions about training-data sources and copyright (Chesterman, 2025). As relevant legal frameworks are still developing,

future system design and deployment should, as far as possible, rely on data and knowledge bases with clear provenance, explicit authorization, and traceability, in order to reduce potential compliance risks.

7.4.2.3 Societal aspects: labour structure, environmental cost, and equity

At the social level, AI may change both the knowledge and labour structure of aquaculture. Traditional farming has relied heavily on experiential judgement, whereas AI systems may transform part of this knowledge into information that can be recorded, analysed, and shared. Farmers' work may also shift from direct observation and manual operation toward system monitoring, anomaly interpretation, and technical maintenance, a transition closely related to the human-in-the-loop role discussed above. This shift is unlikely to eliminate existing positions, as these systems are designed to support rather than replace human judgement. The deployment, maintenance, and servicing of detection equipment and decision-support systems may instead create new technical roles in the aquaculture sector. At the same time, it may also create pressure for skill development. Effective use of these systems requires farmers to understand system outputs, identify errors, and intervene manually when necessary. The implementation of AI technologies should therefore be accompanied by training and capacity building to support this transition.

In addition, AI technologies may raise issues of social justice and fairness within the aquaculture industry. Large companies are usually better positioned to deploy sensing equipment, computing platforms, and AI systems, whereas small and medium-sized farmers may face constraints in cost, technical capacity, and data infrastructure. If digital transformation mainly benefits actors with stronger technological and financial capacity, it may further widen inequalities within the industry. AI development for aquaculture should consider not only model accuracy, but also accessibility, affordability, and maintainability. Beyond fairness in resource access, adoption may also be constrained by trust: farmers may be reluctant to rely on AI-based decisions that lack transparency, particularly when the reasoning behind a recommendation is not clear to the end user. Given these technical, economic, and trust-related barriers, the broader adoption of AI in aquaculture is likely to be a gradual process rather than an immediate shift.

7.5 Final conclusion

Salmon farming, like most intensive animal production systems, faces a wide range of interconnected challenges that no single thesis can resolve. This work contributes

one step forward within that broader landscape: it demonstrates how complementary AI methods can be positioned across the seeing–knowing–acting sequence of perception, welfare assessment, and decision-support layers. Realizing this contribution in practice will depend on the continued availability of representative data, traceable knowledge sources, and human oversight, as discussed above.

The underlying logic, however, is not specific to salmon or even to aquaculture. Many animal production systems face a structurally similar sequence of bottlenecks: a difficult-to-observe biological state, welfare knowledge that is scattered across a growing body of literature, and management decisions that must reconcile operational, welfare, and regulatory constraints. Poultry, swine, and dairy farming share this structure. For example, for dairy cows, where early-stage lameness is similarly difficult to detect (Laschinger et al., 2024; Fabian et al., 2014), welfare evidence is likewise dispersed across the literature (Van Nuffel et al., 2015; Whay & Shearer, 2017), and treatment decisions must also balance welfare, cost, and regulatory constraints (Sadiq et al., 2019). The perception–welfare assessment–decision support pipeline developed in this thesis could, in principle, be re-instantiated in such settings beyond aquaculture.

Future work should test this transferability directly, both by extending the salmon farming case toward fuller integration of its three layers, and by adapting the same pipeline to other production systems where perception, welfare evidence, and decision support face similar challenges.

Appendices

Appendix A

Additional methodological details and results related to Chapter 2

A.1 Description of detection methods and training process

A.1.1 Salmon lice detection methods

In this study, 16 object detection models were selected and tested for their ability to detect salmon lice. The models include Convolutional Neural Network (CNN)- and Transformer-based models. CNNs have been widely used in the field of object detection for a long time and various CNN-based models have evolved. CNNs have powerful spatial feature extraction capabilities that rely on convolution kernels to continuously extract features. CNN-based models can be categorized into two groups: two-stage detection models and one-stage detection models. The two-stage CNN-based models first generate region proposals of salmon lice and then predict the objects and bounding boxes for each region proposal (Diwan et al., 2023). This kind of model can achieve higher accuracy than one-stage CNN-based models but is slower due to multi-stage calculation. Faster R-CNN (Ren et al., 2015), Sparse R-CNN (Sun et al., 2021), and Cascade R-CNN (Cai & Vasconcelos, 2018) were chosen as two-stage CNN-based models in this study. In contrast, one-stage CNN-based models can generate output of both the bounding boxes of salmon lice and their classification, by feeding input to the network only once. The one-stage models are fast but with a relatively low accuracy. In this work, Single Shot MultiBox Detector (SSD) (Liu et al., 2016), RetinaNet (Lin et al., 2017b), and YOLO series were chosen as one-stage CNN-based models. The Transformer-based models rely on the attention mechanism, comprising both self-attention and cross-attention, to detect global dependencies between input and output (Meng et al., 2021). The Transformer-based models started out as an application for machine translation tasks (Vaswani et al., 2017). Later, they were gradually applied to various fields of natural language processing as well as computer vision. Transformer-based object detection models require positional encoding to understand the spatial arrangement of objects within the image, as they do not have inherent spatial understanding like CNN-based models do. Therefore, Transformer-based models are excellent at handling global context and long-range dependencies, making them suitable for capturing relationships between objects in a whole image. In this study, Deformable DETR (Zhu et al., 2021) and RT-DETR (Zhao et al., 2024b) were incorporated as Transformer-based models.

A.1.1.1 Two-stage Convolutional Neural Network- (CNN-) based models

Faster R-CNN (Ren et al., 2015) is a classical two-stage CNN-based network architecture that combines the strengths of two key components: a Region Proposal Network

(RPN) and a Fast R-CNN (Girshick, 2015) detector. The RPN utilizes the feature map from the feature extraction network as input and generates region proposals with a range of scales and aspect ratios. The Fast R-CNN then classifies and refines these proposals to precisely identify and locate objects within an image. RPN shares convolutional layers with Fast R-CNN. It is mapped to a feature vector by performing a sliding window process on the feature map matrix output from the last shared convolutional layer. This vector passes through two fully connected layers: a border regression layer and an object classification layer. The regression layer and the classification layer are two independent and parallel fully connected layers; the regression layer mainly operates on the estimation of the location of the original object, and the classification layer mainly classifies the object and computes the probability that the object belongs to a certain class (e.g., nauplius live, shell or copepodite).

Cascade R-CNN is an improved two-stage CNN-based network based on Faster R-CNN (Cai & Vasconcelos, 2018). It addressed a significant issue in R-CNN networks where the selection of Intersection over Union (IoU) thresholds in the R-CNN has a substantial impact on the quality of the finally detected bounding boxes. The authors proposed a cascade R-CNN structure to tackle this problem. The cascaded R-CNN finds a set of false positive detections that are close to the ground truth (i.e., the larvae identified by the marine biologist) for the next training stage by adjusting bounding boxes at each stage. Different IoU thresholds are used in different cascading stages to recompute positive and negative samples, along with various sampling strategies. This gradual refinement process progressively improves the quality of the bounding boxes, ultimately resulting in high-quality detection results.

Sparse R-CNN is an object detection architecture introduced by Peize et al (Sun et al., 2021). Unlike the R-CNN family, such as the fast-R-CNN, Sparse R-CNN establishes a thorough sparse framework to replace hundreds of thousands of candidates from RPN with a small set of proposal boxes. Sparse R-CNN is composed of a backbone network, a dynamic instance interactive head, and two prediction layers. Feature Pyramid Network (FPN) (Lin et al., 2017a) and ResNet architecture are used as the backbone of Sparse R-CNN. The FPN structure performs top-down up sampling so that the underlying feature map contains more semantic information contained in the image. A fixed small number of proposal boxes are set as regional proposals. The proposal boxes are represented as learned parameters and updated with back-propagation. At the same time, learnable proposal features were introduced in Sparse R-CNN, which are also a set of learnable parameters corresponding to the proposal boxes. The head of the model is a set of iterative dynamic instance interactive head compositions. The newly generated object boxes and object features are used as the proposal features and proposal boxes of the next head. The self-attention module is used before dynamic

instance interaction. The two prediction layers, i.e., the regression layer and the classification layer, were used to predict target location and species, respectively. The network architecture is free from the concepts of anchor box, reference point, and RPN, and does not require Non-Maximum Suppression (NMS), a complex post-processing step that screens candidate detections after inference.

A.1.1.2 One-stage CNN-based models

Single Shot MultiBox Detector (SSD) is a one-stage CNN-based object detection algorithm (Liu et al., 2016). The SSD model comprises three key components: a base network, convolutional feature layers, and convolutional predictors. The base network is built upon a VGG-16 architecture which is a convolutional neural network with 16 layers deep (Simonyan & Zisserman, 2014). It serves as the foundational framework for feature extraction from the input image. The convolutional feature layers play a pivotal role in enhancing the model's ability to identify the objects of interest (e.g., nauplius larvae) in the image. They further process the features obtained from the VGG-based base network, expanding the model's analysis of the image. This process leads to the creation of six feature maps. These feature maps serve as the foundation for the final step in the SSD algorithm. Convolutional predictors use these feature maps to make predictions about the coordinates, confidence scores, and class information for objects of various sizes. In essence, they provide detailed information about the location and classification of objects within the input image.

RetinaNet is a one-stage CNN-based object detection network introduced in 2017 (Lin et al., 2017b). RetinaNet uses ResNet and FPN as the backbone of the network architecture. The input image is processed by Resnet and FPN to generate a feature pyramid with five feature maps. Each output feature map has 256 channels and different step sizes. This design is optimized for detecting objects at different scales. The larger step sizes produce smaller feature maps, which excel at detecting larger objects, while the smaller step sizes generate larger feature maps, which excel at detecting smaller objects. After obtaining the feature pyramid, the head module processes each layer of the feature pyramid using two separate sub-networks, the classification network and the position regression network. While the classification and regression networks do not share parameters, the five output feature maps share the weights of the head module. This ensures consistency and efficiency in processing multiple feature scales. In the classification network, RetinaNet adopts a pioneering loss function, the "Focal loss," to address the problem of extreme imbalance between foreground and background species (e.g., nauplius and plankton respectively), which is highly likely to be encountered in one-stage CNN-based object detection architectures. The Focal loss adds a moderator to the original cross-entropy loss function to make

the loss function more focused on difficult samples. This loss function effectively solves the class imbalance problem inherent in the object detection task and further improves the performance of RetinaNet.

YOLOv3 is a one-stage CNN-based anchor-based object detection model released in 2018 (Redmon & Farhadi, 2018). The backbone proposed in YOLOv3 is Darknet-53. It replaces all maximal pooling layers with stepwise convolutions and adds residual connectivity compared to its predecessor Darknet-19 in YOLO (Redmon et al., 2016), the first real-time end-to-end approach for object detection. It contains a total of 53 convolutional layers. A fundamental feature of YOLOv3 is multiscale prediction. Yolov3 predicts three types of bounding boxes (e.g., "Nauplius Live," "Nauplius Shell," and "Copepodite") at three different scales. This approach enhances the precision of bounding box predictions, particularly for smaller objects.

YOLOv5 is a model released by Ultralytics in 2020 (Ultralytics, 2020). The network architecture of YOLOv5 is mainly composed of a backbone, neck, and head. YOLOv5 uses a modified CSPDarknet53 as the backbone. CSPDarknet53 is a deep neural network that consists of a series of convolutional layers, residual blocks, and up-sampling layers for extracting image features. In the Neck layer, YOLOv5 uses a combination of FPN and a path aggregation network (PAN) to fuse multi-scale features of the image. The PAN structure performs bottom-up downsampling so that the top-level features contain image location information. The two features are finally fused so that feature maps of different sizes contain semantic information and feature information of the image, ensuring accurate prediction of images of different sizes. YOLOv5 is an anchor-based network architecture. In the prediction stage, the model predicts the offset and deformation of the anchor on the grid to determine the position and size of the predicted object.

YOLOX (Ge et al., 2021) is a one-stage CNN-based object detector based on YOLOv3, released in 2021. YOLOX removes the anchor mechanism, making it an anchor-free architecture. This means that it can directly predict the center and the size of the object instead of the known offset and change of the anchor. This reduces the number of predicted boxes, thereby accelerating the NMS process. To compensate for the large imbalance due to the lack of anchors, the authors used center sampling, designating the central 3×3 area of the feature map as the positive region. To avoid ambiguity in the ground truth label due to the overlap of multiple objects, YOLOX adopts SimOTA, which formulates the label assignment as an optimal transmission problem through a top-k strategy. YOLOX uses a decoupled head for making predictions. This solves the conflict between classification and regression tasks. The decoupled head contains a 1×1 convolutional layer to reduce the channel dimension, followed by two parallel branches with two 3×3 convolutional layers for classification and regression

tasks respectively. In addition to the improvements in network architecture, YOLOX also uses several image enhancement strategies such as Mosaic and Mixup.

YOLOv7 was released in 2022 (Wang et al., 2023). At that time, it outperformed all known object detectors in terms of speed and accuracy in the range of 5 FPS to 160 FPS. The architecture of YOLOv7 is also inspired by the Re-parameterized Convolution (RepConv) (Ding et al., 2021). However, the authors found that the identity connections in RepConv would destroy the residuals in ResNet (He et al., 2016) and the concatenation in DenseNet (Huang et al., 2017). Therefore, the identity connection was removed and a new architecture RepConvN was proposed and adopted in YOLOv7. A special feature of YOLOv7 is that its proposed E-ELAN is suitable for models with unlimited stacked computational blocks. In YOLOv7, the authors considered the amount of memory required to keep the layers in memory and the distance required for the gradient to propagate backward through the layers. The shorter the gradient, the better the network's ability to learn. Therefore, they proposed E-ELAN based on Efficient Layer Aggregation Network (ELAN) (Wang et al., 2022a). E-ELAN combines different groups of features by shuffling and merging them to enhance the network's learning ability without destroying the original gradient paths.

YOLOv8 was released by Ultralytics, the same author as YOLOv5, in 2023. YOLOv8 is based on YOLOv5. It adopts a more advanced and efficient architecture. The backbone of YOLOv8 also draws inspiration from the CSPDarkNet-53 network. However, YOLOv8 replaces the C3 module with the C2f (CSPLayer_2Conv) to make the inference faster. As for the head of YOLOv8, it adopts a Decoupled Head similar to YOLOX, which separates the regression branch from the prediction branch. YOLOv8 uses CIoU (Zheng et al., 2021) and distribution focal loss (DFL) functions to compute bounding box loss and use binary cross-entropy to compute classification loss. These loss functions improve the performance of predicting objects, especially when dealing with small objects.

A.1.1.3 Transformer-based models

Detection Transformer (DETR) was released by Facebook in 2020 (Carion et al., 2020). It is a pioneering object detection architecture that combines CNNs for image feature extraction with Transformers for direct object bounding box predictions. The architecture of DETR comprises a multi-layer Transformer encoder for image features and a Transformer decoder with object queries. Multiple prediction heads, including a Feed-Forward Network, generate object classes and bounding box predictions. This approach streamlines object detection and showcases the versatility of the Transformer architecture in computer vision.

Deformable DETR is an object detection architecture based on DETR proposed in 2021 (Zhu et al., 2021). Reported limitations related to DETR include: (1) It requires longer training epochs. (2) Small objects are detected from high-solution feature maps, which leads to low performance when detecting small objects. Inspired by deformable convolution, Deformable DETR solves these problems by introducing a deformable attention module. This module focuses on a small number of sampling locations. It acts as a pre-filter to filter out the prominent key elements from all feature pixels. The deformable attention module only focuses on a small set of key sampling locations around the reference point, regardless of the spatial size of the feature map. This alleviates the problems of slow model convergence and computational complexity associated with high-resolution feature maps. The module can be naturally extended to handle multiple scales. Deformable DETR combines the sparse spatial sampling of deformable convolutions with the relational modeling capabilities of deformers. It effectively alleviates the problems of slow convergence and high complexity of DETR.

RT-DETR is an object detection network proposed in 2023 (Zhao et al., 2024b). It consists of a backbone network, a hybrid encoder, and a Transformer decoder with auxiliary prediction heads. Similar to DETR, the backbone of RT-DETR continues to employ the CNN architecture for image feature extraction. The encoder of RT-DETR consists of two modules, the Attention-based Intra-scale Feature Interaction (AIFI) module and the CNN-based Cross-scale Feature-fusion Module (CCFM). In the encoder, RT-DETR reduces the low-level features due to the lack of semantic concepts. AIFI performs in-scale interactions only for the deepest feature map. This significantly reduces the latency without reducing model performance.

A.1.2 Training process

In this study, all experiments were conducted using the PyTorch framework (Paszke et al., 2019). The computer configuration utilized for these experiments is detailed as follows: an Intel i9-13950HX CPU, 32GB of RAM, an NVIDIA RTX A5500 GPU, and the Pop!_OS 22.04 LTS operating system.

The parameters of each model employed in this study are shown in Table A.1. An optimizer is a crucial component in deep learning that guides the adjustment of various parameters of the loss function during the process of backpropagation. Its primary purpose is to update these parameters in the correct direction with appropriate magnitudes, aiming to iteratively bring the value of the loss function closer to the global minimum. The learning rate (LR) indicates the step size of the gradient moving towards the optimal solution of the loss function in each iteration. Selecting a high learning rate may lead to weights exceeding optimal values, while a low learning rate

Table A.1: Configurations of detection models.

	Model	Backbone	Optimizer	LR	LR momentum	WD factor	Epochs	Input size
Two-stage CNN-based	Faster-R-CNN 50	ResNet-50-FPN	SGD	0.02	0.9	0.0001	24	(768,1344)
	Faster-R-CNN 101	ResNet-101-FPN	SGD	0.02	0.9	0.0001	24	(768,1344)
	Cascade R-CNN	ResNet-101-FPN	SGD	0.02	0.9	0.0001	20	(768,1344)
	Sparse R-CNN50	ResNet-50-FPN	AdamW	0.000025	-	0.0001	36	(768,1344)
	Sparse R-CNN101	ResNet-101-FPN	AdamW	0.000025	-	0.0001	36	(768,1344)
One-stage CNN-based	SSD300	VGG16	SGD	1e-3	0.9	0.0005	24	(300,300)
	SSD512	VGG16	SGD	1e-3	0.9	0.0005	24	(512,512)
	RetinaNet 50	ResNet-50-FPN	SGD	0.01	0.9	0.0001	24	(768,1344)
	RetinaNet 101	ResNet-101-FPN	SGD	0.01	0.9	0.0001	24	(768,1344)
	YOLOv3	DarkNet-53	SGD	0.001	0.9	0.0005	273	(192,320)
	YOLOv5su	CSPDarknet53	AdamW	0.001667	0.9	0.0005	100	(640,640)
	YOLOX	DarkNet-53	SGD	0.01	0.9	0.0005	300	(640,640)
	YOLOv7	RepConvN	Adam	0.01	0.937	0.0005	300	(640,640)
	YOLOv8n	CSPDarknet53	AdamW	0.001667	0.9	0.0005	100	(640,640)
Transformer-based	Deformable DETR	ResNet-50	AdamW	0.0001	-	0.0001	50	(750,1333)
	RT-DETR	HGNetv2	AdamW	0.001667	0.9	0.0005	100	(640,640)

Note: LR denotes learning rate. WD denotes weighing decay.

can result in slow convergence or getting trapped in local minima. LR Momentum is a value that determines how much the previous weight update influences the current update. It can help the network overcome local minima and speed up convergence. Weight decay (WD) serves as a regularization technique to mitigate overfitting risks in neural networks by reducing the impact of less important parameters on the loss function. Notably, the weight decay factor does not adversely impact meaningful weights. The number of epochs is a critical hyperparameter that dictates how many times the learning algorithm will iterate over the entire training dataset. During training, the weights of the network are updated based on data, which can pose challenges in achieving good performance on a small dataset. Transfer learning mitigates this issue by allowing models pre-trained on larger datasets to fine-tune their weights on smaller datasets, thereby enhancing the performance of models trained on the small dataset. In this study, all models were pre-trained on the COCO dataset. Then, training was performed on the salmon lice dataset using transfer learning.

A.2 Obtaining ground truth of salmon lice in seawater

In biomass monitoring, achieving 100% confirmation of salmon lice presence in liquid containing hundreds of thousands of plankton is very challenging. However, we can approach this value closely and minimize human-induced biases to enhance result reliability. In this study, salmon lice as identified by an expert were used as ground truth. Ground-truthing real seawater samples with unknown salmon louse contents is

challenging and will always depend on the performance of a human expert. When setting the ground truth to evaluate the performance of models, it is therefore advisable to clarify the expected error rates of the human expert.

For this work, the expert (Expert A) to set the ground truth for model comparisons has been formally trained in plankton analyses and has identified and counted different planktonic species in seawater samples for well over 1000 hours. Additionally, he became an expert in salmon louse larvae identification after hundreds of hours of identifying, counting, and annotating salmon lice in simple and complex backgrounds. In samples without any other plankton, and in samples with few other plankton, Expert A is likely to find all salmon louse larvae. For more complex samples, however, the performance of the expert should be evaluated against a real ground truth. Laboratory-hatched larvae may differ from larvae in un-spiked plankton samples. A proper and detailed evaluation should include blind counting of samples with known amounts of salmon lice (spiked) and samples with unknown amounts of salmon lice should be analyzed by several experts to find a ground truth as good as possible, to then rate the results of individual experts against this ground truth. Samples with different plankton densities and compositions should be used. Such a detailed and in-depth evaluation was outside of the scope of the present article, as we intend to underscore the applicability of our method more than a proper performance evaluation on real, un-spiked seawater samples. We still evaluated the performance of Expert A, employing the following procedure.

- First, a real plankton sample (about 50 ml sample, about 30 organisms per 80 mm^3) was spiked with 20 salmon louse nauplii. Expert A counted this sample and found all 20 nauplius larvae.

- Second, a large plankton sample has been seeded with salmon lice larvae and a video has been recorded of part of this sample. Two experts, Expert A and Expert B counted salmon lice in the sample blind (that is without previous knowledge of the amount of salmon louse larvae in the sample) and noted the times in the video at which each salmon louse larva first appeared. We assumed that both experts together would find close to all salmon lice in the sample. Therefore, the experts double-checked all instances that were noted down together to define the number of salmon louse larvae in the sample. This was done by scrutinizing frames containing alleged salmon lice together. This resulted in 69 individual salmon louse larvae in the sample, of which Expert B found 66, and Expert A found 68 larvae. Additionally, Expert B wrongly identified 3 organisms as salmon lice, and Expert A one.

We therefore assume in the frame of this article that the false positive and false

negative rates of Expert A for the two videos used to test the ability of our models to identify salmon louse larvae in real, un-spiked marine plankton samples was within few percent. Consequently, the margin of error in the ground truth employed herein likely varies by no more than one salmon louse larva.

A.3 Additional results

A.3.1 Results of salmon lice detection in Dataset NLS

A.3.1.1 Overall salmon louse detection results

All 16 models were trained and tested on the Dataset NLS, which only contained Nauplius Live and Nauplius Shell, but no Copepodite yet (unlike our final Dataset NC). The overall results on the performance of detection and inference speed for Dataset NLS are shown in Table A.2.

The two-stage CNN-based models showed consistent performance on a variety of metrics. Notably, all models had mAP_{50} above 0.96. Furthermore, for the more stringent $mAP_{50:95}$ metric, all models except Faster-R-CNN 101 generated values above 0.6. Cascade R-CNN stood out with a mAP_{50} score of 0.973 and also performed well in $mAP_{50:95}$. In terms of precision and recall, Sparse R-CNN50 achieved the highest values, which are 0.973 and 0.945, respectively, resulting in the best F1 score. On the other hand, Faster-R-CNN 101 achieved the lowest precision and recall, with a recall of 0.928, indicating that a relatively larger number of objects were not detected. It is also worth noting that the two-stage CNN-based models generally exhibited relatively low running speeds with lower than 25 FPS, which poses a challenge for real-time detection.

The one-stage CNN-based models, except for SSD, had similar or better performance compared to the two-stage CNN-based models. YOLOv5su showed the best performance on mAP_{50} and $mAP_{50:95}$, which were 0.989 and 0.723, respectively. This indicates that YOLOv5su can predict the bounding boxes of "Nauplius Live" and "Nauplius Shell" as annotated by the experts well. In addition, YOLOv5su also outperformed the other one-stage models in terms of precision, recall, and F1 score, reaching 0.984, 0.977, and 0.98, respectively. YOLOv8n was close to YOLOv5su in mAP_{50} and precision, F1 score, but YOLOv8n showed lower performance in $mAP_{50:95}$ and recall than YOLOv5su, which were 0.710 and 0.965 respectively. Except for YOLOv3, the overall performance of the YOLO series was good, with $mAP_{50:95}$ over 0.64 and F1 scores over 0.96. The SSD series performed poorly across all metrics. RetinaNet models showed a good performance in mAP_{50}, $mAP_{50:95}$, and precision. However, the recall value was lower, indicating some objects were not accurately detected.

Among the Transformer-based models, RT-DETR performed the best in all metrics, even outperforming the deformable DETR in terms of model size and speed. In terms of speed, RT-DETR achieved 46.9 FPS.

A.3.1.2 Detection results for each class

The F1 scores of each model for "Nauplius Live" and "Nauplius Shell" in the test set of Dataset NLS and inference speed are visualized in Figure A.1 and Figure A.2. YOLO series models, SSD series models, and RT-DETR achieved relatively high inference speeds. In general, the performance of the models was slightly worse on the "Nauplius Shell" compared to the "Nauplius Live." Nevertheless, it is worth noting that the distribution of F1 scores among the different models in both classes was relatively similar. RT-DETR achieved the highest F1 scores in both the "Nauplius Live" and "Nauplius Shell," which were 0.991 and 0.980, respectively. YOLOv5su achieved an F1 score that is close to RT-DETR, but its inference speed is much faster and the model size is smaller than RT-DETR. YOLOv3 and YOLOX had opposite performances for "Nauplius Live" and "Nauplius Shell": YOLOv3 showed better performance on the F1 metric in class "Nauplius Live" than YOLOX but worse performance for class "Nauplius Shell." A similar reversal was observed for both SSD models.

In addition to the F1 scores, the results for each metric obtained by each model for the "Nauplius Live" and "Nauplius Shell" in the test set of dataset NLS are shown in Tables A.3, and A.4.

A.3.2 Results of salmon lice detection in Dataset NC

The results for each metric obtained by each model for the "Nauplius Live," "Nauplius Shell," and "Copepodite" in the test set of dataset NC are shown in Table A.5, A.6, and A.7.

A.3.3 Results of salmon lice detection in seawater using different training strategy

In machine learning, there is no clear guidance for the specific partitioning of the dataset. To investigate whether the proportion of dataset divisions has an effect on model performance, we employed a 5:3:2 ratio to further split the Dataset NC, resulting in a training set containing 4157 images, a validation set of 2495 images, and a test set of 1664 images (referred as Dataset NC532). Subsequently, we retrained both the YOLOv8n and YOLOv5su models on the training set of Dataset NC532 and tested on the test set of Dataset NC532. The results indicate that models trained with a dataset split ratio of 5:3:2 demonstrate comparable overall performance (Table A.8). In comparison to the 7:1:2 split ratio method, the model's precision decreases slightly

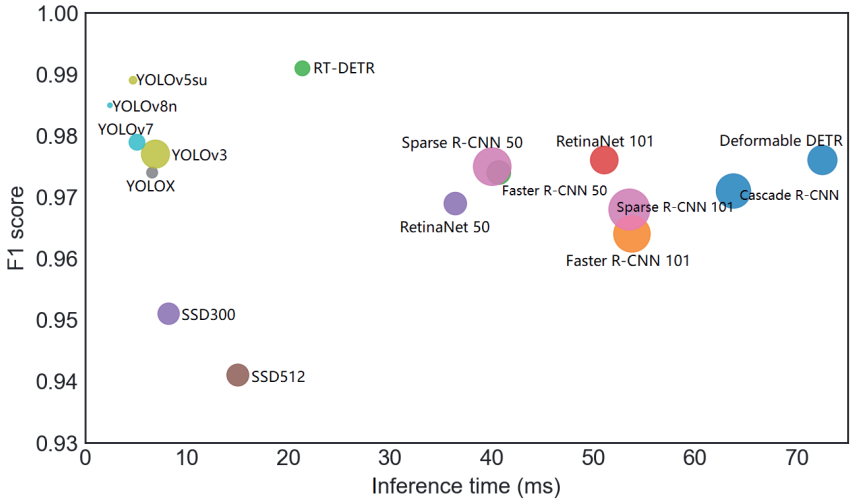


Figure A.1: Detection results of "Nauplius Live" on the test set of Dataset NLS. The size of the bubbles represents the size of the models. Bubbles positioned closer to the top-left corner refer to models that show better performance (high effectiveness and low inference time).

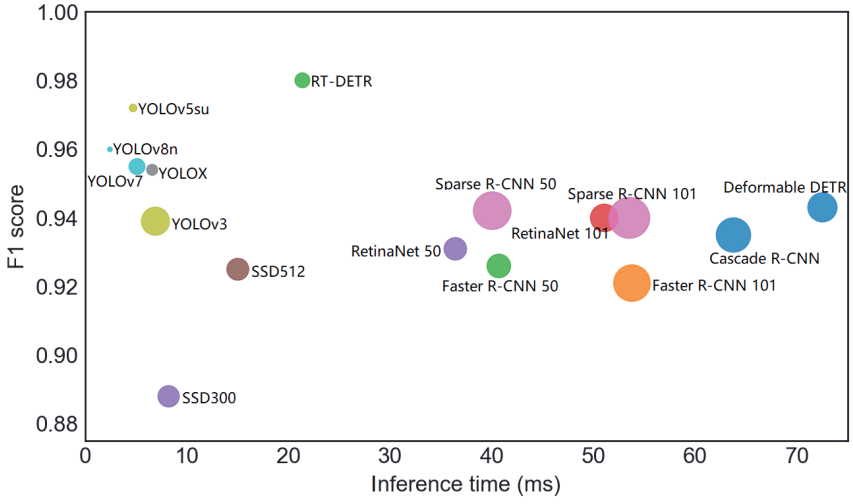


Figure A.2: Detection results of "Nauplius Shell" on the test set of Dataset NLS. The size of the bubbles represents the size of the models. Bubbles positioned closer to the top-left corner refer to models that show better performance (high effectiveness and low inference time).

Table A.2: Overall detection results on the test set of Dataset NLS.

Model category	Model	mAP_{50}	$mAP_{50:95}$	Precision	Recall	F1	Model size (MB)	FPS	Inference time (ms)
Two-stage CNN-based	Faster-R-CNN 50	0.970	0.607	0.959	0.941	0.950	167.3	24.6	40.65
	Faster-R-CNN 101	0.967	0.599	0.958	0.928	0.943	399.5	18.6	53.76
	Cascade R-CNN	**0.973**	**0.641**	0.967	0.939	0.953	354.1	15.7	63.69
	Sparse R-CNN50	0.973	0.624	**0.973**	**0.945**	**0.959**	425.5	**25.0**	40.00
	Sparse R-CNN101	0.969	0.632	0.967	0.942	0.954	501.7	18.7	53.48
One-stage CNN-based	SSD300	0.930	0.546	0.932	0.908	0.919	137.2	122.2	8.18
	SSD512	0.948	0.564	0.945	0.923	0.933	144.2	66.8	14.97
	RetinaNet 50	0.974	0.621	0.962	0.938	0.950	152.1	27.5	36.50
	RetinaNet 101	0.978	0.630	0.972	0.944	0.958	228.4	19.6	51.02
	YOLOv3	0.965	0.565	0.952	0.964	0.958	236.6	144.5	6.92
	YOLOv5su	**0.989***	**0.723***	**0.984**	**0.977**	**0.980**	17.6	212.8	4.70
	YOLOX	0.975	0.649	0.968	0.961	0.964	36.1	151.6	6.70
	YOLOv7	0.985	0.667	0.965	0.970	0.967	75.6	196.1	5.10
	YOLOv8n	0.988	0.710	0.980	0.965	0.972	6.5	**416.0***	2.40
Transformer	Deformable DETR	0.962	0.603	0.969	0.950	0.960	254.7	13.8	72.46
	RT-DETR	**0.988**	**0.700**	**0.988***	**0.983***	**0.985***	66.5	**46.9**	21.32

Note: A green background with $*$ highlights that this value was the highest among all models for that specific metric. A value in bold font signifies that it holds the highest position for that metric within the model category, encompassing two-stage CNN-based models, one-stage CNN-based models, and Transformer-based models. The mAP_{50} stands for the mean value of average precision of all classes when the intersection over union (IoU) is equal to 50%. The $mAP_{50:95}$ stands for the average value of from mAP_{50} to mAP_{95} in 5 % intervals of IoU. Precision stands for the probability of being a positive sample out of all the samples predicted to be positive. Recall is the probability of being predicted as a positive sample among the actual positive samples. The F1 score stands for the harmonic mean of precision and recall. The model size stands for the space needed for model storage. FPS (Frame Per Second) stands for the number of images the model can process per second. The inference time stands for the time used by the model to infer one image to the output result, including classes of objects and locations of objects.

while the recall rate improves. However, overall, models trained with the 5:3:2 dataset split ratio exhibit similar overall performance to those trained with the 7:1:2 ratio. Further, we tested the models' performance on real seawater samples. The results indicate that with a 5:3:2 division of data, YOLOv8 successfully detected 6 out of 7 and 4 out of 5 salmon lice, respectively. On the other hand, YOLOv5 detected all 7 out of 7 and 5 out of 5 results. This indicates that the division of data has an impact on model performance. Different datasets may require varying optimal division ratios. Our future research will include augmenting the dataset with real-world images of seawater and exploring the effects of different allocation ratios on the dataset on model performance.

Table A.3: Nauplius Live detection results for each model in the test set of Dataset NLS.

	Model	AP_{50}	$AP_{50:95}$	Precision	Recall	F1
Two-stage	Faster-R-CNN 50	0.986	0.647	0.973	0.974	0.974
	Faster-R-CNN 101	0.985	0.639	0.966	0.962	0.964
	Cascade R-CNN	**0.987**	**0.686**	0.975	0.966	0.971
	Sparse R-CNN50	0.984	0.663	**0.985**	**0.966**	**0.975**
	Sparse R-CNN101	0.979	0.662	0.978	0.958	0.968
One-stage	SSD300	0.963	0.594	0.952	0.950	0.951
	SSD512	0.954	0.586	0.960	0.922	0.941
	RetinaNet 50	0.986	0.661	0.970	0.967	0.969
	RetinaNet 101	0.988	0.672	0.980	0.972	0.976
	YOLOv3	0.984	0.611	0.968	0.985	0.977
	YOLOv5su	**0.994**	**0.764***	**0.99**	**0.989**	**0.989**
	YOLOX	0.987	0.684	0.967	0.982	0.974
	YOLOv7	0.994	0.705	0.971	0.987	0.979
	YOLOv8n	**0.994**	0.754	**0.99**	0.981	0.985
Transformer	Deformable DETR	0.981	0.641	0.985	0.968	0.976
	RT-DETR	**0.995***	**0.735**	**0.992***	**0.99***	**0.991***

Note: A green background with $*$ highlights that this value represents the highest among all models for that specific metric. A value in bold font signifies that it holds the highest position for that metric within the model category, encompassing two-stage CNN-based models, one-stage CNN-based models, and Transformer-based models. The mAP_{50} stands for the mean value of average precision of all classes when the intersection over union (IoU) is equal to 50%. The $mAP_{50:95}$ stands for the average value of from mAP_{50} to mAP_{95} in 5 % intervals of IoU. Precision stands for the probability of being a positive sample out of all the samples predicted to be positive. Recall is the probability of being predicted as a positive sample among the actual positive samples. The F1 score stands for the harmonic mean of precision and recall.

Table A.4: Nauplius shell detection results for each model in the test set of Dataset NLS.

	Model	AP_{50}	$AP_{50:95}$	Precision	Recall	F1
Two-stage	Faster-R-CNN 50	0.955	0.567	0.946	0.908	0.926
	Faster-R-CNN 101	0.950	0.559	0.950	0.893	0.921
	Cascade R-CNN	0.959	**0.595**	0.959	0.912	0.935
	Sparse R-CNN50	**0.961**	0.586	**0.961**	0.924	**0.942**
	Sparse R-CNN101	0.959	0.584	0.956	**0.925**	0.940
One-stage	SSD300	0.896	0.499	0.912	0.865	0.888
	SSD512	0.942	0.541	0.927	0.924	0.925
	RetinaNet 50	0.963	0.581	0.954	0.909	0.931
	RetinaNet 101	0.968	0.589	0.965	0.916	0.940
	YOLOv3	0.946	0.520	0.935	0.943	0.939
	YOLOv5su	**0.984***	**0.683***	**0.977**	**0.967**	**0.972**
	YOLOX	0.964	0.614	0.969	0.940	0.954
	YOLOv7	0.976	0.629	0.958	0.953	0.955
	YOLOv8n	0.982	0.665	0.969	0.950	0.960
Transformer	Deformable DETR	0.943	0.564	0.954	0.933	0.943
	RT-DETR	**0.982**	**0.664**	**0.984***	**0.976***	**0.98***

Note: A green background with $*$ highlights that this value represents the highest among all models for that specific metric. A value in bold font signifies that it holds the highest position for that metric within the model category, encompassing two-stage CNN-based models, one-stage CNN-based models and Transformer-based models. The mAP_{50} stands for the mean value of average precision of all classes when the intersection over union (IoU) is equal to 50%. The $mAP_{50:95}$ stands for the average value of from mAP_{50} to mAP_{95} in 5 % intervals of IoU. Precision stands for the probability of being a positive sample out of all the samples predicted to be positive. Recall is the probability of being predicted as a positive sample among the actual positive samples. The F1 score stands for the harmonic mean of precision and recall.

Table A.5: Nauplius Live detection results on the test set of Dataset NC.

	Model	mAP_{50}	$mAP_{50:95}$	Precision	Recall	F1
Two-stage	Faster-R-CNN 50	0.973	0.634	0.960	0.952	0.956
	Faster-R-CNN 101	0.973	0.633	0.963	0.956	0.959
	Cascade R-CNN	0.975	**0.663**	**0.970**	**0.967**	**0.969**
	Sparse R-CNN50	0.975	0.658	0.949	0.959	0.954
	Sparse R-CNN101	**0.977**	0.657	0.966	0.961	0.964
One-stage	SSD300	0.952	0.574	0.950	0.937	0.944
	SSD512	0.946	0.588	0.916	0.911	0.913
	RetinaNet 50	0.975	0.647	0.966	0.941	0.953
	RetinaNet 101	0.979	0.660	0.960	0.961	0.960
	YOLOv3	0.963	0.594	0.972	0.971	0.972
	YOLOv5su	0.987	**0.759***	0.988	0.970	0.979
	YOLOX	0.986	0.698	**0.991***	**0.984**	**0.988***
	YOLOv7	**0.994***	0.723	0.985	0.983	0.984
	YOLOv8n	0.989	0.741	0.984	0.971	0.977
Transformer	Deformable DETR	0.980	0.656	0.985	0.974	0.979
	RT DETR	**0.993**	**0.730**	**0.988**	**0.988***	**0.988**

Note: A green background with $*$ highlights that this value represents the highest among all models for that specific metric. A value in bold font signifies that it holds the highest position for that metric within the model category, encompassing two-stage CNN-based models, one-stage CNN-based models and Transformer-based models. The mAP_{50} stands for the mean value of average precision of all classes when the intersection over union (IoU) is equal to 50%. The $mAP_{50:95}$ stands for the average value of from mAP_{50} to mAP_{95} in 5 % intervals of IoU. Precision stands for the probability of being a positive sample out of all the samples predicted to be positive. Recall is the probability of being predicted as a positive sample among the actual positive samples. The F1 score stands for the harmonic mean of precision and recall.

Table A.6: Nauplius Shell detection results on the test set of Dataset NC.

	Model	mAP_{50}	$mAP_{50:95}$	Precision	Recall	F1
Two-stage	Faster-R-CNN 50	0.935	0.556	**0.936**	0.876	0.905
	Faster-R-CNN 101	0.927	0.546	0.917	0.866	0.891
	Cascade R-CNN	0.936	**0.586**	0.929	0.888	0.908
	Sparse R-CNN50	**0.945**	0.580	0.905	**0.935**	**0.920**
	Sparse R-CNN101	0.935	**0.586**	0.913	0.917	0.915
One-stage	SSD300	0.874	0.485	0.940	0.838	0.886
	SSD512	0.901	0.518	0.892	0.879	0.886
	RetinaNet 50	0.939	0.564	0.939	0.868	0.903
	RetinaNet 101	0.942	0.581	0.922	0.881	0.901
	YOLOv3	0.928	0.508	0.930	0.911	0.920
	YOLOv5su	**0.973***	**0.678***	**0.965***	0.939	0.952
	YOLOX	0.959	0.629	0.961	0.937	0.949
	YOLOv7	0.971	0.655	0.958	**0.949**	**0.953***
	YOLOv8n	0.967	0.662	0.959	0.939	0.949
Transformer	Deformable DETR	**0.961**	0.606	**0.962**	0.938	0.950
	RT DETR	0.960	**0.649**	0.957	**0.950***	**0.953***

Note: A green background with $*$ highlights that this value represents the highest among all models for that specific metric. A value in bold font signifies that it holds the highest position for that metric within the model category, encompassing two-stage CNN-based models, one-stage CNN-based models, and Transformer-based models. The mAP_{50} stands for the mean value of average precision of all classes when the intersection over union (IoU) is equal to 50%. The $mAP_{50:95}$ stands for the average value of from mAP_{50} to mAP_{95} in 5 % intervals of IoU. Precision stands for the probability of being a positive sample out of all the samples predicted to be positive. Recall is the probability of being predicted as a positive sample among the actual positive samples. The F1 score stands for the harmonic mean of precision and recall.

Table A.7: Copepodite detection results on the test set of Dataset NC.

	Model	mAP_{50}	$mAP_{50:95}$	Precision	Recall	F1
Two-stage	Faster-R-CNN 50	0.979	0.715	0.953	0.953	0.953
	Faster-R-CNN 101	0.968	0.706	0.935	0.944	0.940
	Cascade R-CNN	0.972	0.749	0.960	0.958	0.959
	Sparse R-CNN50	**0.991**	**0.766**	**0.991***	**0.961**	**0.976**
	Sparse R-CNN101	0.988	0.753	0.979	**0.961**	0.970
One-stage	SSD300	0.930	0.594	0.941	0.896	0.918
	SSD512	0.861	0.520	0.923	0.771	0.840
	RetinaNet 50	0.974	0.702	0.940	0.940	0.940
	RetinaNet 101	0.978	0.722	0.932	0.942	0.937
	YOLOv3	0.968	0.626	0.974	0.974	0.974
	YOLOv5su	**0.991***	**0.839***	**0.979**	0.984	**0.981**
	YOLOX	0.977	0.779	0.970	**0.986**	0.978
	YOLOv7	0.987	0.814	0.961	0.979	0.970
	YOLOv8n	0.987	0.817	0.973	**0.986**	0.980
Transformer	Deformable DETR	0.982	0.725	**0.986**	0.967	0.976
	RT DETR	**0.987**	**0.830**	0.972	**0.995***	**0.983***

Note: A green background with $*$ highlights that this value represents the highest among all models for that specific metric. A value in bold font signifies that it holds the highest position for that metric within the model category, encompassing two-stage CNN-based models, one-stage CNN-based models and Transformer-based models. The mAP_{50} stands for the mean value of average precision of all classes when the intersection over union (IoU) is equal to 50%. The $mAP_{50:95}$ stands for the average value of from mAP_{50} to mAP_{95} in 5 % intervals of IoU. Precision stands for the probability of being a positive sample out of all the samples predicted to be positive. Recall is the probability of being predicted as a positive sample among the actual positive samples. The F1 score stands for the harmonic mean of precision and recall.

Table A.8: Overall detection results on the test set of Dataset NC and Dataset NC532 using models trained with different dataset ratios.

Dataset (split ratio)	Model	mAP_{50}	$mAP_{50:95}$	Precision	Recall	F1
Dataset NC (7:1:2)	YOLOv5su	0.984	0.759	0.977	0.964	0.970
	YOLOv8n	0.981	0.740	0.972	0.965	0.968
Dataset NC532 (5:3:2)	YOLOv5su	0.984	0.749	0.976	0.970	0.973
	YOLOv8n	0.985	0.743	0.961	0.967	0.964

Appendix B

Additional methodological details and results related to Chapter 3

B.1 Result of the orthogonal experiments

Table B.1 presents the results of the orthogonal experiment under various factor configurations.

Table B.1: Result of the orthogonal experiment under various factor configurations.

NO.	*BOI*	*SIM*	*MNS*	*IST*	*IOT*	*MNN*	*CA*	*IA*	*MB*	*GB*	F1-score
1	1	1	1	1	1	1	1	1	1	1	0.523611
2	1	1	1	1	1	1	1	1	2	2	0.579661
3	1	1	1	1	1	1	1	1	3	3	0.574070
4	1	2	2	2	1	2	2	2	1	1	0.672244
5	1	2	2	2	1	2	2	2	2	2	0.805876
6	1	2	2	2	1	2	2	2	3	3	0.841302
7	1	3	3	3	1	3	3	3	1	1	0.819046
8	1	3	3	3	1	3	3	3	2	2	0.856989
9	1	3	3	3	1	3	3	3	3	3	0.848108
10	2	1	1	2	1	2	3	3	1	1	0.703375
11	2	1	1	2	1	2	3	3	2	2	0.611808
12	2	1	1	2	1	2	3	3	3	3	0.535367
13	2	2	2	3	1	3	1	1	1	1	0.861525
14	2	2	2	3	1	3	1	1	2	2	0.832795
15	2	2	2	3	1	3	1	1	3	3	0.852178
16	2	3	3	1	1	1	2	2	1	1	0.684132
17	2	3	3	1	1	1	2	2	2	2	0.689738
18	2	3	3	1	1	1	2	2	3	3	0.716438
19	3	1	2	1	1	3	2	3	1	2	0.812801
20	3	1	2	1	1	3	2	3	2	3	0.750244
21	3	1	2	1	1	3	2	3	3	1	0.744810
22	3	2	3	2	1	1	3	1	1	2	0.662645
23	3	2	3	2	1	1	3	1	2	3	0.585002
24	3	2	3	2	1	1	3	1	3	1	0.628720
25	3	3	1	3	1	2	1	2	1	2	0.824631
26	3	3	1	3	1	2	1	2	2	3	0.845573
27	3	3	1	3	1	2	1	2	3	1	0.810709
28	1	1	3	3	2	2	2	1	1	3	0.697376
29	1	1	3	3	2	2	2	1	2	1	0.635946
30	1	1	3	3	2	2	2	1	3	2	0.767488
31	1	2	1	1	2	3	3	2	1	3	0.833643
32	1	2	1	1	2	3	3	2	2	1	0.839043
33	1	2	1	1	2	3	3	2	3	2	0.819265
34	1	3	2	2	2	1	1	3	1	3	0.709630

Table B.1 continued from previous page

NO.	*BOI*	*SIM*	*MNS*	*IST*	*IOT*	*MNN*	*CA*	*IA*	*MB*	*GB*	F1-score
35	1	3	2	2	2	1	1	3	2	1	0.706971
36	1	3	2	2	2	1	1	3	3	2	0.707334
37	2	1	2	3	2	1	3	2	1	2	0.687186
38	2	1	2	3	2	1	3	2	2	3	0.529059
39	2	1	2	3	2	1	3	2	3	1	0.628864
40	2	2	3	1	2	2	1	3	1	2	0.820113
41	2	2	3	1	2	2	1	3	2	3	0.855453
42	2	2	3	1	2	2	1	3	3	1	0.861271
43	2	3	1	2	2	3	2	1	1	2	0.850219
44	2	3	1	2	2	3	2	1	2	3	0.847696
45	2	3	1	2	2	3	2	1	3	1	0.835093
46	3	1	3	2	2	3	1	2	1	3	0.797924
47	3	1	3	2	2	3	1	2	2	1	0.733926
48	3	1	3	2	2	3	1	2	3	2	0.652043
49	3	2	1	3	2	1	2	3	1	3	0.712308
50	3	2	1	3	2	1	2	3	2	1	0.636623
51	3	2	1	3	2	1	2	3	3	2	0.620599
52	3	3	2	1	2	2	3	1	1	3	0.879146
53	3	3	2	1	2	2	3	1	2	1	0.855924
54	3	3	2	1	2	2	3	1	3	2	0.882317

Appendix C

Additional methodological details and results related to Chapter 6

C.1 Examples of datasets

Table C.1: Example from the single-hop QA dataset illustrating a question–answer (QA) pair.

Question	Which measures must be taken during the preparation of fish for a medicinal bath to prevent a dangerous buildup of ammonia from waste?
Answer	Starve fish before treatments to ensure intestines are empty and reduce the risk of total ammonia nitrogen accumulation from faeces.
Source Chunk	*Handling injuries during winter can lead to secondary bacterial infections such as* Moritella viscosa *and* Vibrio spp. *Carbon dioxide can accumulate in treatment chambers when system water flow is inadequate or when the biological load exceeds design capacity. Temperature, pH, and salinity affect the* NH_3*:*NH_4^+ *ratio and thereby ammonia toxicity in treatment water. The maximum safe short-term (4-hour) exposure limit for unionized ammonia (*NH_3*-N) is 0.1 mg/L. Starve fish before treatments to ensure intestines are empty and reduce the risk of total ammonia nitrogen (TAN) accumulation from faeces. For some substances, such as hydrogen peroxide, measure the concentrations directly in treatment water and record exposure duration* (Noble et al., 2018)

Table C.2: Example of a multi-hop QA pair from the evaluation dataset.

Question	What are the two main limitations for the development of a commercially available salmon louse vaccine, and what is the key challenge that arises from the need to understand ectoparasite-host interactions?
Answer	The two main limitations for the development of a commercially available salmon louse vaccine are the discovery of protective antigens and the need to address vaccine formulation, safety, administration, and efficacy to obtain regulatory approval. The key challenge that arises from the need to understand the complex molecular interactions between vertebrate hosts and ectoparasites is the discovery of key pathway molecules that mediate these ectoparasite-host interactions.
Source Chunk 1	*However, no commercially available anti-sea lice vaccines exist. Previously published results suggest that the discovery of protective antigens, vaccine formulation, safety, administration, and efficacy need to be addressed to obtain regulatory approval and advance in the implementation of vaccination strategies for the control of salmon lice and other fish ectoparasites*
Source Chunk 2	*The challenge of developing vaccines against ectoparasites arises from the need to understand the complex molecular interactions between vertebrate hosts and ectoparasites, which require the discovery of key pathway molecules that mediate ectoparasite–host interactions*

C.2 Case study

This section presents the farm status, recent treatment history, and DSS-generated recommendation for farm A at Week 18 of 2025.

C.2.1 Farm Background

- **Site name and number:** Farm A
- **Country/region:** Not provided
- **Production species:** Not provided
- **Year:** 2025
- **Assessment period:** Weeks 14–18

C.2.2 Recent Farm Status

Table C.3 summarizes the key lice and environmental indicators during Weeks 14–18.

Table C.3: Key farm metrics for farm A during Weeks 14–18 of 2025. AF = Adult female lice.

Week	Adult Female Lice	Mobile Lice	Stationary Lice	Sea Temperature (°C)	AF Limit
14	0.08	0.18	0.00	6.54	0.5
15	0.08	0.18	0.00	7.67	0.5
16	0.11	0.23	0.00	7.01	0.2
17	0.14	0.27	0.00	7.20	0.2
18	0.17	0.26	0.00	7.73	0.2

The farm remained below the regulatory threshold at Week 18, with an adult female lice count of 0.17 compared with the limit of 0.2. However, adult female lice increased continuously from Week 14 to Week 18. Mobile lice also increased from Week 14 to Week 17, followed by a slight decline in Week 18, while stationary lice remained at zero throughout the observation period. Sea temperature showed a gradual overall increase during the same period.

C.2.3 Recent Treatment History

During the period from Week 48 of 2024 to Week 10 of 2025, the farm received three azamethiphos bath treatments and one combined thermal/freshwater treatment. Overall, these interventions produced short-term reductions in lice levels, although the recent upward trend suggests renewed infestation pressure.

Treatment in Week 8 of 2025: Azamethiphos bath Between Week 9 and Week 10, adult female lice decreased from 0.37 to 0.18, and mobile lice decreased from 0.87 to 0.32, while stationary lice remained at 0.00. A further decline was observed in Week 11, when adult female lice dropped to 0.07, indicating short-term efficacy without an immediate rebound.

Treatment in Week 10 of 2025: Thermal + freshwater Between Week 10 and Week 11, adult female lice decreased from 0.18 to 0.07, and mobile lice decreased from 0.32 to 0.05, while stationary lice remained at 0.00. Because thermal and freshwater treatment were applied together, their individual effects cannot be separated. Adult female lice remained low at 0.06 during Weeks 12 and 13.

C.2.4 Decision Context

The management question considered by the DSS was:

> What treatment action should be taken for salmon lice management at farm A in Week 18 of 2025?

C.2.5 DSS Recommendation

The DSS recommended implementing a tarpaulin-lined in-cage freshwater bath within the following 7 days (Week 18–19), to reduce adult female lice and maintain the farm below the regulatory threshold.

C.2.6 Rationale for the Recommendation

This recommendation was based on four main considerations. First, although the farm remained compliant at Week 18, the adult female lice count (0.17) was close to the legal limit (0.2). Second, the increasing trend in adult female lice over four weeks, together with sustained mobile lice pressure, indicated a substantial risk of exceeding the threshold in the near future. Third, the recent treatment history suggested that repeated use of azamethiphos or other high-impact interventions could increase resistance risk and impose additional welfare burdens. Fourth, freshwater treatment was judged to offer a better balance between expected efficacy, operational feasibility, and fish welfare than the main alternative options.

C.2.7 Implementation Considerations

Before treatment, the farm should verify permit requirements, freshwater source quality, and equipment availability. During execution, oxygen, temperature, pH, and fish

behavior should be continuously monitored. Post-treatment lice counts should be performed at 7 and 14 days after treatment to assess efficacy. As a contingency option, hydrogen peroxide treatment may be considered only if sensitivity documentation is available.

C.2.8 Monitoring and Risk Factors

The primary monitoring variable is the weekly adult female lice count, supported by secondary monitoring of fish welfare indicators and feeding response. The regulatory hard limit for adult female lice is 0.2. The observed value at Week 18 was 0.17, which is close to this limit and should therefore be treated as an early warning signal, even though the farm remained technically compliant.

Several uncertainties may affect the effectiveness or feasibility of the plan. These include logistical constraints related to tarpaulin systems and staffing, variation in lice sensitivity to freshwater, fish stress during crowding and bathing, weather conditions required for safe treatment, and the possibility that site-level averages may conceal cage-level peaks.

C.2.9 Alternative Option Considered

The main alternative considered by the DSS was a hydrogen peroxide bath. This option was ranked lower because it involves stricter documentation requirements, possible resistance concerns, and a higher expected welfare and cost burden than freshwater treatment.

C.2.10 Traceability of Case-Study Evidence

Table C.4 summarizes the provenance of the case-study elements used in the farm A analysis. Farm-specific observations are traceable to the structured farm-status input, while external technical and regulatory evidence is traceable to specific retrieved knowledge-base chunks. The Evidence Type column distinguishes farm-state data, binding regulation, technical guidelines, scientific literature, and system-derived outputs.

Table C.4: Traceability matrix for the farm A Week 18 case study.

Element	Source	Chunk ID	Evidence Type
Weekly lice and temperature records (W14–18)	Barentswatch	–	Farm-state data
Recent treatment history	Barentswatch	–	Farm-state data
AF regulatory threshold (0.2, W16–21, Nord-Trøndelag and south)	Forskrift om bekjempelse av lakselus i akvakulturanlegg, §8 (as amended by regulation of 6 March 2017 No. 275)	`e873b026`; `4b3fb08c`	Regulation
Lice-pressure and treatment context	Marques (2018)	`efe3bada`	Literature
Azamethiphos resistance concern	Resistance study (2026)	`51db09cb`	Literature
Treatment rotation rationale	Chile lice review (2023); Resistance action plan (2016)	`6a5d0423`; `8bc5c583`	Literature; Guideline
Thermal treatment welfare burden	Delousing best-practice report (2023)	`82f0277d`	Literature
H_2O_2 constraints: documentation, resistance, welfare, cost	Resistance plan (2016); mortality/H_2O_2 studies (2018/2025); cost study (2024)	`8bc5c583`; `aea67382`; `efef0f2c`; `931a3f9d`	Guideline; Literature
Freshwater treatment efficacy	Freshwater treatment study (2023)	`feab3f09`	Literature
Freshwater water-quality checks	Welfare Indicators handbook (2018); Tiltaksveileder (2020)	`a6bddaf4`; `dbaddb1c`	Guideline
Pre-/post-treatment counting protocol	Tiltaksveileder (2020)	`9fdade90`	Guideline
Bath-treatment site/discharge restrictions	Environmental discharge guide (2019)	`9f45692d`	Guideline
Operational feasibility of bath logistics	Overton (2019)	`6c1528cd`	Literature
Final recommendation (freshwater bath; H_2O_2 contingency)	DSS output	–	System-derived

References

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P. et al. (2024). Phi-4 technical report. arXiv:2412.08905.

Adams, T. P., Proud, R., & Black, K. D. (2015). Connected networks of sea lice populations: dynamics and implications for control. *Aquaculture Environment Interactions*, *6*, 273–284. doi: https://doi.org/10.3354/aei00133.

Akram, W., Din, M. U., Saad Saoud, L., & Hussain, I. (2026). A review of generative AI in aquaculture: Applications, case studies and challenges for smart and sustainable farming. *Aquacultural Engineering*, *112*, 102637. doi: 10.1016/j.aquaeng.2025.102637.

Alaba, S. Y., Nabi, M. M., Shah, C., Prior, J., Campbell, M. D., Wallace, F., Ball, J. E., & Moorhead, R. (2022). Class-aware fish species recognition using deep learning for an imbalanced dataset. *Sensors*, *22*, 8268. doi: 10.3390/s22218268.

Alon-Barkat, S., & Busuioc, M. (2023). Human-AI interactions in public sector decision-making: "automation bias" and "selective adherence" to algorithmic advice. *Journal of Public Administration Research and Theory*, *33*, 153–169. doi: 10.1093/jopart/muac007.

Amatriain, X. (2024). Prompt design and engineering: Introduction and advanced methods. *arXiv preprint arXiv:2401.14423*, .

Arslan, M., Ghanem, H., Munawar, S., & Cruz, C. (2024). A survey on RAG with LLMs. *Procedia Computer Science*, *246*, 3781–3790. doi: 10.1016/j.procs.2024.09.178.

ASA, M. (2025). Mowi ASA (OSE: MOWI): 2025 salmon farming industry handbook.

Asche, F. et al. (2018). Three pillars of sustainability in fisheries. *Proceedings of the National Academy of Sciences*, *115*, 11221–11225. doi: 10.1073/pnas.1807677115.

Asplin, L., Boxaspen, K. K., & Sandvik, A. D. (2011). *Modeling the Distribution and Abundance of Planktonic Larval Stages of Lepeophtheirus salmonis in Norway*. John Wiley & Sons, Ltd. doi: https://doi.org/10.1002/9780470961568.ch1.

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., & Ballas, N. (2023). Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 15619–15629). Los Alamitos, CA, USA: IEEE Computer Society. doi: 10.1109/CVPR52729.2023.01499.

Aunsmo, A., Persson, D., Stormoen, M., Romstad, S., Jamtøy, O., & Midtlyng, P. J. (2023). Real-time monitoring of cause-specific mortality- and losses in industrial salmon farming. *Aquaculture*, *563*, 738969. doi: https://doi.org/10.1016/j.aquaculture.2022.738969.

Azevedo, M. L., Silva, T., Soares, F., Budaev, S., Conceição, L. E. C., & Rønnestad, I. (2025). Development of a novel reference feed intake model for Atlantic salmon (*Salmo salar*) based on temperature and body weight. *Aquacultural Engineering*, *111*, 102562. doi: 10.1016/j.aquaeng.2025.102562.

Ba, Y., Mancenido, M. V., & Pan, R. (2024). How does data diversity shape the weight landscape of neural networks? arXiv:2410.14602.

Badshah, S., & Sajjad, H. (2025). Reference-guided verdict: LLMs-as-Judges in automatic evaluation of free-form QA. arXiv:2408.09235.

Bang, Y., Ji, Z., Schelten, A., Hartshorn, A., Fowler, T., Zhang, C., Cancedda, N., & Fung, P. (2025). Hallulens: LLM hallucination benchmark. doi: 10.48550/arXiv.2504.17550 arXiv:2504.17550 [cs].

Banno, K., Gonçalves, F. M. F., Sauphar, C., Anichini, M., Hazelaar, A., Sperre, L. H., Stolz, C., Aas, G. H., Gansel, L. C., & da Silva Torres, R. (2024). Identifying losers: Automatic identification of

growth-stunted salmon in aquaculture using computer vision. *Machine Learning with Applications*, *16*, 100562. doi: 10.1016/j.mlwa.2024.100562.

Baradad Jurjo, M., Wulff, J., Wang, T., Isola, P., & Torralba, A. (2021). Learning to see by looking at noise. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems* (pp. 2556–2569). Curran Associates, Inc. volume 34.

Barnard, C. J., Behnke, J. M., & Sewell, J. (1996). Social status and resistance to disease in house mice (mus musculus): Status-related modulation of hormonal responses in relation to immunity costs in different social and physical environments. *Ethology*, *102*, 63–84. doi: https://doi.org/10.1111/j.1439-0310.1996.tb01104.x. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1439-0310.1996.tb01104.x.

Bartussek, H. (1999). A review of the animal needs index (ani) for the assessment of animals' well-being in the housing systems for austrian proprietary products and legislation. *Livestock Production Science*, *61*, 179–192. doi: https://doi.org/10.1016/S0301-6226(99)00067-6.

Benthin, J., Kauselmann, K., Krause, E. T., Bracke, M. B. M., & Vonholdt-Wenker, M. L. (2026). Semantic modelling of animal welfare explained – part 1: Calculating overall welfare scores for husbandry systems using the anywel model framework. *Archives Animal Breeding*, *69*, 363–382. doi: 10.5194/aab-69-363-2026.

Biazi, V., & Marques, C. (2023). Industry 4.0-based smart systems in aquaculture: A comprehensive review. *Aquacultural Engineering*, *103*, 102360. doi: 10.1016/j.aquaeng.2023.102360.

Blanquero, R., Carrizosa, E., Ramírez-Cobo, P., & Sillero-Denamiel, M. R. (2021). Variable selection for naïve bayes classification. *Computers & Operations Research*, *135*, 105456. doi: https://doi.org/10.1016/j.cor.2021.105456.

Blokhuis, H. (2007). Welfare Quality: context, progress and aims. In *Assuring animal welfare: from societal concerns to implementation, Second Welfare Quality stakeholder conference, 3–4 May 2007* (pp. 9–12). Berlin Germany.

Boerlage, A. S., Shrestha, S., Leinonen, I., Jansen, M. D., Revie, C. W., Reeves, A., & Toma, L. (2024). Sea lice management measures for farmed Atlantic salmon (*Salmo salar*) in Scotland: Costs and effectiveness. *Aquaculture*, *580*, 740274. doi: 10.1016/j.aquaculture.2023.740274.

Borchel, A., & Nilsen, F. (2026). A review of the salmon louse (Lepeophtheirus salmonis) hyposalinity responses and the efficacy of freshwater delousing. *Reviews in Fisheries Science & Aquaculture*, *34*, 211–222. doi: 10.1080/23308249.2025.2536316. _eprint: https://doi.org/10.1080/23308249.2025.2536316.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* COLT '92 (p. 144–152). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/130385.130401.

Bracke, M. (2008). RICHPIG: a semantic model to assess enrichment materials for pigs. *Animal Welfare*, *17*, 289–304. doi: https://doi.org/10.1017/S0962728600032218.

Bracke, M., Metz, J., Spruijt, B., & Schouten, W. (2002). Decision support system for overall welfare assessment in pregnant sows b: validation by expert opinion. *Journal of Animal Science*, *80*, 1835–1845. doi: https://doi.org/10.2527/2002.8071835x.

Bracke, M., Spruijt, B., & Metz, J. H. M. (1999a). Overall animal welfare assessment reviewed. Part 1: Is it possible? *Assessment of Animal Welfare Collection*, .

Bracke, M. B., Hulsegge, B., Keeling, L., & Blokhuis, H. J. (2004). Decision support system with semantic model to assess the risk of tail biting in pigs: 1. modelling. *Applied Animal Behaviour Science*, *87*, 31–44. doi: https://doi.org/10.1016/j.applanim.2003.12.005.

Bracke, M. B. M., Boumans, I. J. M. M., Nijland, H. J., & Bokkers, E. A. M. (2023). Review: Connecting circularity to animal welfare calls for a 'novel' conceptual framework based on integrity. *animal*, *17*, 100694. doi: 10.1016/j.animal.2022.100694.

Bracke, M. B. M., Metz, J. H. M., & Spruijt, B. M. (1999b). Overall animal welfare reviewed. Part 2: Assessment tables and schemes. *Netherlands Journal of Agricultural Science*, (pp. 293–305). doi: 10.18174/njas.v47i3.467.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32. doi: 10.1023/A:1010933404324.

Browning, H. (2022). Assessing measures of animal welfare. *Biology & Philosophy*, *37*, 36.

Budaev, S., Cusimano, G. M., & Rønnestad, I. (2025). FishMet: A digital twin framework for appetite, feeding decisions and growth in salmonid fish. *Aquaculture, Fish and Fisheries*, *5*, e70064. doi: 10.1002/aff2.70064. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aff2.70064.

Bui, S., Dalvin, S., Vågseth, T., Oppedal, F., Fossøy, F., Brandsegg, H., Jacobsen, Á., á Norði, G., Fordyce, M. J., Michelsen, H. K. et al. (2021). Finding the needle in the haystack: Comparison of methods for salmon louse enumeration in plankton samples. *Aquaculture Research*, *52*, 3591–3604. doi:

https://doi.org/10.1111/are.15202.

Burke, M., Nikolic, D., Fabry, P., Rishi, H., Telfer, T., & Rey Planellas, S. (2025). Precision farming in aquaculture: non-invasive monitoring of Atlantic salmon (Salmo salar) behaviour in response to environmental conditions in commercial sea cages for health and welfare assessment. *Frontiers in Robotics and AI*, *12*. doi: 10.3389/frobt.2025.1574161.

Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving into high quality object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6154–6162). doi: 10.1109/CVPR.2018.00644.

Cantonjos, N. A., & Biswas, A. (2025). AgroAskAI: A multi-agentic AI framework for supporting smallholder farmers' enquiries globally. doi: 10.48550/arXiv.2512.14910 arXiv:2512.14910 [cs.AI] version: 1.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213–229). Springer. doi: 10.1007/978-3-030-58452-8_13.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. doi: 10.48550/arXiv.2104.14294 arXiv:2104.14294 [cs.CV].

Cemri, M. et al. (2025). Why do multi-agent LLM systems fail? doi: 10.48550/arXiv.2503.13657 arXiv:2503.13657 [cs.AI] version: 3.

Cerqueira, V., Pimentel, J., Korus, J., Bravo, F., Amorim, J., Oliveira, M., Swanson, A., Filgueira, R., Grant, J., & Torgo, L. (2024). Forecasting ocean hypoxia in salmonid fish farms. *Frontiers in Aquaculture*, *3*. doi: 10.3389/faquc.2024.1365123.

Channing, C., Hughes, B., & Walker, A. (2001). Spatial distribution and behaviour of laying hens housed in an alternative system. *Applied Animal Behaviour Science*, *72*, 335–345. doi: https://doi.org/10.1016/S0168-1591(00)00206-9.

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, Z. (2024). BGE M3-Embedding: Multilingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv:2402.03216.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* KDD '16 (p. 785–794). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2939672.2939785.

Cheng, N., Yan, Z., Wang, Z., Li, Z., Yu, J., Zheng, Z., Tu, K., Xu, J., & Han, W. (2024). Potential and limitations of LLMs in capturing structured semantics: A case study on srl. In *International Conference on Intelligent Computing* (pp. 50–61). Springer. doi: https://doi.org/10.1007/978-981-97-5663-6_5.

Chesterman, S. (2025). Good models borrow, great models steal: intellectual property rights and generative AI. *Policy and Society*, *44*, 23–37. doi: 10.1093/polsoc/puae006.

Chowdhary, K., & Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*, (pp. 603–649). doi: https://doi.org/10.1007/978-81-322-3972-7_19.

Ángel Cobo, Llorente, I., Luna, L., & Luna, M. (2019). A decision support system for fish farming using particle swarm optimization. *Computers and Electronics in Agriculture*, *161*, 121–130. doi: https://doi.org/10.1016/j.compag.2018.03.036. BigData and DSS in Agriculture.

Collins, L. M., & Part, C. E. (2013). Modelling farm animal welfare. *Animals*, *3*, 416–441. doi: https://doi.org/10.3390/ani3020416.

Cormack, G. V., Clarke, C. L. A., & Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp. 758–759). Boston MA USA: ACM. doi: 10.1145/1571941.1572114.

Costello, M. J. (2006). Ecology of sea lice parasitic on farmed and wild fish. *Trends in Parasitology*, *22*, 475–483. doi: https://doi.org/10.1016/j.pt.2006.08.006.

Costello, M. J. (2009). How sea lice from salmon farms may cause wild salmonid declines in Europe and North America and be a threat to fishes elsewhere. *Proceedings of the Royal Society B: Biological Sciences*, *276*, 3385–3394. doi: https://doi.org/10.1098/rspb.2009.0771.

Costelloe, M., Costelloe, J., O'donohoe, G., Coghlan, N., Oonk, M., & Der Heijden, V. (1998). Planktonic distribution of sea lice larvae, lepeophtheirus salmonis, in killary harbour, west coast of ireland. *Journal of the Marine Biological Association of the United Kingdom*, *78*, 853–874. doi: 10.1017/S0025315400044830.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society:*

Series B (Methodological), *20*, 215–232. doi: https://doi.org/10.1111/j.2517-6161.1958.tb00292.x.

Cui, M., Liu, X., Liu, H., Zhao, J., Li, D., & Wang, W. (2025). Fish tracking, counting, and behaviour analysis in digital aquaculture: A comprehensive survey. *Reviews in Aquaculture*, *17*, e13001. doi: 10.1111/raq.13001. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/raq.13001.

da Silva, M. V., Silva, L. H., Junior, J. D. D., Escarpinati, M. C., Backes, A. R., & Mari, J. F. (2023). Generating synthetic multispectral images using neural style transfer: A study with application in channel alignment. *Computers and Electronics in Agriculture*, *206*, 107668. doi: https://doi.org/10.1016/j.compag.2023.107668.

Day, J. E., Van de Weerd, H. A., & Edwards, S. A. (2008). The effect of varying lengths of straw bedding on the behaviour of growing pigs. *Applied Animal Behaviour Science*, *109*, 249–260. doi: https://doi.org/10.1016/j.applanim.2007.02.006.

De Mol, R. M., Schouten, W., Evers, E., Drost, H., Houwers, H., & Smits, A. (2006). A computer model for welfare assessment of poultry production systems for laying hens. *NJAS-Wageningen Journal of Life Sciences*, *54*, 157–168. doi: https://doi.org/10.1016/S1573-5214(06)80019-9.

DeepSeek-AI et al. (2025). DeepSeek-V3.2: Pushing the frontier of open large language models. doi: 10.48550/arXiv.2512.02556 arXiv:2512.02556 [cs].

Delfosse, C., Pageat, P., Lafont-Lecuelle, C., Asproni, P., Chabaud, C., Cozzi, A., & Bienboire-Frosini, C. (2021). Effect of handling and crowding on the susceptibility of Atlantic salmon (Salmo salar L.) to Lepeophtheirus salmonis (Krøyer) copepodids. *Journal of Fish Diseases*, *44*, 327–336. doi: 10.1111/jfd.13286.

Demir-Kavuk, O., Kamada, M., Akutsu, T., & Knapp, E.-W. (2011). Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features. *BMC bioinformatics*, *12*, 1–10. doi: https://doi.org/10.1186/1471-2105-12-412.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). doi: 10.1109/CVPR.2009.5206848 iSSN: 1063-6919.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). doi: https://doi.org/10.18653/V1/N19-1423.

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). Repvgg: Making vgg-style convnets great again. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13728–13737). doi: 10.1109/CVPR46437.2021.01352.

Diwan, T., Anirudh, G., & Tembhurne, J. V. (2023). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications*, *82*, 9243–9275. doi: 10.1007/s11042-022-13644-y.

Domínguez-May, R., Hernández, J. M., & Velázquez-Abunader, I. (2024). A review of dynamic optimization in aquaculture production economics. *Reviews in Aquaculture*, *16*, 1696–1717. doi: 10.1111/raq.12918. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/raq.12918.

Dvornik, N., Mairal, J., & Schmid, C. (2019). On the importance of visual context for data augmentation in scene understanding. doi: 10.48550/arXiv.1809.02492 arXiv:1809.02492 [cs.CV].

Dyrstad, J. S., & øye, E. R. (2025). Creating synthetic data sets for training of neural networks for automatic catch analysis in fisheries. *Computers and Electronics in Agriculture*, *233*, 110160. doi: https://doi.org/10.1016/j.compag.2025.110160.

EFSA (2009). Scientific report on the effects of farming systems on dairy cow welfare and disease. *EFSA Journal*, *7*, 1143r. doi: https://doi.org/10.2903/j.efsa.2009.1143r.

Fabian, J., Laven, R. A., & Whay, H. R. (2014). The prevalence of lameness on New Zealand dairy farms: A comparison of farmer estimate and locomotion scoring. *The Veterinary Journal*, *201*, 31–38. doi: 10.1016/j.tvjl.2014.05.011.

Fang, F., Bai, Y., Ni, S., Yang, M., Chen, X., & Xu, R. (2024). Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. doi: 10.48550/arXiv.2405.20978 arXiv:2405.20978 [cs.AI].

FAO (2022). *Blue Transformation - Roadmap 2022–2030. A vision for FAO' s work on aquatic food systems*. Rome, Italy: FAO. doi: 10.4060/cc0459en.

FAO (2024). *The State of World Fisheries and Aquaculture 2024. Blue Transformation in action*. Number 2024 in The State of World Fisheries and Aquaculture (SOFIA). Rome, Italy: FAO. doi: 10.4060/cd0683en.

FAO (2025). *Guidelines for Sustainable Aquaculture*. Rome, Italy: FAO. doi: 10.4060/cd3785en.

FAO (2026). *Translating the Guidelines for Sustainable Aquaculture into action. Policy brief*. Technical Report, FAO Rome, Italy. URL: https://openknowledge.fao.org/handle/20.500.14283/cd8563en. doi: 10.4060/cd8563en num Pages: 8.

Feng, Z., Xue, R., Yuan, L., Yu, Y., Ding, N., Liu, M., Gao, B., Sun, J., Zheng, X., & Wang, G. (2026). Multi-agent embodied AI: advances and future directions. *Science China Information Sciences*, *69*, 151202. doi: 10.1007/s11432-025-4820-4.

Fernandez-Gonzalez, V., Ulvan, E., Sanchez-Jerez, P., Diserud, O., Toledo-Guedes, K., Casado-Coy, N., Klebert, P., & Uglem, I. (2022). Abundance of sea lice larvae in plankton samples: determination of optimal sample sizes. *Aquaculture*, *551*, 737919. doi: https://doi.org/10.1016/j.aquaculture.2022.737919.

Fiskeridirektoratet (). Akvakulturstatistikk: Laks, regnbueørret og ørret (offisiell statistikk). https://www.fiskeridir.no/Akvakultur/Tall-og-analyse/Akvakulturstatistikk-tidsserier/Laks-regnbueoerret-og-oerret/salg-av-laks-og-regnbueorret. Accessed: April 13, 2025.

Fiskeridirektoratet (2023). Akvakulturstatistikk: Matfiskproduksjon av laks, regnbueørret og ørret.

Fiskeridirektoratet (2023). Aquaculture statistics: Farmed fish production of salmon, rainbow trout and trout [original title in norwegian: Akvakulturstatistikk: Matfiskproduksjon av laks, regnbueørret og ørret]. https://www.fiskeridir.no/Akvakultur/Tall-og-analyse/Akvakulturstatistikk-tidsserier/Laks-regnbueoerret-og-oerret/Matfiskproduksjon. Accessed: 2025-04-03.

Fitzgerald, A., Ioannou, C. C., Consuegra, S., Dowsey, A., & Garcia de Leaniz, C. (2025). Machine vision applications for welfare monitoring in aquaculture: Challenges and opportunities. *Aquaculture, Fish and Fisheries*, *5*, e70036. doi: 10.1002/aff2.70036. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aff2.70036.

Food and Agriculture Organization of the United Nations (FAO) (2025). *GLOBEFISH: Quarterly Salmon Analysis - February 2025*. Technical Report, Food and Agriculture Organization of the United Nations. URL: https://openknowledge.fao.org/handle/20.500.14283/cd4468en.

Foris, B., Sheng, K., Dürnberger, C., Oczak, M., & Rault, J.-L. (2025). AI for one welfare: the role of animal welfare scientists in developing valid and ethical AI-based welfare assessment tools. *Frontiers in Veterinary Science*, *12*. doi: 10.3389/fvets.2025.1645901.

Franek, L., & Jiang, X. (2013). Orthogonal design of experiments for parameter learning in image segmentation. *Signal Processing*, *93*, 1694–1704. doi: https://doi.org/10.1016/j.sigpro.2012.08.016. Special issue on Machine Learning in Intelligent Image Processing.

Fraser, D. (2003). Assessing animal welfare at the farm and group level: the interplay of science and values. *Animal welfare*, *12*, 433–443. doi: https://doi.org/10.1017/S0962728600026038.

Fraser, D., Weary, D. M., Pajor, E. A., & Milligan, B. N. (1997). A scientific conception of animal welfare that reflects ethical concerns. *Animal Welfare*, *6*, 187–205. doi: 10.1017/S0962728600019795.

Føre, M. et al. (2018). Precision fish farming: A new framework to improve production in aquaculture. *Biosystems Engineering*, *173*, 176–193. doi: 10.1016/j.biosystemseng.2017.10.014.

Gao, J., Pham, Q. H., Varghese, S., Saurav, S., & Hoskere, V. (2025). RAGalyst: Automated human-aligned agentic evaluation for domain-specific RAG. arXiv:2511.04502.

Garcia, R., Prados, R., Quintana, J., Tempelaar, A., Gracias, N., Rosen, S., Vågstøl, H., & Løvall, K. (2020). Automatic segmentation of fish using deep learning with application to fish size measurement. *ICES Journal of Marine Science*, *77*, 1354–1366. doi: 10.1093/icesjms/fsz186.

Garlock, T., Asche, F., Anderson, J., Ceballos-Concha, A., Love, D. C., Osmundsen, T. C., & Pincinato, R. B. M. (2022). Aquaculture: The missing contributor in the food security agenda. *Global Food Security*, *32*, 100620. doi: 10.1016/j.gfs.2022.100620.

Garlock, T. M. et al. (2024). Environmental, economic, and social sustainability in aquaculture: the aquaculture performance indicators. *Nature Communications*, *15*, 5274. doi: 10.1038/s41467-024-49556-8.

Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). YOLOX: Exceeding YOLO series in 2021. arXiv:2107.08430.

Geiping, J., Goldblum, M., Somepalli, G., Shwartz-Ziv, R., Goldstein, T., & Wilson, A. G. (2023). How much data are augmentations worth? an investigation into scaling laws, invariance, and implicit regularization. arXiv:2210.06441.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, *2*, 665–673. doi: 10.1038/s42256-020-00257-z.

Georgopoulou, D. G., Stavrakidis-Zachou, O., Mitrizakis, N., & Papandroulakis, N. (2021). Tracking and analysis of the movement behavior of European seabass (Dicentrarchus labrax) in aquaculture systems. *Frontiers in Animal Science*, *2*. doi: 10.3389/fanim.2021.754520.

Gephart, J. A. et al. (2021). Environmental performance of blue foods. *Nature*, *597*, 360–365. doi: 10.1038/s41586-021-03889-2.

Ghadimi, A., & Beigy, H. (2022). Hybrid multi-document summarization using pre-trained language models. *Expert Systems with Applications*, *192*, 116292. doi: https://doi.org/10.1016/j.eswa.2021.116292.

Girshick, R. (2015). Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1440–1448). doi: 10.1109/ICCV.2015.169.

Gjerde, B., & Aslam, M. L. (2025). Prediction of the salmon lice density on wild sea trout from the mean predicted lice density in the sea: A cross-validation of the data in myksvoll et al. (2018). *Aquaculture*, *599*, 742149. doi: https://doi.org/10.1016/j.aquaculture.2025.742149.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples.

Grattafiori, A. et al. (2024). The Llama 3 herd of models. arXiv:2407.21783.

Guo, D., Zhu, Q., Yang, D., Xie, Z., Dong, K., Zhang, W., Chen, G., Bi, X., Wu, Y., Li, Y. et al. (2024). DeepSeek-Coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, .

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). Knn model-based approach in classification. In R. Meersman, Z. Tari, & D. C. Schmidt (Eds.), *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE* (pp. 986–996). Berlin, Heidelberg: Springer Berlin Heidelberg.

Gupta, A., Bringsdal, E., Knausgård, K. M., & Goodwin, M. (2022). Accurate wound and lice detection in atlantic salmon fish using a convolutional neural network. *Fishes*, *7*, 345. doi: 10.3390/fishes7060345.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, .

Hall, Z., Subbiah, M., Zollo, T. P., McKeown, K., & Zemel, R. (2026). Guiding LLM decision-making with fairness reward models. In *Proceedings of the 39th International Conference on Neural Information Processing Systems* NIPS '25. Red Hook, NY, USA: Curran Associates Inc. doi: 10.52202/085713-4867.

Hamilton, J. D. (2020). *Time Series Analysis*. Princeton University Press. Google-Books-ID: BeryDwAAQBAJ.

Hamre, L. A., Eichner, C., Caipang, C. M. A., Dalvin, S. T., Bron, J. E., Nilsen, F., Boxshall, G., & Skern-Mauritzen, R. (2013). The salmon louse Lepeophtheirus salmonis (copepoda: Caligidae) life cycle has only two chalimu stages. *PLOS ONE*, *8*, e73539. doi: 10.1371/journal.pone.0073539.

Han, K., Jang, J., Kim, H., Jeong, G., & Kim, H. (2025). Exploring the impact of instruction-tuning on LLM's susceptibility to misinformation. doi: 10.48550/arXiv.2507.18203 arXiv:2507.18203 [cs].

Hansen, L. P., & Windsor, M. L. (2006). Interactions between aquaculture and wild stocks of Atlantic salmon and other diadromous fish species: Science and management, challenges and solutions: An introduction by the conveners. *ICES Journal of Marine Science*, *63*, 1159–1161. doi: 10.1016/j.icesjms.2006.05.003.

Havforskningsinstituttet (). Lakseluskartet. https://www.hi.no/forskning/marine-data-forskningsdata/lakseluskart/html/lakseluskart.html. Accessed: April 13, 2025.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum Contrast for Unsupervised Visual Representation Learning . In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 9726–9735). Los Alamitos, CA, USA: IEEE Computer Society. doi: 10.1109/CVPR42600.2020.00975.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). doi: 10.1109/CVPR.2016.90 iSSN: 1063-6919.

He, Z., Jiang, Y., Wang, X., Xie, Y., Cheng, Y., & Mei, J. (2026). Machine learning for extracting morphological phenotypic traits and estimating weight in largemouth bass. *Aquaculture and Fisheries*, *11*, 323–332. doi: 10.1016/j.aaf.2025.09.004.

Hendrycks, D. et al. (2021). The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 8340–8349). Montreal, QC, Canada.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*, 261–266. doi: 10.1126/science.aaa8685. arXiv:https://www.science.org/doi/pdf/10.1126/science.aaa8685.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of*

Statistics, *36*, 1171 – 1220. doi: https://doi.org/10.1214/009053607000000677.

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 1314–1324). doi: 10.1109/ICCV.2019.00140.

Hridoy, M. A. A. M., Bordin, C., Masood, A., & Masood, K. (2025). Predictive modelling of aquaculture water quality using IoT and advanced machine learning algorithms. *Results in Chemistry*, *16*, 102456. doi: 10.1016/j.rechem.2025.102456.

Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., & Pfister, T. (2023). Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, .

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2261–2269). Los Alamitos, CA, USA: IEEE Computer Society. doi: 10.1109/CVPR.2017.243.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, *43*, 42:1–42:55. doi: 10.1145/3703155.

Huntingford, F. A., Adams, C., Braithwaite, V. A., Kadri, S., Pottinger, T. G., Sandøe, P., & Turnbull, J. F. (2006). Current issues in fish welfare. *Journal of Fish Biology*, *68*, 332–372. doi: 10.1111/j.0022-1112.2006.001046.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0022-1112.2006.001046.x.

Igboeli, O. O., Burka, J. F., & Fast, M. D. (2014). Lepeophtheirus salmonis: a persisting challenge for salmon aquaculture. *Animal Frontiers*, *4*, 22–32. doi: 10.2527/af.2014-0004.

Inc., S. (2019). Streamlit. https://github.com/streamlit/streamlit. Accessed: 2025-07-07.

Ingenbleek, P. T., Immink, V. M., Spoolder, H. A., Bokma, M. H., & Keeling, L. J. (2012). EU animal welfare policy: Developing a comprehensive policy framework. *Food Policy*, *37*, 690–699. doi: https://doi.org/10.1016/j.foodpol.2012.07.001.

Iversen, A., Asche, F., Hermansen, Ø., & Nystøyl, R. (2020). Production cost and competitiveness in major salmon farming countries 2003–2018. *Aquaculture*, *522*, 735089. doi: https://doi.org/10.1016/j.aquaculture.2020.735089.

Jeong, J., Arriagada, G., & Revie, C. W. (2023). Targets and measures: Challenges associated with reporting low sea lice levels on Atlantic salmon farms. *Aquaculture*, *563*, 738865. doi: 10.1016/j.aquaculture.2022.738865.

Jeong, J., & Revie, C. W. (2020). Appropriate sampling strategies to estimate sea lice prevalence on salmon farms with low infestation levels. *Aquaculture*, *518*, 734858. doi: https://doi.org/10.1016/j.aquaculture.2019.734858.

Jeong, J., Stormoen, M., McEwan, G. F., Thakur, K. K., & Revie, C. W. (2021). Salmon lice should be managed before they attach to salmon: Exploring epidemiological factors affecting *Lepeophtheirus salmonis* abundance on salmon farms. *Aquaculture*, *541*, 736792. doi: 10.1016/j.aquaculture.2021.736792.

Ji, Z., Yu, T., Xu, Y., Lee, N., Ishii, E., & Fung, P. (2023). Towards mitigating hallucination in large language models via self-reflection. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023* (pp. 1827–1843). Singapore: Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.123.

Jiang, J.-P., Liu, S.-Y., Cai, H.-R., Zhou, Q.-L., & Ye, H.-J. (2026). Representation learning for tabular data: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *48*, 6488–6508. doi: 10.1109/TPAMI.2026.3657217.

Jonsson, B., Jonsson, N., & Hansen, L. P. (2003). Atlantic salmon straying from the river imsa. *Journal of Fish Biology*, *62*, 641–657. doi: https://doi.org/10.1046/j.1095-8649.2003.00053.x.

Joseph, K. J., Khan, S., Khan, F. S., & Balasubramanian, V. N. (2021). Towards open world object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5826–5836). doi: 10.1109/CVPR46437.2021.00577.

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*, 531–538. doi: 10.1002/sam.11583.

Jung, Y., Byun, S., Kim, B., Ul Amin, S., & Seo, S. (2024). Harnessing synthetic data for enhanced detection of pine wilt disease: An image classification approach. *Computers and Electronics in Agriculture*, *218*, 108690. doi: https://doi.org/10.1016/j.compag.2024.108690.

Kamalloo, E., Thakur, N., Lassance, C., Ma, X., Yang, J.-H., & Lin, J. (2024). Resources for brewing BEIR: Reproducible reference models and an official leaderboard. In *Proceedings of the 47th International*

ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1431–1440). Washington DC USA: ACM. doi: 10.1145/3626772.3657862.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 6769–6781). Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550.

Kashiha, M., Pluk, A., Bahr, C., Vranken, E., & Berckmans, D. (2013). Development of an early warning system for a broiler house using computer vision. *Biosystems Engineering*, *116*, 36–45. doi: 10.1016/j.biosystemseng.2013.06.004.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. volume 30.

Keeling, L., Tunón, H., Olmos Antillón, G., Berg, C., Jones, M., Stuardo, L., Swanson, J., Wallenbeck, A., Winckler, C., & Blokhuis, H. (2019). Animal welfare and the united nations sustainable development goals. *Frontiers in Veterinary Science*, *6*. doi: 10.3389/fvets.2019.00336.

Khanam, R., & Hussain, M. (2024). YOLOv11: An overview of the key architectural enhancements. arXiv:2410.17725.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 3992–4003). doi: 10.1109/ICCV51070.2023.00371.

Klimesova, B., Lyashevska, O., Ruane, N. M., D' Arcy, J., Talbot, A., Rodger, H., & O' Dwyer, K. (2025). Effects of temporal, geographical and environmental factors on salmon lice (Lepeophtheirus salmonis) levels of Atlantic salmon (Salmo salar) in Ireland. *Scientific Reports*, *15*, 34614. doi: 10.1038/s41598-025-18210-8.

Kollar, T., Laskey, M., Stone, K., Thananjeyan, B., & Tjersland, M. (2021). Simnet: Enabling robust unknown object manipulation from pure synthetic data via stereo. doi: 10.48550/arXiv.2106.16118 arXiv:2106.16118 [cs.RO].

Konovalov, D. A., Saleh, A., Efremova, D. B., Domingos, J. A., & Jerry, D. R. (2019). Automatic weight estimation of harvested fish from images. In *2019 Digital Image Computing: Techniques and Applications (DICTA)* (pp. 1–7). doi: 10.1109/DICTA47822.2019.8945971.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*, 150. doi: 10.3390/info10040150.

Kozhipuram, A. V., Shailendra, S., & Kadel, R. (2025). Retrieval-augmented generation vs. baseline LLMs: A multi-metric evaluation for knowledge-intensive content. *Information*, *16*. doi: 10.3390/info16090766.

Kragesteen, T. J., Simonsen, K., Visser, A. W., & Andersen, K. H. (2018). Identifying salmon lice transmission characteristics between Faroese salmon farms. *Aquaculture Environment Interactions*, *10*, 49–60. doi: https://doi.org/10.3354/aei00252.

Krishnan, N. (2025). Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications. doi: 10.48550/arXiv.2504.21030 arXiv:2504.21030 [cs.MA].

Kristmundsson, J., Patursson, Ø., Potter, J., & Xin, Q. (2023). Fish monitoring in aquaculture using multibeam echosounders and machine learning. *IEEE Access*, *11*, 108306–108316. doi: 10.1109/ACCESS.2023.3320949.

Krolicka, A., Nilsen, M. M., Hansen, B. K., Jacobsen, M. W., Provan, F., & Baussant, T. (2022). Sea lice (Lepeophtherius salmonis) detection and quantification around aquaculture installations using environmental DNA. *PLOS ONE*, *17*, e0274736. doi: 10.1371/journal.pone.0274736.

Lamping, C., Derks, M., Groot Koerkamp, P., & Kootstra, G. (2022). ChickenNet - an end-to-end approach for plumage condition assessment of laying hens in commercial farms using computer vision. *Comput. Electron. Agric.*, *194*. doi: 10.1016/j.compag.2022.106695.

Larionov, D., Takeshita, S., Zhang, R., Chen, Y., Leiter, C., Wang, Z., Greisinger, C., & Eger, S. (2025). DeepSeek vs. o3-mini: How well can reasoning LLMs evaluate mt and summarization? *arXiv preprint arXiv:2504.08120*, .

Laschinger, J., Linnenkohl, S., Fuerst-Waltl, B., & Kofler, J. (2024). Lameness prevalence in lactating and dry cows - expert's, student's and farmers' assessments. *Tierarztliche Praxis. Ausgabe G, Grosstiere/Nutztiere*, *52*, 192–202. doi: 10.1055/a-2335-8275.

Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *Ieee Access*, *8*, 193907–193934. doi: https://doi.org/10.1109/ACCESS.2020.3031549.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. doi: 10.1038/nature14539.

Lerøy Seafood Group (2022). *Policy: Control of Salmon Lice*. Policy Document, Lerøy Seafood Group. URL: https://www.leroyseafood.com/en/sustainability/sustainability-library-2022/policies/policy-control-of-salmon-lice/ accessed: 2025-07-02.

Li, M., Wu, J., Wang, X., Chen, C., Qin, J., Xiao, X., Wang, R., Zheng, M., & Pan, X. (2023). Aligndet: Aligning pre-training and fine-tuning in object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6866–6876). Paris, France: IEEE. doi: https://doi.org/10.1109/ICCV51070.2023.00632.

Liao, Z., & Couillet, R. (2019). A large dimensional analysis of least squares support vector machines. *IEEE Transactions on Signal Processing*, *67*, 1065–1074. doi: https://doi.org/10.1109/TSP.2018.2889954.

Lima, J. A. d. O. (2025). Poly-vector retrieval: Reference and content embeddings for legal documents. doi: 10.48550/arXiv.2504.10508 arXiv:2504.10508 [cs].

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017a). Feature Pyramid Networks for Object Detection . In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 936–944). Los Alamitos, CA, USA: IEEE Computer Society. doi: 10.1109/CVPR.2017.106.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017b). Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 2999–3007). doi: 10.1109/ICCV.2017.324 iSSN: 2380-7504.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13* (pp. 740–755). Cham: Springer. doi: 10.1007/978-3-319-10602-1_48.

Liu, A., Feng, B., Wang, B., Wang, B., Liu, B., Zhao, C., Dengr, C., Ruan, C., Dai, D., Guo, D. et al. (2024). DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, .

Liu, M., Zhang, C., & Lin, C. (2025a). GAB-YOLO: a lightweight deep learning model for real-time detection of abnormal behaviors in juvenile greater amberjack fish. *Frontiers in Marine Science*, *12*. doi: 10.3389/fmars.2025.1574580.

Liu, Q., Wang, W., & Willard, J. (2025b). Effects of prompt length on domain-specific tasks for large language models. *arXiv preprint arXiv:2502.14255*, .

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21–37). Springer. doi: 10.1007/978-3-319-46448-0_2.

Liu, X., Wu, C., Menta, M., Herranz, L., Raducanu, B., Bagdanov, A. D., Jui, S., & van de Weijer, J. (2020). Generative feature replay for class-incremental learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 915–924). doi: 10.1109/CVPRW50498.2020.00121.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, .

Lu, H., Li, Y., Zhang, Y., Chen, M., Serikawa, S., & Kim, H. (2017). Underwater optical image processing: a comprehensive review. doi: 10.48550/arXiv.1702.03600 arXiv:1702.03600 [cs.CV].

Luo, J. et al. (2025a). Large language model agent: A survey on methodology, applications and challenges. doi: 10.48550/arXiv.2503.21460 arXiv:2503.21460 [cs].

Luo, Z., Chen, R., Li, S., Zheng, J., & Guo, J. (2025b). FishMambaNet: A Mamba-Based vision model for detecting fish diseases in aquaculture. *Fishes*, *10*, 649. doi: 10.3390/fishes10120649.

Luthman, O., Jonell, M., & Troell, M. (2019). Governing the salmon farming industry: Comparison between national regulations and the ASC salmon standard. *Marine Policy*, *106*, 103534. doi: 10.1016/j.marpol.2019.103534.

kumar M, P., K, S. K., P, K., & C, S. (2025). Fish disease detection in aquaculture using pseudo hamiltonian neural network optimized with Philippine eagle optimization algorithm. *Knowledge-Based Systems*, *317*, 113374. doi: 10.1016/j.knosys.2025.113374.

Ma, H., Zhang, C., Bian, Y., Liu, L., Zhang, Z., Zhao, P., Zhang, S., Fu, H., Hu, Q., & Wu, B. (2023). Fairness-guided few-shot prompting for large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in Neural Information Processing Systems* (pp. 43136–43155). Curran Associates, Inc. volume 36.

Mandikal, P., & Mooney, R. (2024). Sparse meets dense: A hybrid approach to enhance scientific document retrieval. doi: 10.48550/arXiv.2401.04055 arXiv:2401.04055 [cs].

Marchant-Forde, J. N. (2015). The science of animal behavior and welfare: Challenges, opportunities, and global perspective. *Frontiers in Veterinary Science*, *2*, 16. doi: 10.3389/fvets.2015.00016.

Maron, M. E. (1961). Automatic indexing: An experimental inquiry. *J. ACM*, *8*, 404–417. doi: 10.1145/321075.321084.

Mascarenhas, S., & Agarwal, M. (2021). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for image classification. In *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)* (pp. 96–99). IEEE volume 1. doi: 10.1109/CENTCON52345.2021.9687944.

Mathisen, B. M., Bach, K., & Aamodt, A. (2021). Using extended siamese networks to provide decision support in aquaculture operations. *Applied Intelligence*, *51*, 8107–8118. doi: 10.1007/s10489-021-02251-3.

Mathisen, B. M., Haro, P., Hanssen, B., Björk, S., & Walderhaug, S. (2016). Decision support systems in fisheries and aquaculture: A systematic review. doi: 10.48550/arXiv.1611.08374 arXiv:1611.08374 [cs].

McBeath, A. J., Penston, M. J., Snow, M., Cook, P. F., Bricknell, I. R., & Cunningham, C. O. (2006). Development and application of real-time PCR for specific detection of Lepeophtheirus salmonis and Caligus elongatus larvae in Scottish plankton samples. *Diseases of aquatic organisms*, *73*, 141–150. doi: https://doi.org/10.3354/dao073141.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133. doi: 10.1007/BF02478259.

Mellor, D. J., Beausoleil, N. J., Littlewood, K. E., McLean, A. N., McGreevy, P. D., Jones, B., & Wilkins, C. (2020). The 2020 five domains model: Including human–animal interactions in assessments of animal welfare. *Animals*, *10*, 1870. doi: https://doi.org/10.3390/ani10101870.

Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., & Wang, J. (2021). Conditional DETR for fast training convergence. arXiv:2108.06152.

Michel, P. (2021). Learning neural models for natural language processing in the face of distributional shift. *arXiv preprint arXiv:2109.01558*, .

Michielon, A. et al. (2024). Mind the step: An artificial intelligence-based monitoring platform for animal welfare. *Sensors*, *24*, 8042. doi: 10.3390/s24248042.

Microsoft (2020). Visual object tagging tool (VoTT). https://github.com/microsoft/VoTT. Version 2.2.0, Accessed: 2024-07-22.

Mishra, S., Panda, R., Phoo, C. P., Chen, C.-F. R., Karlinsky, L., Saenko, K., Saligrama, V., & Feris, R. S. (2022). Task2sim: Towards effective pre-training and transfer from synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 9194–9204). New Orleans, LA, USA. doi: https://doi.org/10.1109/CVPR52688.2022.00898.

Misund, A., Thorvaldsen, T., Strand, A. V., Oftebro, T. L., & Dahle, S. W. (2024). Opportunities and challenges in new production systems for salmon farming in Norway—Industry perspective. *Marine Policy*, *170*, 106394. doi: 10.1016/j.marpol.2024.106394.

Mohamadi, H., Keyvanrad, M. A., & Mohammadi, M. R. (2025). Feature based methods in domain adaptation for object detection: A review paper. arXiv:2412.17325.

Moriarty, M., Ives, S. C., Murphy, J. M., & Murray, A. G. (2023). Modelling parasite impacts of aquaculture on wild fish: The case of the salmon louse (lepeophtheirus salmonis) on out-migrating wild Atlantic salmon (salmo salar) smolt. *Preventive Veterinary Medicine*, *214*, 105888. doi: 10.1016/j.prevetmed.2023.105888.

Morton, A., Routledge, R., Peet, C., & Ladwig, A. (2004). Sea lice (Lepeophtheirus salmonis) infection rates on juvenile pink (Oncorhynchus gorbuscha) and chum (Oncorhynchus keta) salmon in the nearshore marine environment of British Columbia, Canada. *Canadian Journal of Fisheries and Aquatic Sciences*, *61*, 147–157. doi: https://doi.org/10.1139/f04-016.

Murray, A. G., & Moriarty, M. (2021). A simple modelling tool for assessing interaction with host and local infestation of sea lice from salmonid farms on wild salmonids based on processes operating at multiple scales in space and time. *Ecological Modelling*, *443*, 109459. doi: https://doi.org/10.1016/j.ecolmodel.2021.109459.

Mustafa, A., Rankaduwa, W., & Campbell, P. (2001). Estimating the cost of sea lice to salmon aquaculture in eastern canada. *The Canadian veterinary journal*, *42*, 54.

Namazifar, M., Papangelis, A., Tur, G., & Hakkani-Tür, D. (2021). Language model is all you need: Natural language understanding as question answering. In *ICASSP 2021-2021 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7803–7807). IEEE. doi: https://doi.org/10.1109/ICASSP39728.2021.9413810.

Naylor, R. L., Hardy, R. W., Buschmann, A. H., Bush, S. R., Cao, L., Klinger, D. H., Little, D. C., Lubchenco, J., Shumway, S. E., & Troell, M. (2021). A 20-year retrospective review of global aquaculture. *Nature*, *591*, 551–563. doi: 10.1038/s41586-021-03308-6.

Neik, T. X., Dolatabadian, A., Danilevicz, M. F., Upadhyaya, S. R., Zhang, F., Batley, J., & Edwards, D. (2025). Plant disease epidemiology in the age of artificial intelligence and machine learning. *Agriculture Communications*, *3*, 100089. doi: 10.1016/j.agrcom.2025.100089.

Nelson, E., Robinson, S., Feindel, N., Sterling, A., Byrne, A., & Pee Ang, K. (2018). Horizontal and vertical distribution of sea lice larvae (Lepeophtheirus salmonis) in and around salmon farms in the Bay of Fundy, Canada. *Journal of fish diseases*, *41*, 885–899. doi: https://doi.org/10.1111/jfd.12692.

Nilsen, A., Nielsen, K. V., & Bergheim, A. (2020). A closer look at closed cages: Growth and mortality rates during production of post-smolt Atlantic salmon in marine closed confinement systems. *Aquacultural Engineering*, *91*, 102124. doi: 10.1016/j.aquaeng.2020.102124.

Niu, J. et al. (2025). Mineru2.5: A decoupled vision-language model for efficient high-resolution document parsing. arXiv:2509.22186.

Noble, C. et al. (2018). *Welfare Indicators for farmed Atlantic salmon: tools for assessing fish welfare*. Nofima. Accepted: 2018-12-21T01:00:34Z.

Nogueira, R., & Cho, K. (2019). Passage Re-ranking with BERT. doi: 10.48550/ARXIV.1901.04085 version Number: 5.

á Norði, G., Simonsen, K., Danielsen, E., Eliasen, K., Mols-Mortensen, A., Christiansen, D., Steingrund, P., Galbraith, M., & Patursson, Ø. (2015). Abundance and distribution of planktonic lepeophtheirus salmonis and caligus elongatus in a fish farming region in the faroe islands. *Aquaculture Environment Interactions*, *7*, 15–27. doi: 10.3354/aei00134.

Olaussen, J. O. (2018). Environmental problems and regulation in the aquaculture industry. insights from norway. *Marine Policy*, *98*, 158–163. doi: https://doi.org/10.1016/j.marpol.2018.08.005.

Oppedal, F., Dempster, T., & Stien, L. H. (2011). Environmental drivers of Atlantic salmon behaviour in sea-cages: A review. *Aquaculture*, *311*, 1–18. doi: 10.1016/j.aquaculture.2010.11.020.

Osco, L. P., Wu, Q., de Lemos, E. L., Gonçalves, W. N., Ramos, A. P. M., Li, J., & Marcato, J. (2023). The segment anything model (sam) for remote sensing applications: From zero to one shot. *International Journal of Applied Earth Observation and Geoinformation*, *124*, 103540. doi: https://doi.org/10.1016/j.jag.2023.103540.

Overton, K., Dempster, T., Oppedal, F., Kristiansen, T. S., Gismervik, K., & Stien, L. H. (2019). Salmon lice treatments and salmon mortality in Norwegian aquaculture: a review. *Reviews in Aquaculture*, *11*, 1398–1417. doi: 10.1111/raq.12299. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/raq.12299.

Pan, Y., Sun, J., Yu, H., Luck, J., Bai, G., Chamara, N., Ge, Y., & Awada, T. (2024). Building multi-agent copilot towards autonomous agricultural data management and analysis. doi: 10.48550/arXiv.2411.00188 arXiv:2411.00188 [cs.AI].

Parent, M. I., Stryhn, H., Hammell, K. L., Grant, J., & Vanderstichel, R. (2024). Estimation and comparison of connectivity measures for the dispersal of Lepeophtheirus salmonis sea lice among Atlantic salmon farms in new Brunswick, Canada. *Journal of Experimental Marine Biology and Ecology*, *572*, 151992. doi: https://doi.org/10.1016/j.jembe.2024.151992.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Pavlidis, M., Papaharisis, L., Adamek, M., Steinhagen, D., Jung-Schroers, V., Kristiansen, T., Theodoridi, A., & Otero Lourido, F. (2023). *Research for PECH Committee –Animal welfare of farmed fish*. Study PE 747.257, European Parliament, Policy Department for Structural and Cohesion Policies Brussels. URL: https://www.europarl.europa.eu/thinktank/en/document/IPOL_STU(2023)747257. doi: 10.2861/598154.

Penston, M., Millar, C., Zuur, A., & Davies, I. (2008). Spatial and temporal distribution of lepeophtheirus salmonis (krøyer) larvae in a sea loch containing Atlantic salmon, salmo salar l., farms on the north-west coast of scotland. *Journal of Fish Diseases*, *31*, 361–371. doi: 10.1111/j.1365-2761.2008.00915.x.

Penston, M. J., & Davies, I. M. (2009). An assessment of salmon farms and wild salmonids as sources of Lepeophtheirus salmonis (Krøyer) copepodids in the water column in Loch Torridon, Scotland. *Journal of Fish Diseases*, *32*, 75–88. doi: 10.1111/j.1365-2761.2008.00986.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2761.2008.00986.x.

Peters, M. E., Ruder, S., & Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*, .

Pettersen, J. M., Bracke, M. B., Midtlyng, P. J., Folkedal, O., Stien, L. H., Steffenak, H., & Kristiansen, T. S. (2014). Salmon welfare index model 2.0: an extended model for overall welfare assessment of caged Atlantic salmon, based on a review of selected welfare indicators and intended for fish health professionals. *Reviews in Aquaculture*, *6*, 162–179. doi: 10.1111/raq.12039.

Pettersen, R., Braa, H. L., Gawel, B. A., Letnes, P. A., Sæther, K., & Aas, L. M. S. (2019). Detection and classification of lepeophterius salmonis (krøyer, 1837) using underwater hyperspectral imaging. *Aquacultural Engineering*, *87*, 102025. doi: 10.1016/j.aquaeng.2019.102025.

Peña Fernández, A., Norton, T., Tullo, E., van Hertem, T., Youssef, A., Exadaktylos, V., Vranken, E., Guarino, M., & Berckmans, D. (2018). Real-time monitoring of broiler flock's welfare status using camera-based technology. *Biosystems Engineering*, *173*, 103–114. doi: 10.1016/j.biosystemseng.2018.05.008.

Picon, A., San-Emeterio, M. G., Bereciartua-Perez, A., Klukas, C., Eggers, T., & Navarra-Mestre, R. (2022). Deep learning-based segmentation of multiple species of weeds and corn crop using synthetic and real image datasets. *Computers and Electronics in Agriculture*, *194*, 106719. doi: https://doi.org/10.1016/j.compag.2022.106719.

Porter, M., Duncan, N., Mitchell, D., & Bromagea, N. (1999). The use of cage lighting to reduce plasma melatonin in Atlantic salmon (Salmo salar) and its effects on the inhibition of grilsing. *Aquaculture*, *176*, 237–244. doi: https://doi.org/10.1016/S0044-8486(99)00113-1.

Pratiwy, F. M., & Aisyah, A. (2026). Literature review: The role of IoT and AI in water quality monitoring in aquaculture. *GPH-International Journal of Applied Science*, *9*, 98–111. doi: 10.5281/zenodo.18632014.

Qin, L., Chen, Q., Feng, X., Wu, Y., Zhang, Y., Li, Y., Li, M., Che, W., & Yu, P. S. (2026). Large language models meet NLP: a survey. *Frontiers of Computer Science*, *20*, 2011361. doi: 10.1007/s11704-025-50472-3.

Rabanal, H. R., & Fishery and Aquaculture Economics and Policy Division (1998). *History of aquaculture*. Technical Report, . URL: https://openknowledge.fao.org/handle/20.500.14283/ag158e edition: 1 Num Pages: 19.

Rabe, B., Murray, A. G., Ives, S. C., Moriarty, M., & Morris, D. J. (2024). Searching for sea lice: surveillance to assess environmental infection pressures to model and inform sea lice infestation management. *Aquaculture Environment Interactions*, *16*, 241–262. doi: 10.3354/aei00483.

Rayamajhi, A., Jahanifar, H., & Mahmud, M. S. (2024). Measuring ornamental tree canopy attributes for precision spraying using drone technology and self-supervised segmentation. *Computers and Electronics in Agriculture*, *225*, 109359. doi: https://doi.org/10.1016/j.compag.2024.109359.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). doi: 10.1109/CVPR.2016.91.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, .

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-Networks. In *Conference on Empirical Methods in Natural Language Processing*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). FasterR-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Rethmeier, N., & Augenstein, I. (2023). A primer on contrastive pretraining in language processing: Methods, lessons learned, and perspectives. *ACM Computing Surveys*, *55*, 1–17. doi: https://doi.org/10.1145/3561970.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, *3*, 333–389. doi: 10.1561/1500000019.

Rocha, G. M., Martinelli, A. B., Utz, L. R. P., & Loureiro Fernandes, L. F. (2019). Using internal transcribed spacers for calanoida (crustacea, copepoda) species identification. *Marine Biodiversity*, *49*, 1565–1571. doi: 10.1007/s12526-018-0902-9.

Rosendal, G. K., & Olesen, I. (2022). Overcoming barriers to breeding for increased lice resistance in farmed Atlantic salmon: A case study from norway. *Aquaculture*, *548*, 737574. doi: 10.1016/j.aquaculture.2021.737574.

Rossi, L., Zoli, M., Capoccioni, F., Pulcini, D., Martini, A., & Bacenetti, J. (2024). Insights into different marine aquaculture infrastructures from a life cycle perspective. *Aquacultural Engineering*, *107*, 102462. doi: 10.1016/j.aquaeng.2024.102462.

Roth, B., Rappensperger, S., Qiu, T., Imamović, H., Wörmann, J., & Shen, H. (2025). Resource-efficient adaptation of large language models for text embeddings via prompt engineering and contrastive fine-tuning. doi: 10.48550/arXiv.2507.22729 arXiv:2507.22729 [cs].

Rouzegar, H., & Makrehchi, M. (2024). Enhancing text classification through llm-driven active learning and human annotation. *arXiv preprint arXiv:2406.12114*, .

Roy, J. L., & Boulding, E. G. (2024). Comparison of DNA metabarcoding and morphological diet analysis of lumpfish (Cyclopterus lumpus) as methods of estimating sea lice cleaning efficacy inside commercial atlantic salmon (salmo salar) sea cages. *Aquaculture*, *586*, 740817. doi: https://doi.org/10.1016/j.aquaculture.2024.740817.

Ruis, F. A., Liezenga, A. M., Heslinga, F. G., Ballan, L., Eker, T. A., den Hollander, R. J. M., van Leeuwen, M. C., Dijk, J., & Huizinga, W. (2024). Improving object detector training on synthetic data by starting with a strong baseline methodology. In K. E. Manser, C. L. Howell, R. M. Rao, & C. D. Melo (Eds.), *Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II* (p. 1303511). International Society for Optics and Photonics National Harbor, Maryland, United States: SPIE volume 13035. doi: https://doi.org/10.1117/12.3013441.

Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach , 4/E.*. Pearson.

Sadiq, M. B., Ramanoon, S. Z., Shaik Mossadeq, W. M., Mansor, R., & Syed Hussain, S. S. (2019). Dairy farmers' perceptions of and actions in relation to lameness management. *Animals*, *9*, 270. doi: 10.3390/ani9050270.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, .

Salama, N., Murray, A., & Rabe, B. (2011). Modelling dispersal of salmon lice in a large fjordic system: Loch linnhe, scotland. In *19th International Congress on Modelling and Simulation, Perth, Australia* (pp. 12–16). Perth, Australia.

Saleh, A., Sheaves, M., & Rahimi Azghadi, M. (2022). Computer vision and deep learning for fish classification in underwater habitats: A survey. *Fish and Fisheries*, *23*, 977–999. doi: 10.1111/faf.12666. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/faf.12666.

Salehi, F., Abbasi, E., & Hassibi, B. (2019). The impact of regularization on high-dimensional logistic regression. *Advances in Neural Information Processing Systems*, *32*. doi: https://doi.org/10.48550/arXiv.1906.03761.

Samsing, F., Johnsen, I., Dempster, T., Oppedal, F., & Treml, E. A. (2017). Network analysis reveals strong seasonality in the dispersal of a marine parasite and identifies areas for coordinated management. *Landscape Ecology*, *32*, 1953–1967. doi: https://doi.org/10.1007/s10980-017-0557-0.

Samways, K. M. (2025). Supplementation of adult Atlantic salmon (*Salmo salar*) restores biofilm biomass and benthic metabolism. *Fisheries Research*, *291*, 107556. doi: 10.1016/j.fishres.2025.107556.

Santurtun, E., Broom, D. M., & Phillips, C. J. C. (2018). A review of factors affecting the welfare of Atlantic salmon (Salmo salar). *Animal Welfare*, *27*, 193–204. doi: 10.7120/09627286.27.3.193.

Sarker, I. H. (2022). AI-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems. *Sn Computer Science*, *3*, 158. doi: 10.1007/s42979-022-01043-x.

Sauphar, C., Stolz, C., Tuene, S. A., Gansel, L. C., & Aas, G. H. (2023). Atlantic salmon (salmo salar) distribution and vertical size-stratification in a commercial sea cage: A case study. *Aquaculture*, (p. 740356). doi: 10.1016/j.aquaculture.2023.740356.

Schneider, K. J., Urvoy, C., Müller, C., Brucker, L., Christ, F., Kimmich, S., Rauen, A., & Harmsen, B. (2025). InKalkTier–enabling livestock farming experts to model and manage their data. In *45. GIL-Jahrestagung, Digitale Infrastrukturen für eine nachhaltige Land-, Forst-und Ernährungswirtschaft* (pp. 375–380). Gesellschaft für Informatik eV.

Schram, T. A. (1993). Supplementary descriptions of the developmental stages of lepeophtheirus salmonis (krøyer, 1837) (copepoda: Caligidae). (p. 30–47). Chichester: Ellis Horwood Limited.

Sentence-Transformers (n.d.). all-minilm-l12-v2. https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2. Accessed: 2025-06-03.

Serra-Llinares, R. M., Bjørn, P. A., Finstad, B., Nilsen, R., Harbitz, A., Berg, M., & Asplin, L. (2014). Salmon lice infection on wild salmonids in marine protected areas: an evaluation of the Norwegian'National Salmon Fjords'. *Aquaculture Environment Interactions*, *5*, 1–16. doi: https://doi.org/10.3354/aei00090.

Shaikh, O., Gligoric, K., Khetan, A., Gerstgrasser, M., Yang, D., & Jurafsky, D. (2024). Grounding gaps in language model generations. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies (Volume 1: Long Papers) (pp. 6279–6296). Mexico City, Mexico: Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.348.

Shephard, S., & Gargan, P. (2017). Quantifying the contribution of sea lice from aquaculture to declining annual returns in a wild Atlantic salmon population. *Aquaculture Environment Interactions*, *9*, 181–192. doi: https://doi.org/10.3354/aei00223.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, .

Singh, K., Navaratnam, T., Holmer, J., Schaub-Meyer, S., & Roth, S. (2024). Is synthetic data all we need? benchmarking the robustness of models trained with synthetic images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (pp. 2505–2515). Seattle, WA, USA. doi: https://doi.org/10.1109/CVPRW63382.2024.00257.

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Amin, M., Hou, L., Clark, K., Pfohl, S. R., Cole-Lewis, H. et al. (2025). Toward expert-level medical question answering with large language models. *Nature Medicine*, (pp. 1–8). doi: https://doi.org/10.1038/s41591-024-03423-7.

Skarðhamar, J., Fagerli, M. N., Reigstad, M., Sandvik, A. D., & Bjørn, P. A. (2019). Sampling planktonic salmon lice in Norwegian fjords. *Aquaculture Environment Interactions*, *11*, 701–715. doi: 10.3354/aei00342.

Souza, V. R., Christ, D. C. F., Wiese, I. S., Naves, T. F., de Lima Soares, T. W., da Silva Soares, A., & de Oliverira Berretta, L. (2025). Automated synthetic data for computer vision: Blender-COCO pipeline enhancing YOLOv8. In *2025 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (pp. 1–6). doi: 10.1109/SIBGRAPI67909.2025.11223399 iSSN: 2377-5416.

de Souza P. Moreira, G., Ak, R., Schifferer, B., Xu, M., Osmulski, R., & Oldridge, E. (2024). Enhancing Q&A text retrieval with ranking models: Benchmarking, fine-tuning and deploying rerankers for RAG. arXiv:2409.07691.

Stien, L. H., Bracke, M. B., Folkedal, O., Nilsson, J., Oppedal, F., Torgersen, T., Kittilsen, S., Midtlyng, P. J., Vindas, M. A., Øverli, Ø., & Kristiansen, T. S. (2013). 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. doi: [10.1111/j.1753-5131.2012.01083.x](https://doi.org/10.1111/j.1753-5131.2012.01083.x).

Stämpfli, K., Buchwalder, T., Fröhlich, E., & Roth, B. (2012). Influence of front curtain design on nest choice by laying hens. *British Poultry Science*, *53*, 553–560. doi: 10.1080/00071668.2012.725201. PMID: 23281747.

Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., & Luo, P. (2021). Sparse R-CNN: End-to-end object detection with learnable proposals. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 14449–14458). doi: 10.1109/CVPR46437.2021.01422.

Tamber, M. S., Bao, F. S., Xu, C., Luo, G., Kazi, S., Bae, M., Li, M., Mendelevitch, O., Qu, R., & Lin, J. (2025). Benchmarking LLM faithfulness in RAG with evolving leaderboards. In S. Potdar, L. Rojas-Barahona, & S. Montella (Eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track* (pp. 799–811). Suzhou (China): Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-industry.54.

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 6105–6114). PMLR.

Tang, H., & Jia, K. (2023). A new benchmark: On the utility of synthetic data with blender for bare supervised learning and downstream domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 15954–15964). Vancouver, Canada: IEEE. doi: https://doi.org/10.1109/CVPR52729.2023.01531.

Taranger, G. L. et al. (2015). Risk assessment of the environmental impact of Norwegian Atlantic salmon farming. *ICES Journal of Marine Science*, *72*, 997–1021. doi: 10.1093/icesjms/fsu132.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M. et al. (2025). Gemma 3 technical report. *arXiv:2503.19786*, .

Team, X. L.-C. et al. (2026). MiMo-V2-flash technical report. doi: 10.48550/arXiv.2601.02780 arXiv:2601.02780 [cs].

Thompson, C. R., Bron, J. E., Bui, S., Dalvin, S., Fordyce, M. J., Furmanek, T., á Norði, G., & Skern-Mauritzen, R. (2021). Illuminating the planktonic stages of salmon lice: A unique fluorescence signal for rapid identification of a rare copepod in zooplankton assemblages. *Journal of Fish Diseases*, *44*, 863–879. doi: https://doi.org/10.1111/jfd.13345.

Thompson, C. R. S., Bron, J., Bui, S., Dalvin, S., Fordyce, M. J., Á Norði, G., & Skern-Mauritzen, R.

(2022). A novel method for the rapid enumeration of planktonic salmon lice in a mixed zooplankton assemblage using fluorescence. *Aquaculture Research*, *53*, 2317–2329. doi: 10.1111/are.15750.

Thornburg, J. (2025). Feed the fish: A review of aquaculture feeders and their strategic implementation. *Journal of the World Aquaculture Society*, *56*, e70016. doi: 10.1111/jwas.70016. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jwas.70016.

Thorstad, E. B., Todd, C. D., Uglem, I., Bjørn, P. A., Gargan, P. G., Vollset, K. W., Halttunen, E., Kålås, S., Berg, M., & Finstad, B. (2015). Effects of salmon lice Lepeophtheirus salmonis on wild sea trout Salmo trutta—a literature review. *Aquaculture Environment Interactions*, *7*, 91–113. doi: 10.3354/aei00142.

Thorvaldsen, T., Frank, K., & Sunde, L. M. (2019). Practices to obtain lice counts at Norwegian salmon farms: status and possible implications for representativity. *Aquaculture Environment Interactions*, *11*, 393–404. doi: 10.3354/aei00323.

Torrissen, O., Jones, S., Asche, F., Guttormsen, A., Skilbrei, O. T., Nilsen, F., Horsberg, T. E., & Jackson, D. (2013). Salmon lice –impact on wild salmonids and salmon aquaculture. *Journal of Fish Diseases*, *36*, 171–194. doi: https://doi.org/10.1111/jfd.12061.

Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., & Birchfield, S. (2018). Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Salt Lake City, UT, USA: IEEE.

Troell, M. et al. (2023). Perspectives on aquaculture's contribution to the sustainable development goals for improved human and planetary health. *Journal of the World Aquaculture Society*, *54*, 251–342. doi: 10.1111/jwas.12946. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jwas.12946.

Truett, G. E., Heeger, P., Mynatt, R., Truett, A., Walker, J., & Warman, M. (2000). Preparation of PCR-quality mouse genomic DNA with hot sodium hydroxide and tris (hotshot). *Biotechniques*, *29*, 52–54. doi: https://doi.org/10.2144/00291bm09.

Ultralytics, G. J. (2020). YOLOv5 by Ultralytics.

Urbina, M. A., Cumillaf, J. P., Paschke, K., & Gebauer, P. (2019). Effects of pharmaceuticals used to treat salmon lice on non-target species: Evidence from a systematic review. *Science of The Total Environment*, *649*, 1124–1136. doi: 10.1016/j.scitotenv.2018.08.334.

Ursinus, W., Schepers, F., De Mol, R., Bracke, M., Metz, J., & Koerkamp, P. G. (2009). COWEL: a decision support system to assess welfare of husbandry systems for dairy cattle. *Animal Welfare*, *18*, 545–552. doi: https://doi.org/10.1017/S096272860000097X.

Ushio, A., & Camacho-Collados, J. (2022). T-NER: an all-round python library for transformer-based named entity recognition. *arXiv preprint arXiv:2209.12616*, .

Van Nuffel, A., Zwertvaegher, I., Pluym, L., Van Weyenberg, S., Thorup, V. M., Pastell, M., Sonck, B., & Saeys, W. (2015). Lameness detection in dairy cows: Part 1. how to distinguish between non-lame and lame cows based on differences in locomotion or behavior. *Animals*, *5*, 838–860. doi: 10.3390/ani5030387.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need, . *30*.

Vladika, J., & Matthes, F. (2025). On the influence of context size and model choice in retrieval-augmented generation systems. doi: 10.48550/arXiv.2502.14759 arXiv:2502.14759 [cs].

Vonholdt-Wenker, M. L., Benthin, J., Kauselmann, K., Bracke, M. B. M., & Krause, E. T. (2026). Semantic modelling of animal welfare explained – part 2: The basis of welfare weighting and usage of scientific information. *Archives Animal Breeding*, *69*, 383–396. doi: 10.5194/aab-69-383-2026.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, *2018*, 7068349. doi: 10.1155/2018/7068349. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2018/7068349.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024a). YOLOv10: Real-time end-to-end object detection. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, & C. Zhang (Eds.), *Advances in Neural Information Processing Systems* (pp. 107984–108011). Curran Associates, Inc. volume 37.

Wang, B., Xu, C., Zhao, X., Ouyang, L., Wu, F., Zhao, Z., Xu, R., Liu, K., Qu, Y., Shang, F. et al. (2024b). Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*, .

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7464–7475). doi: 10.1109/CVPR52729.2023.00721.

Wang, C.-Y., Liao, H.-Y. M., & Yeh, I.-H. (2022a). Designing network design strategies through gradient path analysis. arXiv:2211.04800.

Wang, C.-Y., Yeh, I.-H., & Mark Liao, H.-Y. (2024c). YOLOv9: Learning what you want to learn using programmable gradient information. In *European Conference on Computer Vision* (pp. 1–21). Springer Cham: Springer Nature Switzerland. doi: 10.1007/978-3-031-72751-1_1.

Wang, H., Blok, V., & van Hilten, M. (2025a). ELSA Labs for responsible AI: a novel approach for addressing ethical, legal, social issues. *Journal of Responsible Innovation*, *12*, 2563944. doi: 10.1080/23299460.2025.2563944. _eprint: https://doi.org/10.1080/23299460.2025.2563944.

Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., & Wei, F. (2022b). Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, .

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., & Wei, F. (2024d). Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, .

Wang, L., Zhang, X., Yang, K., Yu, L., Li, C., Hong, L., Zhang, S., Li, Z., Zhong, Y., & Zhu, J. (2021). Memory replay with data compression for continual learning.

Wang, Q., Nguyen, T.-T., Huang, J. Z., & Nguyen, T. T. (2018). An efficient random forests algorithm for high dimensional data classification. *Advances in Data Analysis and Classification*, *12*, 953–972. doi: https://doi.org/10.1007/s11634-018-0318-1.

Wang, S., Liu, J., Song, S., Cheng, J., Fu, Y., Guo, P., Fang, K., Zhu, Y., & Dou, Z. (2024e). Domainrag: A chinese benchmark for evaluating domain-specific retrieval-augmented generation. doi: 10.48550/arXiv.2406.05654 arXiv:2406.05654 [cs.CL].

Wang, Y., Ji, P., Yang, C., Li, K., Hu, M., Li, J., & Sartoretti, G. (2025b). MCTS-Judge: Test-time scaling in LLM-as-a-Judge for code correctness evaluation. arXiv:2502.12468.

Wang, Y., Zhao, S., Wang, Z., Fan, M., Zhang, X., Zhang, Y., Wang, Z., Huang, H., & Liu, T. (2025c). RAG+: Enhancing retrieval-augmented generation with application-aware reasoning. doi: 10.48550/arXiv.2506.11555 arXiv:2506.11555 [cs].

Webster, J. (1994). Assessment of animal welfare: The five freedoms. *Animal Welfare: A Cool Eye Towards Eden; Blackwell Science: Oxford, UK*, (pp. 10–14).

Whay, H. R., & Shearer, J. K. (2017). The Impact of Lameness on Welfare of the Dairy Cow. *Veterinary Clinics of North America: Food Animal Practice*, *33*, 153–164. doi: 10.1016/j.cvfa.2017.02.008.

Wood-Gush, D., & Beilharz, R. (1983). The enrichment of a bare environment for animals in confined conditions. *Applied Animal Ethology*, *10*, 209–217. doi: https://doi.org/10.1016/0304-3762(83)90142-6.

Wu, H. (2013). Application of orthogonal experimental design for the automatic software testing. *Applied mechanics and materials*, *347*, 812–818. doi: https://doi.org/10.4028/www.scientific.net/AMM.347-350.812.

Wu, S., Xiong, Y., Cui, Y., Wu, H., Chen, C., Yuan, Y., Huang, L., Liu, X., Kuo, T.-W., Guan, N., & Xue, C. J. (2025). Retrieval-augmented generation for natural language processing: A survey. doi: 10.48550/arXiv.2407.13193 arXiv:2407.13193 [cs].

Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, *396*, 39–64. doi: 10.1016/j.neucom.2020.01.085.

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., & He, L. (2022). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, *135*, 364–381. doi: https://doi.org/10.1016/j.future.2022.05.014.

Wu, X., Yao, W., Chen, J., Pan, X., Wang, X., Liu, N., & Yu, D. (2024). From language modeling to instruction following: Understanding the behavior shift in llms after instruction tuning. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (pp. 2341–2369). Mexico City, Mexico: Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.130.

Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D., & Nie, J.-Y. (2024). C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* SIGIR '24 (p. 641–649). New York, NY, USA: Association for Computing Machinery. doi: https://doi.org/10.1145/3626772.3657878.

Xiao, T., Nerini, F. F., Matthews, H. D., Tavoni, M., & You, F. (2025). Environmental impact and net-zero pathways for sustainable artificial intelligence servers in the USA. *Nature Sustainability*, *8*, 1541–1553. doi: 10.1038/s41893-025-01681-y.

Xiao, X., Liu, T., He, S., Li, P., Gu, Y., Li, P., & Dong, J. (2026). A multi-fish tracking and behavior modeling framework for high-density cage aquaculture. *Sensors*, *26*, 256. doi: 10.3390/s26010256.

Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., & Lan, X. (2020). A review of object detection based on deep learning. *Multimedia Tools and Applications*, *79*, 23729–23791. doi: 10.1007/s11042-020-08976-6.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C. et al. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, .

Yang, J., Xiao, R., Zhou, Y., Nie, Y., Guo, Z., & Shen, Y. (2026). Hypoxic surfacing fish detection for abnormal behavior warning based on MSYOLO in Recirculating Aquaculture Systems. *Information Processing in Agriculture*, . doi: 10.1016/j.inpa.2026.03.003.

Yang, L., Liu, Y., Yu, H., Fang, X., Song, L., Li, D., & Chen, Y. (2021). Computer vision models in intelligent aquaculture with emphasis on fish detection and behavior analysis: A review. *Archives of Computational Methods in Engineering*, *28*, 2785–2816. doi: 10.1007/s11831-020-09486-2.

Yao, L., Mao, C., & Luo, Y. (2019). KG-BERT: BERT for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, .

Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, *4*, 100211. doi: 10.1016/j.hcc.2024.100211.

Yildiz, H. Y., Chatzifotis, S., Anastasiadis, P., Parisi, G., & Papandroulakis, N. (2021). Testing of the Salmon Welfare Index Model (SWIM 1.0) as a computational welfare assessment for sea-caged european sea bass. *Italian Journal of Animal Science*, *20*, 1423–1430. doi: 10.1080/1828051X.2021.1961106.

Yue, K., & Shen, Y. (2022). An overview of disruptive technologies for aquaculture. *Aquaculture and Fisheries*, *7*, 111–120. doi: 10.1016/j.aaf.2021.04.009.

Yue, M. (2025). A survey of large language model agents for question answering. *arXiv preprint arXiv:2503.19213*, .

Zhang, B. et al. (2025a). Sentient agent as a judge: Evaluating higher-order social cognition in large language models. arXiv:2505.02847.

Zhang, C., Bracke, M., da Silva Torres, R., & Gansel, L. C. (2024a). Rapid detection of salmon louse larvae in seawater based on machine learning. *Aquaculture*, *592*, 741252. doi: 10.1016/j.aquaculture.2024.741252.

Zhang, C., Gansel, L. C., Bracke, M., & da Silva Torres, R. (2025b). An image synthesis framework for enhanced salmon louse larvae (*Lepeophtheirus Salmonis*) detection in complex seawater conditions. *Computers and Electronics in Agriculture*, *239*, 110985. doi: 10.1016/j.compag.2025.110985.

Zhang, C., da Silva Torres, R., Gansel, L. C., & Bracke, M. (2026). Automated extraction of scientific statements for integrated assessment of animal welfare using language models. *Smart Agricultural Technology*, *13*, 101814. doi: 10.1016/j.atech.2026.101814.

Zhang, J., Huang, J., Jin, S., & Lu, S. (2024b). Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*, 5625–5644. doi: 10.1109/TPAMI.2024.3369699.

Zhang, R., Qiu, M., Tan, Z., Zhang, M., Lu, V., Peng, J., Xu, K., Agudelo, L. Z., Qian, P., & Chen, T. (2025c). Symbiotic cooperation for web agents: Harnessing complementary strengths of large and small llms. arXiv:2502.07942.

Zhang, T., Yang, Y., Liu, Y., Liu, C., Zhao, R., Li, D., & Shi, C. (2024c). Fully automatic system for fish biomass estimation based on deep neural network. *Ecological Informatics*, *79*, 102399. doi: 10.1016/j.ecoinf.2023.102399.

Zhang, Y., Li, M., Long, D., Zhang, X., Lin, H., Yang, B., Xie, P., Yang, A., Liu, D., Lin, J., Huang, F., & Zhou, J. (2025d). Qwen3 embedding: Advancing text embedding and reranking through foundation models. arXiv:2506.05176.

Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., Yang, L., Zhang, W., Jiang, J., & Cui, B. (2024a). Retrieval-augmented generation for AI-Generated content: A survey. doi: 10.48550/arXiv.2402.19473 arXiv:2402.19473 [cs] version: 5.

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024b). Detrs beat yolos on real-time object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 16965–16974). Los Alamitos, CA, USA: IEEE Computer Society. doi: 10.1109/CVPR52733.2024.01605.

Zhao, Z., Yuan, H., Liu, J., Chen, H., Ying, H., Zhou, S., & Yu, S. (2025). Evaluating entity retrieval in electronic health records: a semantic gap perspective. doi: 10.48550/arXiv.2502.06252 arXiv:2502.06252 [cs] version: 1.

Zheng, L., Zhang, C., Bracke, M., Gansel, L., & da Silva Torres, R. (2024). Detecting salmon lice in seawater using synthetic datasets. In *2024 International Conference on Machine Learning and Applications*

(ICMLA) (pp. 894–899). Miami, FL, USA. doi: https://doi.org/10.1109/ICMLA61862.2024.00130.

Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., & Zuo, W. (2021). Enhancing geometric factors in model learning and inference for object detection and instance segmentation. arXiv:2005.03572.

Zhou, X. et al. (2025). The use of data for health and welfare management of farmed salmons in Norway, Scotland, and Ireland. *Preventive Veterinary Medicine*, *239*, 106512. doi: 10.1016/j.prevetmed.2025.106512.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). Deformable DETR: Deformable transformers for end-to-end object detection. arXiv:2010.04159.

Zonderland, J. J., Wolthuis-Fillerup, M., van Reenen, C. G., Bracke, M. B., Kemp, B., den Hartog, L. A., & Spoolder, H. A. (2008). Prevention and treatment of tail biting in weaned piglets. *Applied Animal Behaviour Science*, *110*, 269–281. doi: https://doi.org/10.1016/j.applanim.2007.04.005.

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, *111*, 257–276. doi: [10.1109/JPROC.2023.3238524](https://doi.org/10.1109/JPROC.2023.3238524).

Zulkifli, I. (2013). Review of human-animal interactions and their impact on animal productivity and welfare. *Journal of animal science and biotechnology*, *4*, 1–7. doi: https://doi.org/10.1186/2049-1891-4-25.

Acknowledgements

Four years have passed in the blink of an eye. As I finish this thesis, I am also coming to the end of my journey as a student. Looking back, these four years have been filled with moments of excitement and achievement, but also challenges, frustration, and uncertainty. What made this journey truly special was the people I met along the way. I have been fortunate to be surrounded by so many kind, supportive, and interesting people, and I would like to take this opportunity to thank everyone who has been part of these memorable years.

First and foremost, I would like to express my sincere gratitude to my three supervisors, Ricardo, Marc, and Lars.

Ricardo, thank you for your trust, patience, and guidance throughout these four years. Your dedication to research and attention to detail have taught me a lot about how to think critically and approach scientific problems. I still remember how you patiently slowed down when I struggled to follow our discussions in the beginning, and how you always encouraged me when I faced difficulties or even doubted myself. Your questions during our meetings often helped me see what really mattered.

Marc, thank you for your encouragement and challenging me to look at my research from different perspectives. Your comments often pushed me to think beyond the technical details and consider the bigger picture. I especially appreciate how you respected my ideas and encouraged me to develop my own opinions. I still remember when you suggested that I write a review paper, and I explained why I preferred to focus on something else first. Instead of being disappointed, you were happy that I had my own plan. That moment meant a lot to me.

Lars, thank you for your enthusiasm, kindness, and constant encouragement. Your extensive knowledge and experience in aquaculture have been invaluable to my research. I will never forget the warm welcome from you and your family when you invited me to stay at your home during my visit to Norway. I also remember how nervous I was before giving my first lecture in NTNU, and how your encouragement

helped me take that first step.

I feel very lucky to have learned from and worked with all three of you!!

I would also like to thank **Ioannis** for creating such an energetic and welcoming atmosphere at AIN, and **Peter** and **Eldert** for supporting the opportunity for me to pursue my PhD at Wageningen University & Research. **Hariëtte**, thank you for always greeting me with a warm smile and being so willing to help. **Miranda, Areerat, Sam, and Paul**, thank you for your help with everything from course registration and conference arrangements to the final steps of my PhD. Your support made many things much easier along the way.

I feel especially lucky to have spent these four years as part of both AIN and ABE. So much of what I will remember comes from the everyday moments we shared. **Leandro**, I could not have asked for a better officemate, and I am even happier to have gained such a close friend. **Rubens**, it was wonderful to meet you at WUR and even better to meet again in Brazil! I hope to meet you again! **Taniya**, thank you for sharing an office with me and giving me the opportunity to co-supervise a master's student. **Vassilis**, your Signal meetings were always entertaining, and I hope your stand-up comedy career continues to flourish! **Diego**, thank you for being such a dedicated and inspiring friend and officemate. **Run**, thank you for helping me discover more about life in the Netherlands. **Jingye**, our discussions always gave me something new to think about. **Lu, Hilmy, Rob, Thijs, and Beichen**, I really enjoyed our reinforcement learning reading sessions. I learned a lot from reading and discussing papers together. **Vishal, Aike, and Yingjie**, thank you for organizing so many interesting activities and meetings that made AIN such a lively place. **Guilherme**, I am glad to have made another good friend from Brazil, and I hope we will have more opportunities to work together. **Xiongguo and Tiana**, it was lovely to become colleagues again after our master's studies, and I wish you both all the best. **Eva**, I still wish I could fully understand your research—it sounds complicated but interesting! **Anna, Michiel, Alex, Ron, Michele, Youssef, Negar, and Taoyu**, thank you for the lunches, coffee breaks, and conversations.

I am equally grateful to all the friends who made life outside research so enjoyable. **Ziye**, our runs, parties, and conversations always helped me relax, and I have learned a lot from your perspective on many things. You are a good thinker. Thank you for helping me out all the time. **Jianchao**, I have learned a lot from our discussions about research, and I will miss our coffee and conversations. **Jingxin**, I really enjoyed our dinners, chats, and badminton games. I will definitely meet you when I visit Beijing! **Maedeh**, thank you for the fun conversations and for sharing an office at ABE. **Lei**, I will remember our chats and the days we spent working on your theses. **Xin** and

Laiquan, thank you for helping me during the early days of my PhD. **Shutong**, even though we did not meet often, our conversations were always enjoyable. **Yuxiang and Fei**, I enjoyed sharing an office and having parties together. **Zhenpeng**, our hikes helped me discover how much I enjoy hiking. **Xindong**, our gym sessions brought some welcome balance to busy days. **Julio**, your Spanish and ability to connect with people has always impressed me. **Jingrui, Shuyi, and George**, I am glad our paths crossed during these years. **Jinhan**, thank you for sharing your experiences and advice about work. To everyone else at AIN and ABE, and to the many friends I cannot mention individually, thank you for making these four years so full of good memories.

I also want to thank my thesis committee, **Prof. Bedir Tekinerdogan**, **Prof. Geert Wiegertjes**, **Dr Øystein Patursson**, and **Dr Peihua Han**, for their careful reading of this thesis and their critical feedback, which helped me bring my PhD journey to a fulfilling close. Thank you!

最后，我要感谢我的家人。我的父母，谢谢你们一直以来对我毫无保留的支持。即使家里困难，也从未让我有过压力，反而让我放心向前追求自己的目标。我还要感谢我的姑姑和姑父，中学六年，你们一直照顾和教导我，帮助我努力考上了大学，没有你们的支持与鼓励，我不可能走到今天。我还要感谢我的表哥，从小到大你一直是我的榜样，你的步伐激励着我，而你也在我迷茫犹豫的时候给予我最直接的建议，你甚至比我还要相信我自己，谢谢你！我还要感谢我的伴侣，张浏玥，谢谢你对我的陪伴，这一路走来一直以来鼓励我，相信我。你对我至关重要！Your love and support have meant so much to me, especially during the difficult moments.

Four years may seem like a long time, yet they have gone by so quickly. Beyond this thesis, what I will treasure most are the people I have met and the ordinary but wonderful moments we shared: conversations in the office, morning coffees, meals together, and all the places we explored along the way. Although my PhD journey is coming to an end, our friendships and memory never end. In the future, we may live at the different corner of the planet, but our paths will cross again somewhere in the world.

Good morning :)

About the author

Chao Zhang was born on February 18, 1998, in Jiangsu, China. He received his BSc degree in Agricultural Engineering from Jiangsu University in 2019, followed by an MSc degree in Agricultural Engineering from Zhejiang University in 2022. During his master' s studies, his research focused on applying computer vision (CV) techniques to fish processing monitoring.

In the final year of his MSc program, he was awarded a scholarship from the China Scholarship Council (CSC) to support his doctoral studies overseas. He subsequently began his PhD at Wageningen University & Research, jointly affiliated with the Agricultural Biosystems Engineering group and the Artificial Intelligence group. His doctoral research focused on the application of artificial intelligence to support sustainable salmon farming.

Peer-reviewed journal publications

1. **Zhang, C.**, da Silva Torres, R., Gansel, L. C., & Bracke, M. (2026). Automated extraction of scientific statements for integrated assessment of animal welfare using language models. *Smart Agricultural Technology*, 13, 101814. `https://doi.org/10.1016/j.atech.2026.101814`

2. **Zhang, C.**, Gansel, L. C., Bracke, M., & da Silva Torres, R. (2025). An image synthesis framework for enhanced salmon louse larvae (*Lepeophtheirus salmonis*) detection in complex seawater conditions. *Computers and Electronics in Agriculture*, 239(B), 110985. `https://doi.org/10.1016/j.compag.2025.110985`

3. **Zhang, C.**, Bracke, M., da Silva Torres, R., & Gansel, L. C. (2024). Rapid detection of salmon louse larvae in seawater based on machine learning. *Aquaculture*, 592, 741252. `https://doi.org/10.1016/j.aquaculture.2024.741252`

Peer-reviewed and accepted conference publications

1. **Zhang, C.**, Stival, L., Gansel, L. C., Bracke, M., & da Silva Torres, R. (2026). Online synthetic training for annotation-scarce salmon lice detection. Accepted at the *SIBGRAPI Conference on Graphics, Patterns and Images*.

2. Zheng, L., **Zhang, C.**, Bracke, M., Gansel, L. C., & da Silva Torres, R. (2024). Detecting salmon lice in seawater using synthetic datasets. In *Proceedings of the 2024 International Conference on Machine Learning and Applications (ICMLA)*, 894–899. https://doi.org/10.1109/ICMLA61862.2024.00130

Manuscripts under review

1. **Zhang, C.**, Gansel, L. C., Bracke, M., & da Silva Torres, R. (2026). S-MADSS: An agentic framework for evidence-grounded, compliance-oriented decision support in Atlantic salmon farming. Under review at *Smart Agricultural Technology*.

2. de Castro Pereira, R., da Silva Torres, R., **Zhang, C.**, Borges, D. L., Lobo Jr., M., & Pedrini, H. (2026). DiCODeFusion: Diversity-informed clustering for object detector fusion. Under review at *Neurocomputing*.

PE&RC Training and Education Statement

With the training and education activities listed below the PhD candidate has complied with the requirements set by the Graduate School for Production Ecology and Resource Conservation (PE&RC) which comprises of a minimum total of 30 ECTS (= 22 weeks of activities)



Review/project proposal

- Advancing Salmon Welfare Assessment through Artificial Intelligence-based Semantic Modelling and Environmental Monitoring | 4.5 ECTS

In-depth / Topical / On-site Postgraduate Courses

- International DaNuMa Summer School "Machine Learning in Animal Science" | CAU Kiel, German | 2024 | 2 ECTS
- Convolutional Neural Networks | Coursera | 2023 | 1.25 ECTS
- Autonomous Greenhouse Challenge | WUR, Tecent | 2024 | 2.3 ECTS
- Photosynthesis-related research hackathon | International Institute of Tropical Agriculture (IITA), University of Cambridge (U.K.), Jan Ingenhousz Institute (Netherlands) | 2026 | 1.5 ECTS

Invited review of journal manuscripts

- Aquaculture | Implications of inter-observer variability for image-based monitoring of salmon lice (Lepeophtheirus salmonis) in commercial aquaculture | 2026 | 1 ECTS

Competence, skills and career-oriented activities

- Research data management | Wageningen University & Research Library (WUR) | 2023 | 0.45 ECTS
- Introduction to latex | PE&RC (WUR) | 2023 | 0.1 ECTS
- Searching and Organizing Literature for PhD candidates | Wageningen University & Research Library (WUR) | 2023 | 0.6 ECTS
- Scientific Writing | Wageningen in'to Languages (WUR) | 2025 | 2 ECTS

Scientific Integrity/Ethics in science activities

- Introduction to Scientific Integrity (in-person training) | WUR, WGS | 2026 | 0.6 ECTS

PE&RC Retreat, PE&RC Day, and other PE&RC events

- PE&RC retreat | 2023 | 0.9 ECTS
- PE&RC day | 2025 | 0.3 ECTS

National/local scientific meetings, seminars, and discussion groups

- Vision and robotics thematic meeting | 2024, 2025 | 3 ECTS
- Modelling & Simulation Discussion Group | 2024 | 1.5 ECTS
- Digital Innovation Expo | 2024 | 0.3 ECTS

International symposia, workshops and conferences

- Workshop in NTNU (With TUMSAP, Japan) | Norway | 2023 | 1.3 ECTS
- AgEng 2024: Agricultural Engineering Challenges in Existing and New Agroecosystems | Greece | 2024 | 2.2 ECTS
- AQUA 2024 | Denmark | 2024 | 2.5 ECTS
- International Conference on Machine Learning and Applications (ICMLA) | USA | 2024 | 1.3 ECTS
- AQUA 2025 | Spain | 2025 | 2.2 ECTS

Lecturing/Supervision of practicals/tutorials

- FTE 40306 - Advanced machine learning | 2025 | 5 days | 1.5 ECTS
- PE&RC Graduate Course - Introduction to machine learning | 2025 | 4 days | 1.2 ECTS
- AIN 30306 - Artificial Intelligence in Science and Engineering | 2025 | 2 days | 0.6 ECTS
- FTE35306 - Machine learning | 2026 | 1 day | 0.3 ECTS
- PE&RC Graduate Course - Introduction to machine learning | 2026 | 4 days | 1.2 ECTS

BSc/MSc thesis supervision

- Research topic 1: Detecting salmon lice using synthetic datasets | 3 ECTS

Authorship Statement

PhD candidate's name : Chao Zhang

First promotor : Prof. Ricardo da Silva Torres

Title of PhD thesis : Artificial Intelligence for Salmon Farming: Perception, Welfare Modelling, and Decision Support

Chapter 1: General Introduction: I formulated the research scope, wrote the chapter, and revised it based on feedback from my promotor and supervisors.

Chapter 2: Rapid detection of salmon louse larvae in seawater based on machine learning: I conceptualized the study in consultation with my promotor and supervisors and developed the methodology in this chapter. I jointly prepared the dataset. I conducted the experiments and compare 16 different models. I wrote the initial draft and revised it based on comments from my promotor and supervisors.

Chapter 3: An image synthesis framework for enhanced salmon louse larvae (Lepeophtheirus salmonis) detection in complex seawater conditions: I conceptualized the study in consultation with my promotor and supervisors and developed the synthetic data framework presented in this chapter. I jointly prepared the dataset, conducted the experiments, and performed the data analysis. I wrote the initial draft and revised it based on comments from my promotor and supervisors.

Chapter 4: Online Synthetic Training for Annotation-Scarce Salmon Lice Detection: I conceptualized the study in consultation with my promotor, supervisors, and a postdoctoral researcher who is a co-author of this work. I developed the SDOST method presented in this chapter, conducted all experiments, and performed the data analysis. I wrote the initial draft and revised it based on comments from my co-authors.

Chapter 5: Automated extraction of scientific statements for integrated assessment of animal welfare using language models: I conceptualized the study in consultation with my promotor and supervisors. I developed the language model-based method for classifying welfare-related statements, conducted all experiments, performed the data analysis, and developed the user interface. I wrote the initial draft and revised it based on comments from my promotor and supervisors.

Chapter 6: S-MADSS: An Agentic Framework for Evidence-Grounded, Compliance-Oriented Decision Support in Atlantic Salmon Farming: I conceptualized the study in consultation with my promotor and supervisors. I developed the S-MADSS framework presented in this chapter, collected the data used in the study, conducted all experiments, and performed the data analysis. I wrote the initial draft and revised it based on comments from my promotor and supervisors.

Chapter 7: General Discussion: I drafted and wrote the entire chapter. I revised the text twice based on comments and discussions with my promotor and supervisors.

In all chapters except for Chapter 2, I used ChatGPT for language support, including improving the writing style, and correcting grammatical errors. I remain fully responsible for the final content, scientific accuracy, analyses, interpretations, and conclusions presented in this thesis.

Signature promotor

Ricardo Torres

Date: 15-07-2026

Signature PhD candidate

Chao Zhang

Date: 15-07-2026

This research was supported by the China Scholarship Council (CSC) (grant number 202206320053), Nederlandse Organisatie voor Wetenschappelijk Onderzoek, Netherlands (grant number Nwa.1332.20.002), and the EU Horizon project AI Foundation Models in Agricultural Sciences (grant number 101293777).

Financial support from the Agricultural Biosystems Engineering Group (ABE) and the Artificial Intelligence Group (AIN) at Wageningen University for my PhD research, thesis preparation, and PhD defense is gratefully acknowledged.

Cover design by Chao Zhang and Liuyue Zhang.

Printed by proefschriftmaken.nl

