# Fish Prosject

by

Yusef Said, Christoffer Eigill Prag & Yngve Magnussen

in

IKT450 Neaural Network

Faculty of Technology and Science

Universitetet i Agder

Grimstad, November 2024

# Contents

# 1 Introduction

Fish species are vital to agriculture and biodiversity, as is the precision of identifying fish species for further examination for population management, biodiversity monitoring, and conservation [1], [2]. Convenient approaches, such as traditional morphometric techniques (TM) [1], are time-consuming and some of the most primitive methods, which rely on physical measurements, can be inefficient and harmful to fish species [1]–[3]. There are similar instances of labor-heavy and invasive procures, which can be harmful for fish species to go through [1], [2], [4].

With the advancement in Machine Learning (ML) models, particularly Siamese Neural Networks (SNNs), which offer an alternative solution to accurately identifying tasked objects. This project aims to address the aforementioned issues by exploring SNNs as a potential solution for developing an ML model that can serve as a tool for identifying various fish species.

In this report, the following objectives will be emphasized:

- To implement and design an ML model based on SNN, which will be trained by utilizing fish images as its data, to compare pairs of fish images, and based on the comparison determine their similarity.
- To implement an API that will enable users to both test and verify how the model would react to any random images, to accurately verify if the image is considered to be a fish or not, and how close the similarities the uploaded image is up against existing images from the data the ML model is trained on.

The system will be utilizing the data from the Fish4Knowledge repository [5], which will provide the necessary data needed to train the SNN, as displayed in 1.1 displaying a variety of different fish images. It contains a diverse set of labeled underwater images of fish species. This will aid the SNN to recognize and differentiate between images which based on the dataset determine whether they satisfy characteristics from the dataset if it are classified to be an image of a fish or not. By using a comparison method.

The report will explore the design, implementation, and evaluation of the system, while also reflecting on the issues and potential future improvements the ML model could have. The source code and additional resources for this project are available on GitHub [6] at: `https://github.com/yngvemag/uia-ikt450-fish-identification`.

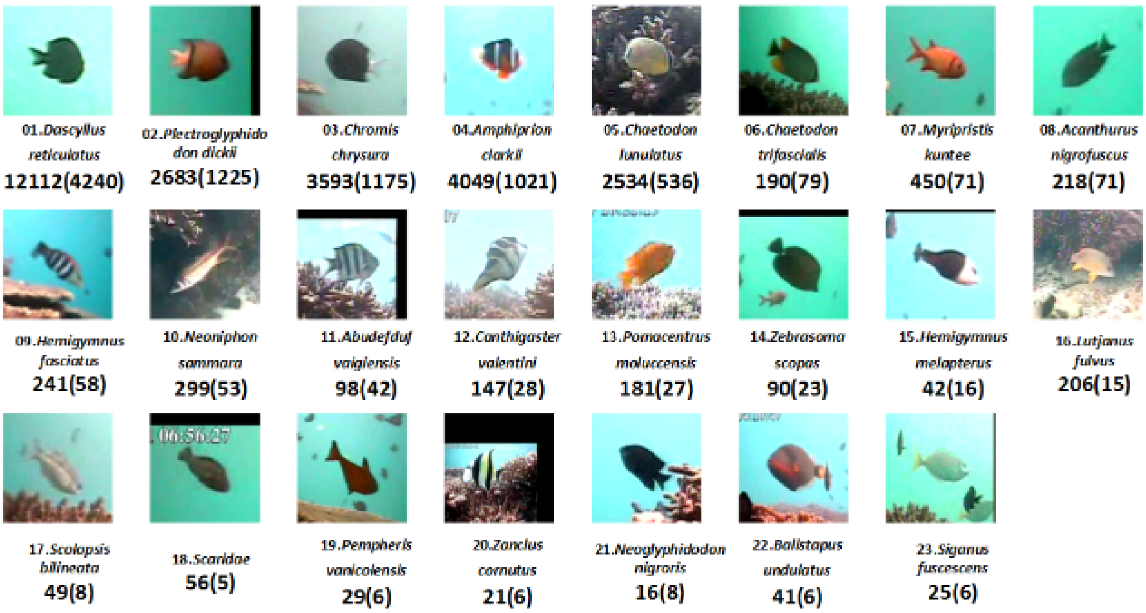



**Figure 1.1:** An overview of the underwater image data captured from Fish4Knowledge [7]. The dataset includes a variety of fish species, captured from different angles and under varying image qualities and noise levels. This diversity is essential for training the Siamese Neural Network (SNN).

# 2 Literature Review

In the following chapter, a comprehensive overview of the relevant literature and foundational concepts related to the study is provided. An overview of the SNNs, the Fish4Knowledge dataset, and a comparison with object detection methods such as Mask Region-based Convolutional Neural Networks (Mask R-CNN) and Single Shot MultiBox Detector (SSD) will be thoroughly explored. Methods such as Mask R-CNN and SSD perform well in object detection tasks but are less effective in tasks requiring fine-grained similarity learning. Since the objective of this project is to identify fish species based on their visual similarity, these limitations are addressed by leveraging SNNs, which are specifically designed to learn similarity metrics efficiently. The chapter will provide a closing conclusion with a review of relevant related work to contextualize the study.

## 2.1 Siamese Neural Network

SNNs are a type of neural network architecture that is primarily designed for similarity learning, this is often conducted where the network compares two images and provides a measurement based on distance. If the distance is equal to zero (0.00) then it's considered a 100% match, for instance, for a distance measurement where two images are compared, we anticipate the similarities to be close if it's an image of two dogs. However, we can anticipate some comparison differences if it's two different species.

The diagram in figure 2.1, displays, how the SNN model works, where it first needs to obtain a labeled dataset of images [8], for example, a set of datasets with images of dogs and cats. The dataset is then utilized for training the SNN model, once the training process is complete, the network is tested by feeding it images, and the network will determine whether it is a dog or a cat.

The architecture of SNN as illustrated in Figure 2.2, provides a simple design for how similarity learning by processing two input samples is conducted [8]. The sample inputs go through the neural network branches with shared weights. From the branches, feature embeddings for each sample input are extracted, which makes it possible that the same transformations are applied to both sample inputs. Furthermore, the result at the end of this is a comparison measurement of the two sample inputs translated into a distance metric (e.g., Euclidean distance). The metric determines the similarity between the two sample inputs.

**Loss Functions:**

In SNN, a function that plays a major role is the Loss Function. The loss function has the role of distinguishing similar and dissimilar pairs of images [8]. Two various types of loss functions are often

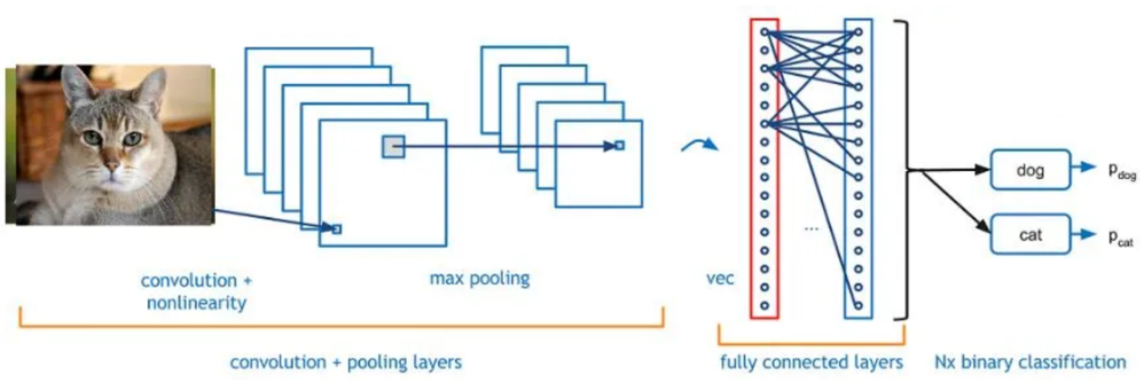



**Figure 2.1:** Overview of the SNN workflow [8]. The structure of the SNN for binary classification showcases the process of feature extraction through convolutional and pooling layers. It lastly followed by classification layers to determine and identify the input that belongs to a specific class (e.g., cat or dog). The expected output would be cat positive, and dog negative.

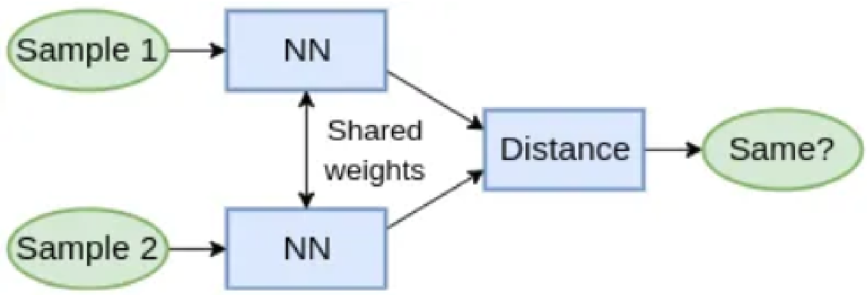



**Figure 2.2:** Overview of the SNN architecture [8]. It displays two input samples, and how they will be processed through identical subnetworks with shared weights to compare embeddings utilizing a distance function for measurements.

employed, the first one being a contrastive loss, and the second one being a triplet loss function. For the contrastive loss function, it evaluates the network's ability to differ between two pairs of images, this is conducted by minimizing the distance between the two pairs of images. Furthermore, it proceeds then with maximizing the two pairs of images if it is dissimilar, this is done by a margin threshold [8]. The second loss function: Triplet loss, operates on a triplet of images—anchor, positive, and negative—to refine similarity and dissimilarity measurements [8].

**Advantages:**

- Robust to class imbalance; works well with limited data [8].
- Excels in semantic similarity tasks; groups similar entities in embedding space [8].
- Generalizes effectively with minimal training data [8].

**Challanges:**

- High computational requirements due to pairwise/triplet comparisons [8].
- Lacks probabilistic outputs, limiting some applications [8].
- Fine-tuning hyperparameters (e.g., margin values) is critical but challenging [4].

## 2.2 Fish4Knowledge Dataset

The Fish4Knowledge (F4K), has gathered a vast collection of underwater footage of different fish species, resulting in a large-scale dataset collection. The collection was established to help researchers in marine biology and computer vision[5]. The Data collection offers a wide range of annotated underwater fish footage, which provides an ideal environment around the object as well, which is useful when creating and evaluating fish recognition algorithm systems.

Millions of fish examples from a variety of fish species have been included in the collection. The footage includes a variety of settings, which involve different conditions of the environment of the fishes as displayed in figure 2.3. Some of these involve lightning, occlusion, and water clarity. This has resulted in some of the footage being less clear on some occasions while others are rather more clear. In some instances, the footage can be harder to see while other times not. The variety of clarity is essential as it would be useful when training a detection model, which can adapt to a variety of different environments when analyzing different images.


**(a)** Fish image 1. **(b)** Fish image 2. **(c)** Fish image 3.

**Figure 2.3:** Examples of fish images being analyzed in the model from the F4K dataset [7].

The F4K dataset has many variety of features but one key feature in its vast data collection is its comprehensive annotation, which includes bounding boxes. These bounding boxes help identify in the images the focused object that needs to be focused on. It also includes species labels and metadata such as location and the date and time of the captured footage. Such annotations details which the collection includes, contribute that the dataset can be utilized for numerous tasks which helps researchers and students to use the dataset for both classification and object detection tasks. For this project, the data collection that is labeled images will be leveraged to train the SNN. To create pairs of images, and from the pairs, it can be possible to compare different fish species and determine their weight of similarity distances. Different pairs from different species will be labeled and considered as dissimilar fish species.

The challenge the dataset may offer is the fact that there is a significant class imbalance. This means that certain fish species are at times overrepresented while other fish species have limited footage samples. This imbalance can potentially affect the trained model, in its performance and level of accuracy. This would essentially mean the trained model, would be more suited for some fish species while not equally suited for other fish species. This can be considered to be a model which is biased in its predictions. Where it would essentially favor some fish species over others, that can be interpreted as some fish species will be more dominant than others classes. To address this issue, a solution such as data augmentation techniques can help normalize this problem, by rotation adjustments, flipping, and color adjustments could be applied during the preprocessing stages. To better improve the model's robustness and generalization, making it less biased and rather more fair in its determination of fish species it's trained on detecting based on the dataset collection.

By utilizing the F4K dataset, this project will benefit from a rich and diverse dataset that can aid in the development of a robust fish similarity detection system. While also addressing potential issues it may pose with real-world data.

## 2.3 Comparison with Other Methods

To better understand the effectiveness of the SNN approach for fish species detection, it's helpful to review and compare with other existing methods that are common, such as Mask Region-based Convolutional Neural Network (Mask R-CNN) [9] and Single Shot MultiBox Detector (SSD) Neural Network [10]. The design, function, and applicability of certain computer vision applications can differ across these approaches.

### 2.3.1 Mask R-CNN

Mask R-CNN is a well-liked framework for object detection, and instance segmentation [9]. It's built on top of Faster R-CNN, and it incorporates a parallel branch for object mask prediction, which enables both bounding box detection as well as pixel-level segmentation. Mask R-CNN is capable of locating numerous fish at once in a single footage. It provides for each fish-detected object a class labeled for fish species identification [9]. A visual illustration of how Mask R-CNN result is displayed in figure 2.4.

- **Advantages:**
  - It is effective for recognizing objects in crowded or overlapping environments since it provides both segmentation masks and bounding boxes.
  - It can detect many species in single footage, which might be an eventual requirement for this project.

- **Challenges:**

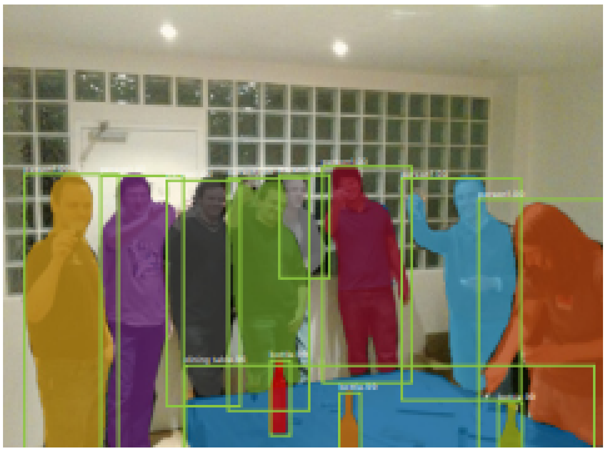


**Figure 2.4:** Visual representation of Mask R-CNN, displaying object detected of people captured in green bounding boxes, and segmentation masks, which is displayed by categorizing each object detected into separate colors to aid in differentiating the objects detected [9].

  - high computational responsibility, leading to considerable training and inference resources.
  - The quality of the annotated data collection has an essential effect on the performance, and inaccuracies in object detection could transfer to more complicated tasks like classification.

### 2.3.2 SSD Neural Network

SSD is an efficient object detection framework, it simultaneously identifies objects and predicts what class the object is most likely labeled as, making it a quick and effective model for object detection [10]. Since SSD is a single-stage detector, compared to Mask R-CNN, the processing for SSD is faster and doesn't require as much processing power. A visual illustration of how SSD results is displayed in figure 2.5.

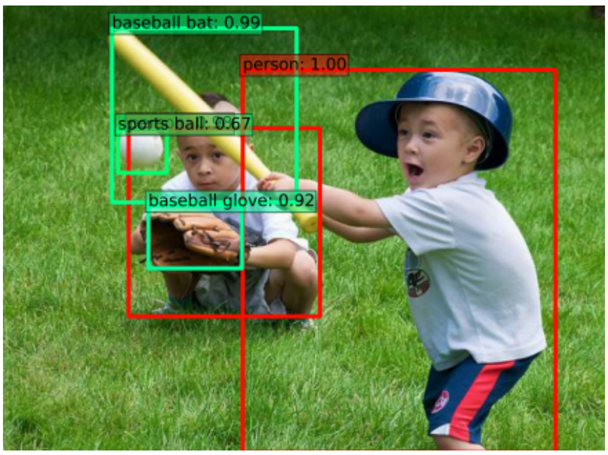



**Figure 2.5:** Visual representation of SSD capturing different objects from an image, and making predictions as well as naming the objects based on it's detection. Bounding boxes is also displayed to capture the intended object it has detected [10].

- **Advantages:**

- It is faster than two-stage approaches such as Mask R-CNN, which makes it well-suited for real-time applications.
- It has a simple design that offers a balance between computing efficiency and precision.

- **Challenges:**
  - It is compared to Mask R-CNN less accurate at detecting small or overlapping objects.
  - It is less efficient for fine-grained tasks, that may require a keen eye and precision, such as detecting fish species.

**SNN in Comparison** By emphasizing on training metrics for similarity instead of directly classifying or detecting objects, the SNN approach proposes a significantly different method. Based on this, the SNN works effectively for tasks like one-shot learning and pairwise comparison for similarity purposes of different species.

- **Advantages:**
  - As a result it learns a similarity metric, instead of depending on distinct class labels, it is robust to class imbalance.
  - It can effectively generalize to classes that haven't been encountered with minimal retraining.

- **Challenges:**
  - It does not allow multi-object detection, requiring further preprocessing or auxiliary models such as Mask R-CNN are needed.
  - It can be computationally demanding throughout training, therefore necessary to test with smaller datasets before doing wrong with larger datasets and pairwise comparisons.

Mask R-CNN and SSD excel in object detection, however, they struggle with fine-grained visual similarity, in comparison to SNN. This makes the aforementioned approaches less ideal for this project, while SNN is better suited. SNN is specifically designed for learning embeddings, which makes it a more appropriate method for fish species identification. However, future integration of Mask R-CNN with SNN, could improve the system and expand its potential limitations.

# 3 Implementation

The main goal of this project is to classify fish species with SNN. This type of network is designed to identify similarities between pairs of images. We have divided this project workflow into three main stages. Stage one is data preparation, stage two is model training and the final step is model evaluation.

## 3.1 Data Preparation

This section will outline the preparation of the dataset utilized for training and evaluating the SNN. The section will include details on how the dataset is organized, and loaded, and the creation of image pairs for training, testing, and reference.

### 3.1.1 Dataset Overview

The dataset used for this project is sourced from the Fish4Knowledge repository, a comprehensive collection of underwater fish images acquired from live video streams. The dataset includes 27,370 manually labeled fish images organized into 23 distinct clusters, with each cluster representing a unique fish species. The grouping is based on morphological characteristics such as the presence or absence of specific fins, shapes, or other anatomical features, as defined by marine biologists.

To facilitate training and evaluation of the SNN, the dataset was downloaded and organized as displayed in Figure 3.1. We selected all 23 fish species for the project. Each species was extracted into separate directories, maintaining the original structure. The dataset is imbalanced with one species having 12,112 images, while the least frequent species (e.g., Neoglyphidodon nigroris) has only 16 images.

### 3.1.2 Dataset Loading

The dataset consists of images grouped into subfolders, with each subfolder representing a specific fish species. Images within a subfolder belong to the same class. In the preparation phase, a fish dataset with both training and test data is included before training. The PyTorch DataLoader is used to efficiently load and manage these datasets in batches, enabling scalable training and evaluation.

Images are loaded into memory in pairs. Pair of fish within the same class that are labeled as 1 and pair of fish from different classes are labeled as 0 (not similar) 3.2. The DataLoader ensures that these pairs are shuffled and batched correctly during training, maintaining an even distribution of positive and

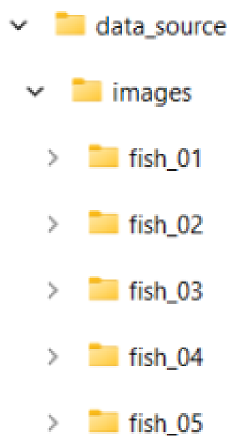



**Figure 3.1:** Overview of the data_source folder.

negative pairs for balanced learning.

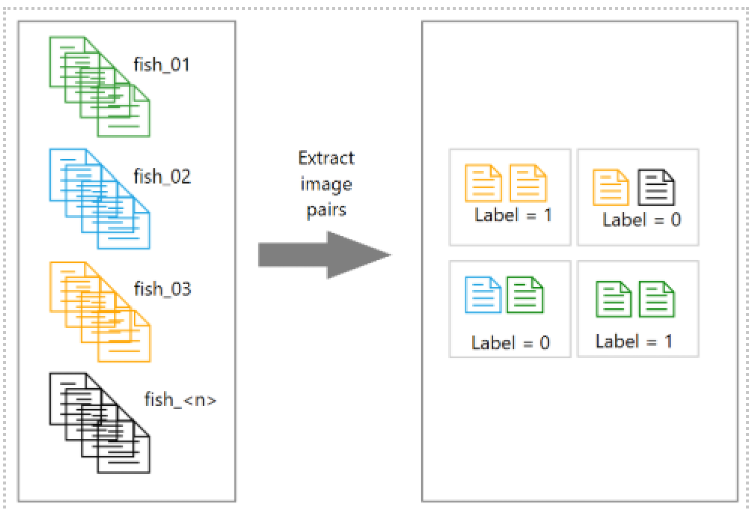



**Figure 3.2:** Extract image pair

The number of image pairs that will be used for training and test data is configurable with a setup parameter. If this parameter is set to 200 there will be created 200 pairs from each class that will have the label as 1, and also 200 pair of image pairs from different classes that will be labeled as 0.

In addition to the training and test datasets, we also extract a reference dataset. This dataset contains one representative image for each fish species. The reference dataset is used during testing to compare a query image (e.g., a fish image that the model has not seen before) against the reference images.

## 3.2 Model training

For the following section, a detailed overview of the architecture of the SNN will be illustrated in figure 3.3. How SNN is utilized for feature extraction, is the process of training the model through the usage of image pairs. Additionally, the evaluation methodology to measure the performance will be further outlined. Key aspects include the network structure, loss function, and accuracy metrics.

### 3.2.1 Design and Architecture of the SNN

A SNN with a convolutional backbone is used for feature extraction. The network computes similarity scores between image pairs. The network has two main components, the convolutional backbone responsible for extraction of high-level representation from the input images and fully connected layers that map the high-dimensional output into lower-dimensional features. The output from these layers is in fixed size so it is easier to compare images. The convolutional network (CNN) consists of a convolutional layer with 3 input channels to handle RGB images, 128 output channels for feature maps, and a 3x3 size kernel. Further, it has an activation function (ReLU -rectified linear unit) to introduce non-linearity. The fully connected layer has i input size of 256 channels and an output size of 512. This layer also has the ReLU activation function for linearity and at last, a linear layer that reduces the final vector size to 256.

#### Training Process

The model is trained using pairs of images selected from the dataset, where each pair is associated with a similarity label. Label 1 i associated with an image pair that should be of similar species, and label 0 indicates that the image pair is of different species. The train loop goes through every image pair created and is passed to the SNN‘s shared convolution network for comparison. The network creates feature representation of the image and the Euclidean distance of these features is calculated. A low distance shows that the features match and therefore similar, and a high Euclidean distance indicates that these images are different.

The loss function plays a central role in the backpropagation process. The key role of the loss-function is to minimize distances for images in the same class and maximize distances for image pair from different classes. The training process is configured with the Contrastive Loss function, which is designed to handle tasks that involve learning the similarity between image pairs. To optimize weight and bias this model is currently set up with the Adam (Adaptive Moment Estimation) optimizer. Both the loss function and the optimizer can be changed for this training process by changing the input parameters. Figure 3.3 explains the model’s training process.

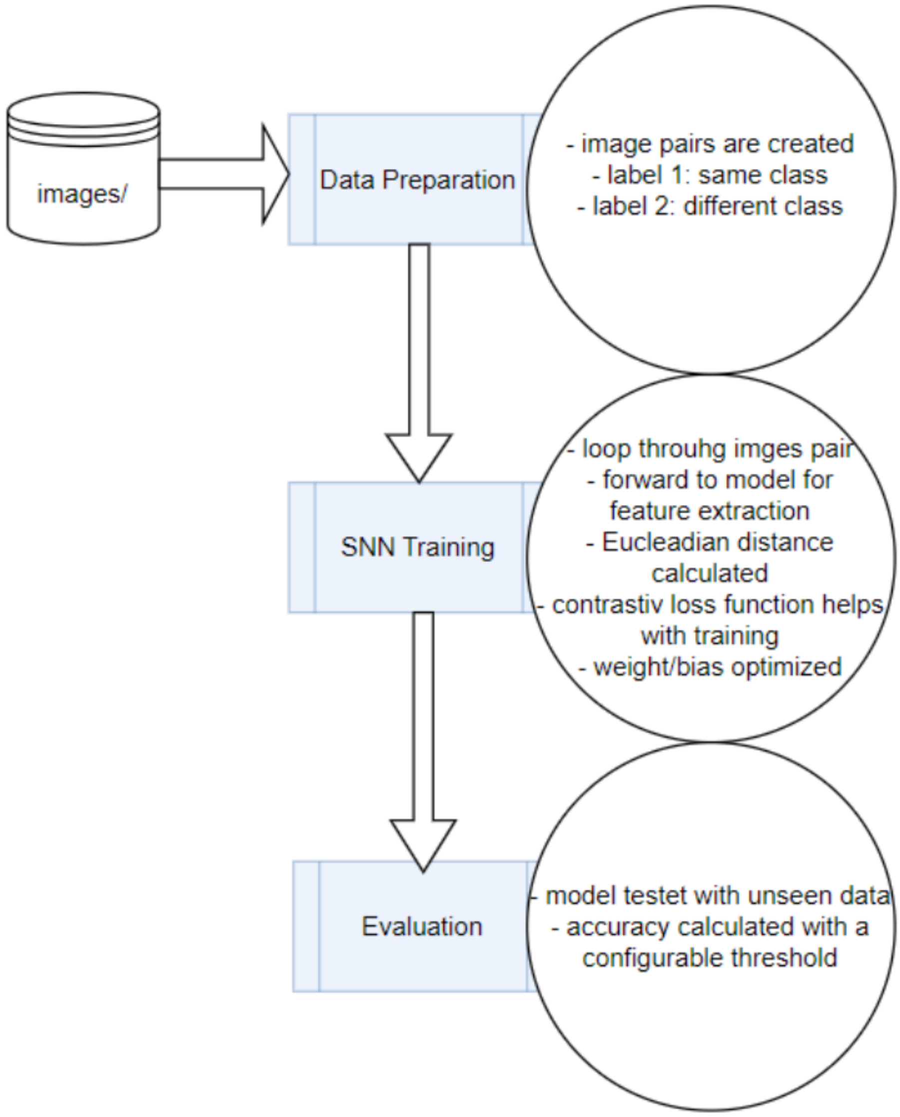



**Figure 3.3:** Model training

#### Evaluation

The trained SNN is evaluated using a test dataset that was not part of the training process. The test dataset contains image pairs with associated labels, where labels equal to 1 indicate similar species and labels equal 2 different species. The goal is to measure how good the model is at making predictions with unseen data. The accuracy metrics are then calculated with this formula

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \cdot 100$$

To predict that images are from the same class, the Euclidean distances of the features should be less than the configured threshold. If the distance exceeds the configured threshold, images are predicted to be from different species.

## 3.3 Web API Integration

**Flask API for Fish Image Analysis:** The Web API is designed to allow users to upload images for analysis. Flask, a lightweight and versatile web framework [11], is utilized to manage image uploads efficiently. Once the user has uploaded their desired images, the images are preprocessed through resizing and normalization to ensure compatibility with the model. The preprocessed images are then passed to a pre-trained SNN for similarity scoring or classification. The expected results are subsequently presented through a web interface that displays the analysis scores of the images.

**Tools and Key Features:**

- **Tools:** Flask for routing, PyTorch for model inference, and Pillow for image processing.
- **Features:**
  - Supports pairwise image comparisons.
  - Provides a secure mechanism for handling image uploads.
  - Includes an HTML-based interface for image submission and result visualization.

## 3.4 Project Architecture

The project is structured with a modular and organized design to ensure maintainability, scalability, and ease of collaboration. To provide clarity and separation of concerns each directory and file handles a specific functionality. Below is an explanation of the roles of the key components. Visual illustration of the source folder src/ is displayed in figure 3.4.

- **constants.py:** Centralized configuration file to manage constants such as hyperparameters, paths, and thresholds.
- **main.py:** The entry point for the training and evaluation workflow integrates all modules to load.
- **webapp/:** Contains the API implementation with a homepage interface for where random images can be uploaded and analyzed after SNN training is complete.
- **data/:** Contains modules responsible for handling datasets, including data preparation and loading.
- **models/:** Networks used with this project will be implemented in this folder.
- **training/:** Contains script for both training and evaluation.
- **transform/:** Implementation of transformations for preprocessing images before feeding them into the network will be located here.

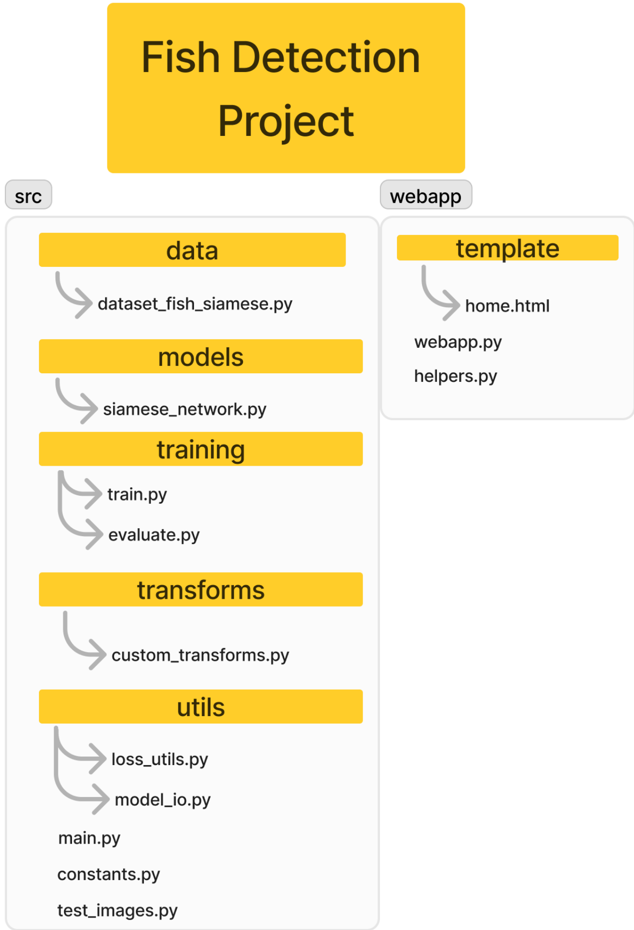



**Figure 3.4:** Overview over the src/ and webapp folder structure which contains the core files for the Fish Detection project.

- **utils/:** A collection of utility scripts that provide helper functions for different stages of the workflow.

# 4 Results & Discussion

The following chapter analyzes the SNN-based fish detection systems performance, focusing especially on its accuracy results, loss, scalability, and usability. The chapter will also address challenges with the project implementation, and emphasize the web application for fish species analysis and comparison results.

## 4.1 Results

### 4.1.1 Accuracy

**Pros:** The model demonstrated near-perfect accuracy (99.72%) during training and a high testing accuracy of 98.81%, verifying its generalizability. The minimal variation between training and testing accuracy suggests that the model does not overfit and can potentially perform well on real-world data.

**Cons:** While the testing accuracy is high, further testing on more diverse and unseen datasets is needed to confirm its robustness in practical scenarios. The high accuracy during training may still obscure subtle inter-class similarities, which could be challenging in edge cases.

### 4.1.2 Loss Values

**Advantages:** The network successfully learned meaningful feature representations, as indicated by consistently low loss values during training and evaluation. This consistency highlights the efficiency of the training process and the suitability of the Siamese Network for similar learning tasks.

**Challenges:** Despite low loss values, the fixed loss threshold may not adapt well to edge cases where inter-class similarity is high. This could lead to potential misclassifications in cases with subtle visual differences.

### 4.1.3 Scalability and Efficiency

**Advantages:** The model's architecture works well for single-species similarity detection, demonstrating scalability for other datasets.

**Challenges:** Training time is a significant drawback, requiring approximately 14 hours to complete. Optimizations, such as adaptive pair formation or smaller batch sizes, could help reduce computational overhead. Multi-species detection or generalization to new fish species has not been thoroughly tested, necessitating further research.

### 4.1.4 Usability

**Advantages:** The current implementation is effective for its intended purpose and demonstrates usability in fish species comparison tasks.

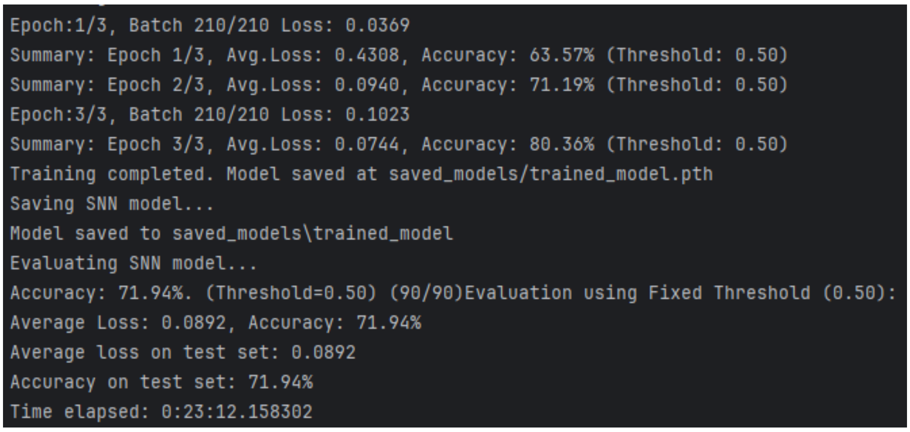
**Challenges:** Usability can be enhanced by integrating additional features, such as combining object detection (e.g., Mask R-CNN) with SNN for multi-object classification. Dynamic thresholds for similarity detection could improve accuracy for edge cases, where fixed thresholds (e.g., 0.30) may not suffice.

### 4.1.5 Web Application

The web application is built using Flask API [11], providing an interactive platform for users to upload images for analysis by the SNN system. As demonstrated in Figure 4.1, the application processes the uploaded images, computes similarity scores, and visualizes the results. The similarity is quantified using a distance measure: smaller distances (close to zero) indicate high similarity, while larger distances signify dissimilarity. To enhance user understanding, the analysis results are accompanied by a percentage score and color-coded indicators. Green signifies high similarity (low distance), orange denotes uncertain similarity, and red highlights significant dissimilarity (high distance).

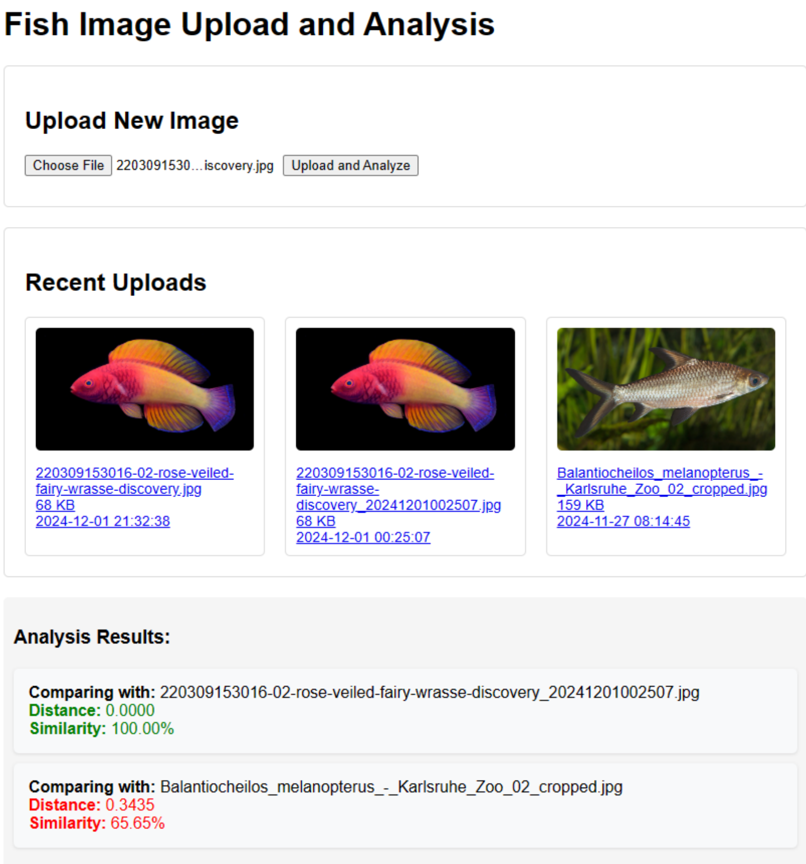
Figure 4.2a illustrates an example where non-fish images (e.g., a car and calculator) are analyzed, producing high dissimilarity scores, as the model is not trained for such themes. In contrast, Figure 4.2b presents results for fish species images, where the model effectively recognizes similarities and provides clear matches based on its training.

This web application bridges research and usability, enabling real-time testing of diverse images and challenging the SNN model's capabilities. It serves as a foundation for exploring alternative methods for fish species identification while ensuring scalability and accessibility for future improvements and applications.



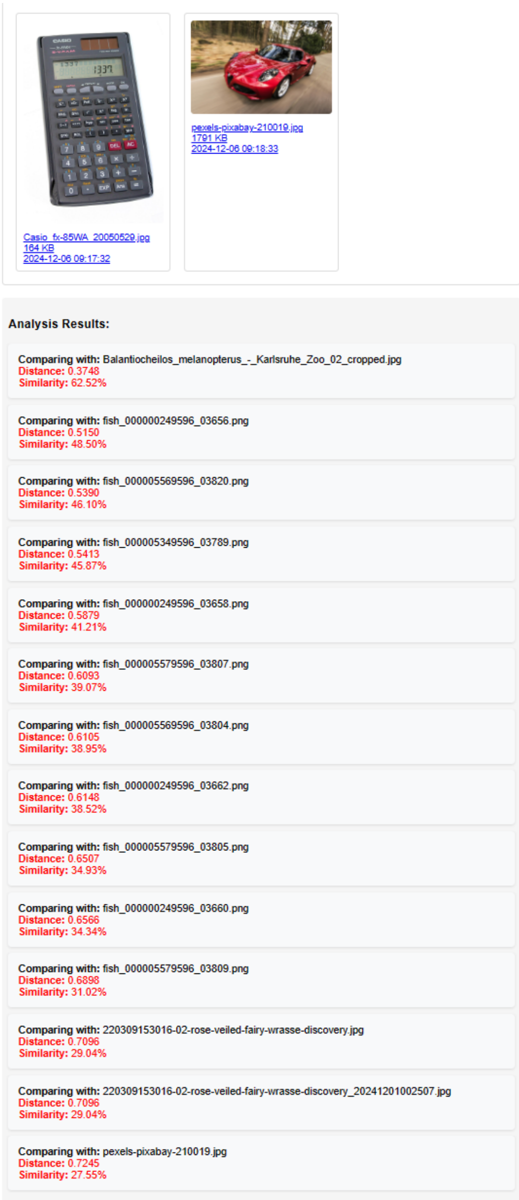

**(a)** SNN training complete, for a small dataset of F4K.


**(b)** Landing page interface, users are given the option to upload images, by clicking the "Choose File", then "Upload and Analyze".

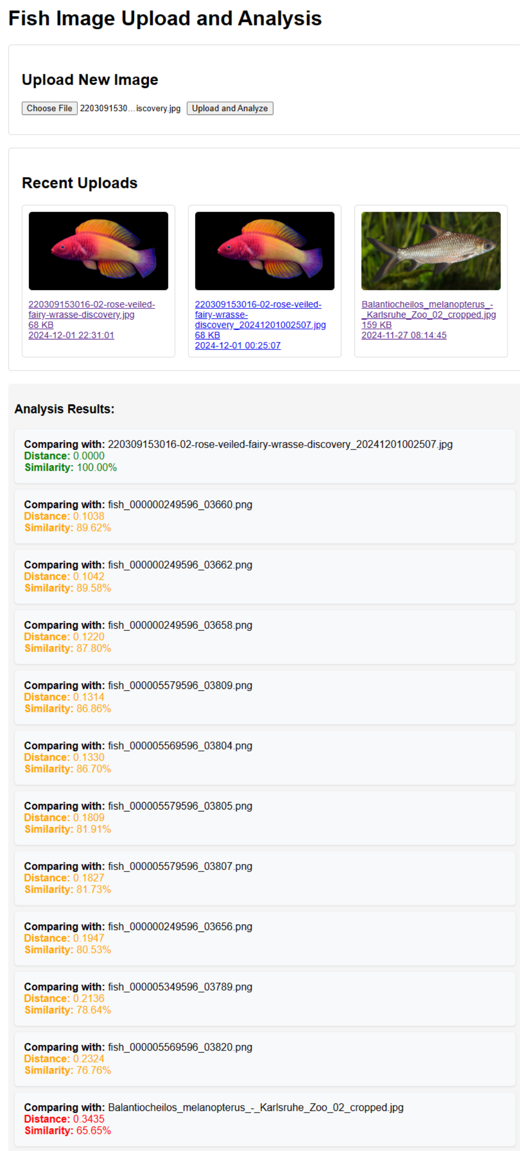



**(c)** Overview of the interface when random fish images are uploaded and analyzed [12], [13]. Analysis Results compare the images uploaded.

**Figure 4.1:** Overview of the Web Application: After SNN training, the API analyzes input images, providing distance and similarity measurements. Green indicates high similarity, while red represents low similarity.




**(a)** Result for Car and Calculator image [14], [15].




**(b)** Result for two different fish species images [12], [13].

**Figure 4.2:** Overview of two different test results, one with a different theme which the SNN model is not trained for in figure 4.2a, while the other test is for a recognizable theme which the SNN model is trained for in figure 4.2b. The analysis result indicates a clear message that it finds better match results.

## 4.2 Initial Journey of Developing the Fish Prediction System

The fish prediction system's original design focused on utilizing object detection and classification by combining custom networks with pre-trained models. The goal was to employ a modular approach to effectively identify and classify fish species. However, the challenges faced and the lessons learned during the process led to significant design changes that resulted in the current approach. Below is a summary of the early development and evolution of the system design.

Our first method was using a Siamese Neural Network in conjunction with a pre-trained Mask R-CNN model (ResNet-50) for Object Detection to identify fish in photos. Bounding boxes of the objects could be recognized by the model, and they could be linked to class labels for each fish type. The output of Mask R-CNN—that is, the identified regions of interest and their characteristics was fed straight into a Siamese neural network for training. Fish species were to be classified using image pair similarities, and the SNN was to be trained using the regions identified by the Mask R-CNN. We discovered very soon that dependency was introduced when the SNN was trained using the output of another network (Mask R-CNN). If the object detection model underperformed or made errors, these propagated into the SNN, degrading its performance. Although the combined approach was successfully implemented, the SNN's classification accuracy was not optimal. It was challenging to obtain significant results in the SNN due to the reliance on Mask R-CNN's performance.

We chose to switch to Direct SNN Training. The original approach's shortcomings were addressed by restructuring the system design. We started to train the SNN directly from image pairs rather than Mask R-CNN outputs. We labeled each image pair that comes from the same species with label 1 and different species with label 0. The SNN now functions without relying on any object detection model, which strengthens the system. By training directly with labeled image pairs, the network was able to acquire feature representations without being limited by the object detection model's possible errors. Figure 4.3 illustrates the initial workflow which included Mask R-CNN in the earlier stages of the process before utilizing the SNN training.

The next step will be to reintroduce object detection into the workflow using a more sophisticated approach, even though the current SNN design works well for classifying fish. The Mask R-CNN can be used as a tool to recognize and extract objects (fish) from an image and then feed the output into a pre-trained SNN to classify the detected objects. This will make it possible to identify and categorize several fish in a single image and also classify fish in an image that is not centered. Visual illustration for the SNN and Mask R-CNN side by side for clarity purposes is displayed in figure 4.4, showcasing how the architecture of the two.

The final dataflow as illustrated in figure 4.5, showcases a future improved version where both Mask R-CNN and SNN are combined, the architecture is a proposed solution for a future work as a potential improvement on the current architecture of the Fish Species Detection which is only based on SNN.

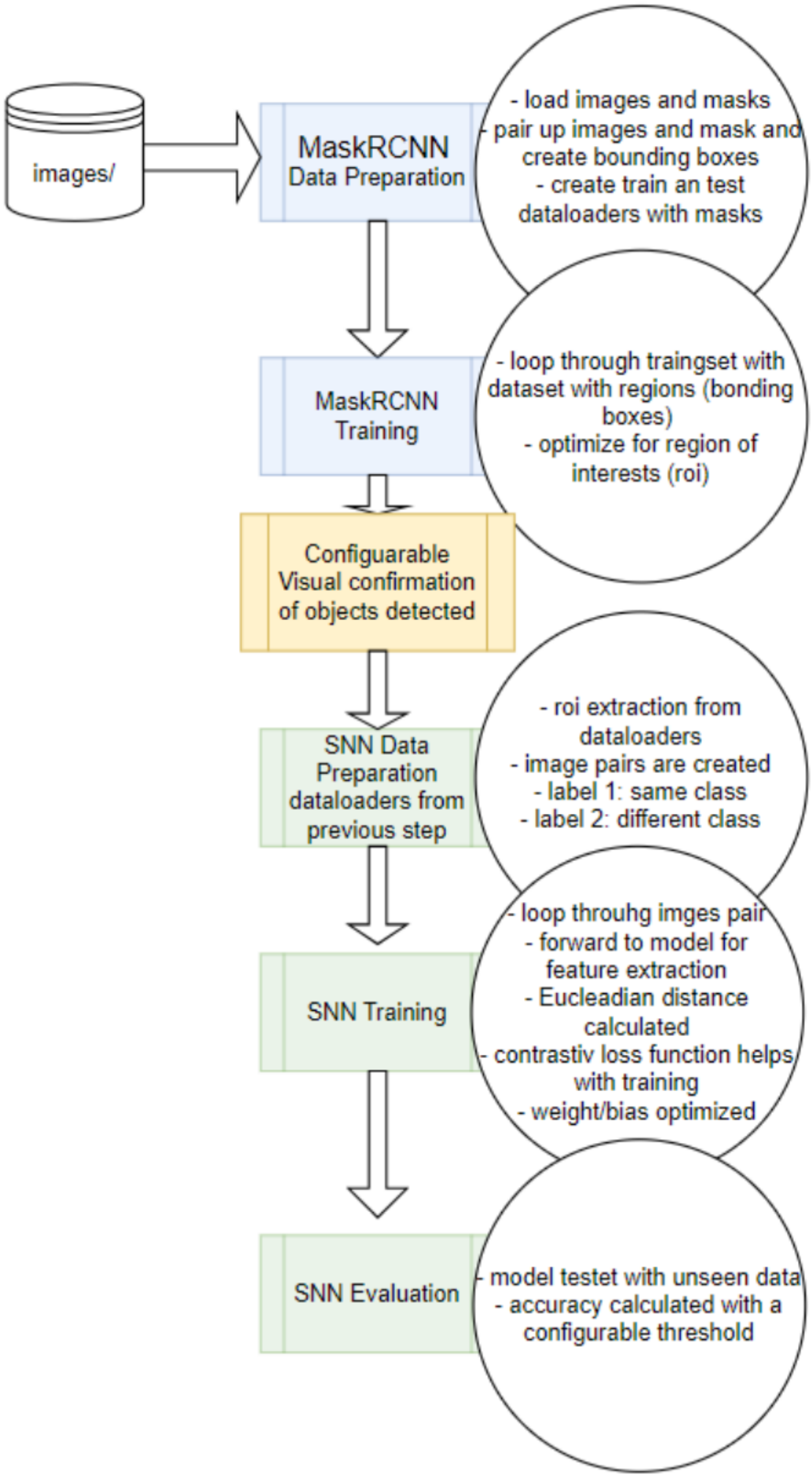



**Figure 4.3:** Overview of the initial workflow for the training model, which displays how each stage of the data preparation and training phases are conducted before the final SNN evaluation.

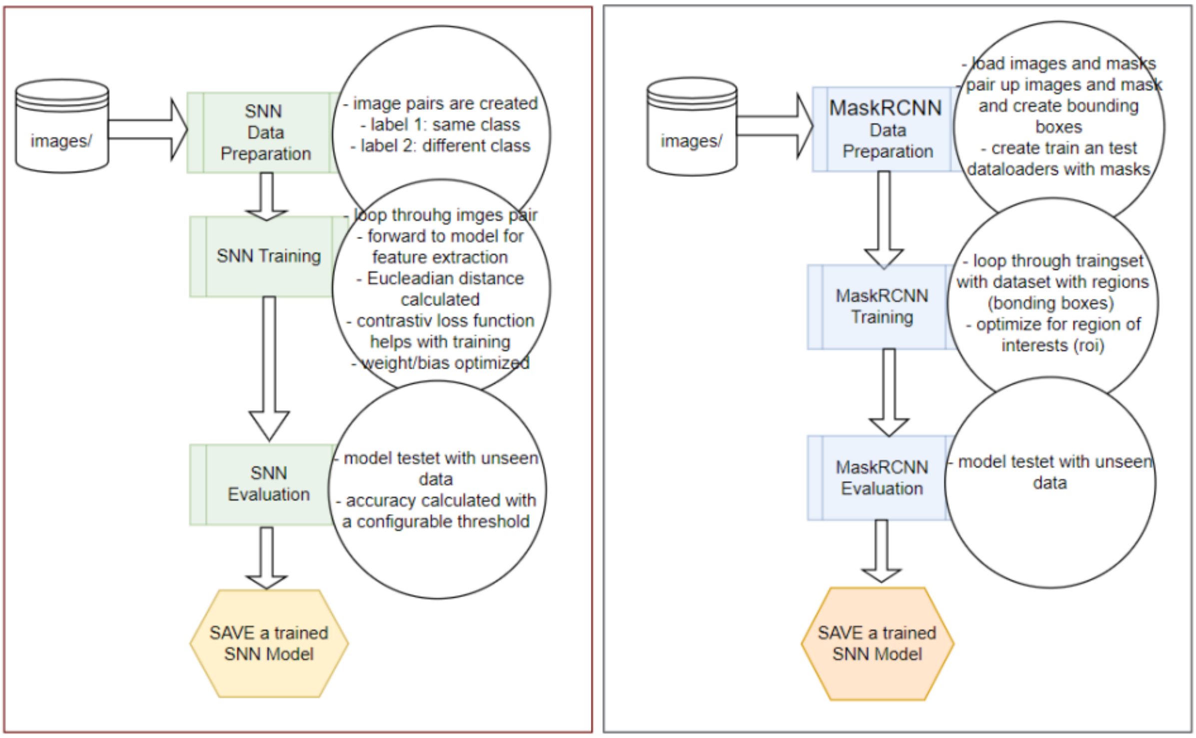



**Figure 4.4:** Overview of the two models, SNN and Mask R-CNN, showcasing the dataflow for each stage of the process before final evaluation.

## 4.3 Challenges

The dataset has a significant class imbalance. The most frequent species has nearly 1,000 times more images than the least frequent species. This imbalance creates challenges during training, as the model may become biased toward the more frequent species, reducing its ability to generalize to underrepresented classes.

To avoid this we added a pair count, a fixed number of pairs per class, restriction to balance the dataset and ensure computational efficiency while avoiding overfitting to dominant species. Also, random sampling was applied to make sure that less frequent species were represented in both training and testing sets.

Images used for testing and training were captured in a range of real-world situations. Variations in lighting can affect the quality of images, and variations in the position of the fish can also affect the similarity calculation done by the SNN network. The classification task is made easier by the fact that fish are mostly centered in the images in the current dataset. The current setup might not work well if fish appear partially outside the frame or off-center. Therefore the next step is to do object detection integration. Before feeding fish regions into the SNN, an object detection model (such as Mask R-CNN) is used as a pre-processing step to identify and crop them. By using this method, the model would be able to identify several fish in a single picture, even if they are partially obscured or not centered.

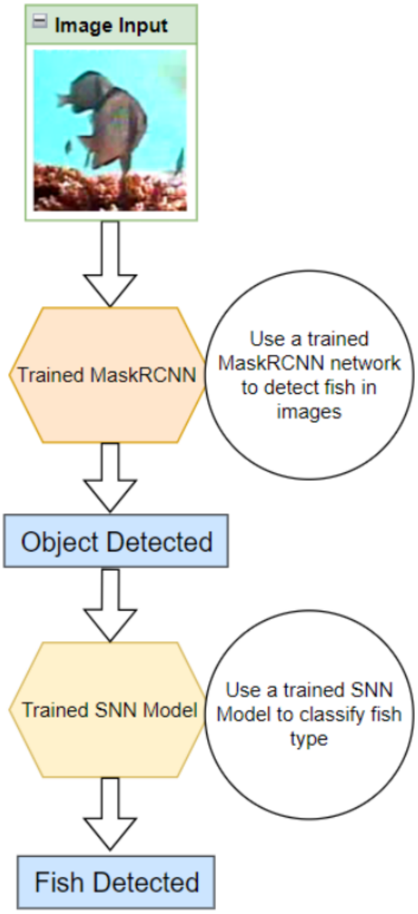



**Figure 4.5:** Overview of the final dataflow as a future improvement on the architecture which incorporates Mask R-CNN and SNN models for object detection. The process from where images from F4K dataset is utilized [7] to train the model.

The web application wasn't fully completed and implemented due to more time where invested in developing the initial architecture of having a combination of Mask R-CNN and SNN work together with the model training. The Web application was, to begin with not what the team had its main focus on as the detection implementation was the main focus. Due to last-resort changes, and not enough time to test the web application further, but rather a test on a smaller dataset with lower accuracy was the only time left to be conducted. The web application has its usage however it would not be functional without having the SNN model first set in stone.

The web application was not fully implemented and tested due to time constraints, as the majority of resources were dedicated to developing the initial architecture that combined Mask R-CNN with SNN for model training. However, since the initial architecture design proved to present low accuracy results, the team's focus shifted to a primary focus on improving the accuracy and performance of the SNN model, which further delayed and made it difficult to work on the web application. Implementing and optimizing the detection system, was the core focus of the project, with the web application being a secondary priority. Consequently, only limited testing was conducted using a smaller dataset with reduced accuracy as in the results from the training model shown in Figure 4.1a. Although the web application demonstrates potential usability, it relies heavily on a fully trained and functional ML model such as SNN, which was prioritized over web interface enhancements due to time limitations.

# 5 Conclusions

The primary objective of the Fish Detection project was to design and implement a system based on a Siamese Neural Network (SNN) to classify images using similarity learning. The SNN achieved near-perfect performance, with a training accuracy of 99.72% and a testing accuracy of 99.81%. These results demonstrate the model's ability to generalize effectively within the Fish4Knowledge (F4K) dataset. The high accuracy highlights the SNN's capability to differentiate fish species with precision, thereby meeting the project's core objectives.

The integration of a web application added a practical interface for testing and analysis, allowing image uploads for real-world evaluation. Although the web application was not fully developed or tested on larger datasets due to time constraints, initial tests on a smaller dataset (using a pre-trained SNN) showed an accuracy of 71.94%. These results indicate that the web application was functional, even with limited training data, and provided a proof of concept for real-world usability.

Several challenges were encountered during the project. Dataset imbalance led to potential biases in prediction results, and high computational demands made model training time-intensive, sometimes exceeding 14 hours. Despite these challenges, the project successfully created a functional SNN-based detection system.

For future work, the project would benefit from incorporating a more diverse dataset to reduce biases and improve model accuracy. Additionally, enhancing the web application's functionality and expanding its testing scope would further demonstrate the SNN's potential for fish species detection in practical applications.

# References


[1] A. K. Dwivedi and K. De, "Role of morphometrics in fish diversity assessment: Status, challenges and future prospects," *National Academy Science Letters*, vol. 47, no. 2, pp. 123–126, Mar. 2024. DOI: 10.1007/s40009-023-01323-x. [Online]. Available: https://doi.org/10.1007/s40009-023-01323-x.

[2] V. Allken, N. O. Handegard, S. Rosen, T. Schreyeck, T. Mahiout, and K. Malde, "Fish species identification using a convolutional neural network trained on synthetic data," *ICES Journal of Marine Science*, vol. 76, no. 1, pp. 342–349, 2019. DOI: 10.1093/icesjms/fsy147.

[3] M. Axelsson, E. Peterson, A. Gräns, and P. E. Ljung, "Fish identification, marking, and tagging methods," Swedish University of Agricultural Sciences, 2020, Accessed on 12/06/2024. [Online]. Available: https://jordbruksverket.se/download/18.439d53c71803f95f84911e2a/1701887752470/Fish-identification-marking-and-tagging-methods-tga.pdf.

[4] L. S. Lemos, L. M. Angarica, R. A. Hauser-Davis, and N. Quinete, "Cortisol as a stress indicator in fish: Sampling methods, analytical techniques, and organic pollutant exposure assessments," *International Journal of Environmental Research and Public Health*, vol. 20, no. 6237, 2023, Accessed on 12/06/2024. DOI: 10.3390/ijerph20136237. [Online]. Available: https://doi.org/10.3390/ijerph20136237.

[5] R. B. Fisher, *Fish4knowledge homepage*, https://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/, Accessed on 12/06/2024, 2024.

[6] Y. S. Yngve Magnussen and C. E. Prag, *Fish detection project repository*, https://github.com/yngvemag/uia-ikt450-fish-identification, Accessed: December 7, 2024, 2024.

[7] R. B. F. Phoenix X. Huang Bastiaan B. Boom, *Fish recognition ground-truth data*, Available at https://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/GROUNDTRUTH/RECOG/, Accessed on 12/04/2024, 2024.

[8] R. Nag, *A comprehensive guide to siamese neural networks*, https://medium.com/@rinkinag24/a-comprehensive-guide-to-siamese-neural-networks-3358658c0513, Accessed on 12/06/2024, Nov. 2022.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *arXiv preprint arXiv:1703.06870*, 2018, Accessed on 12/06/2024. [Online]. Available: https://arxiv.org/pdf/1703.06870.

[10] W. Liu, D. Anguelov, D. Erhan, *et al.*, "Ssd: Single shot multibox detector," *arXiv preprint arXiv:1512.02325*, 2016, Accessed on 12/06/2024. [Online]. Available: https://arxiv.org/pdf/1512.02325.

[11] P. Projects, *Welcome to flask — flask documentation (3.1.x)*, https://flask.palletsprojects.com/en/stable/, Accessed on 12/06/2024, 2024.

[12] W. Contributors, *Balantiocheilos melanopterus - karlsruhe zoo 02 - sølvhai*, https://no.wikipedia.org/wiki/S%C3%B8lvhai#/media/Fil:Balantiocheilos_melanopterus_-_Karlsruhe_Zoo_02.jpg, Accessed on 12/06/2024, 2024.

[13] C. Staff, *Newly discovered rainbow-colored fish lives in the ocean's 'twilight zone'*, https://edition.cnn.com/2022/03/10/world/rose-fairy-wrasse-rainbow-fish-scn/index.html, Accessed on 12/06/2024, 2022.

[14] P. Contributors, *Ferrari california, lotus elise, alfa romeo 8c competizione - image on pxhere*, https://pxhere.com/no/photo/453672, Accessed on 12/06/2024, 2024.

[15] W. Contributors, *File: Casio fx-85wa 20050529.jpg*, https://no.m.wikipedia.org/wiki/Fil:Casio_fx-85WA_20050529.jpg, Accessed on 12/06/2024, 2024.