



# Toward phytoplankton parasite detection using autoencoders

Simon Bilik<sup>1,2</sup> · Daniel Batrakanov<sup>1</sup> · Tuomas Eerola<sup>1</sup> · Lumi Haraguchi<sup>3</sup> · Kaisa Kraft<sup>3</sup> · Silke Van den Wyngaert<sup>4</sup> · Jonna Kangas<sup>4</sup> · Conny Sjöqvist<sup>5</sup> · Karin Madsen<sup>5</sup> · Lasse Lensu<sup>1</sup> · Heikki Kälviäinen<sup>1</sup> · Karel Horak<sup>2</sup>

Received: 15 March 2023 / Revised: 16 June 2023 / Accepted: 14 August 2023  
© The Author(s) 2023

## Abstract

Phytoplankton parasites are largely understudied microbial components with a potentially significant ecological influence on phytoplankton bloom dynamics. To better understand the impact of phytoplankton parasites, improved detection methods are needed to integrate phytoplankton parasite interactions into monitoring of aquatic ecosystems. Automated imaging devices commonly produce vast amounts of phytoplankton image data, but the occurrence of anomalous phytoplankton data in such datasets is rare. Thus, we propose an unsupervised anomaly detection system based on the similarity between the original and autoencoder-reconstructed samples. With this approach, we were able to reach an overall F1 score of 0.75 in nine phytoplankton species, which could be further improved by species-specific fine-tuning. The proposed unsupervised approach was further compared with the supervised Faster R-CNN-based object detector. Using this supervised approach and the model trained on plankton species and anomalies, we were able to reach a highest F1 score of 0.86. However, the unsupervised approach is expected to be more universal as it can also detect unknown anomalies and it does not require any annotated anomalous data that may not always be available in sufficient quantities. Although other studies have dealt with plankton anomaly detection in terms of non-plankton particles or air bubble detection, our paper is, according to our best knowledge, the first that focuses on automated anomaly detection considering putative phytoplankton parasites or infections.

**Keywords** Phytoplankton anomalies · Phytoplankton parasites · Anomaly detection · Autoencoders · Object detection · Faster R-CNN

---

S. Bilik and D. Batrakanov have contributed equally to this work.

✉ Simon Bilik  
bilik@vut.cz

Daniel Batrakanov  
daniel.batrakanov@lut.fi

Tuomas Eerola  
tuomas.eerola@lut.fi

Lumi Haraguchi  
lumi.haraguchi@syke.fi

Kaisa Kraft  
kaisa.kraft@syke.fi

Silke Van den Wyngaert  
silke.vandenwyngaert@utu.fi

Jonna Kangas  
jonna.e.kangas@utu.fi

Conny Sjöqvist  
conny.sjoqvist@abo.fi

Karin Madsen  
karin.madsen@abo.fi

Lasse Lensu  
lasse.lensu@lut.fi

Heikki Kälviäinen  
heikki.kalviainen@lut.fi

Karel Horak  
horak@vut.cz

- 1 Computer Vision and Pattern Recognition Laboratory, Department of Computational Engineering, Lappeenranta-Lahti University of Technology LUT, Yliopistonkatu 34, 53850 Lappeenranta, Finland
- 2 Department of Control and Instrumentation, Faculty of Electrical Engineering and Communication, Brno University of Technology, Technická 3058/10, 616 00 Brno, Czech Republic
- 3 Marine Ecology Measurements, Finnish Environment Institute, Agnes Sjöbergin Katu 2, 00790 Helsinki, Finland
- 4 Environmental and Marine Biology, Åbo Akademi University, Henrikinkatu 2, 20014, Turku, Finland
- 5 Environmental and Marine Biology, Åbo Akademi University, Henrikinkatu 2, 20014 Turku, Finland

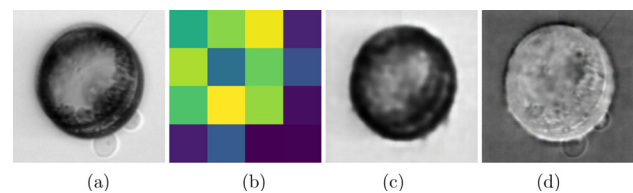
## 1 Introduction

Phytoplankton are key players in aquatic systems, where they mediate biogeochemical cycles and form the base of multiple food webs [1]. The dynamics of phytoplankton populations result from the interplay between resource availability and mortality losses [2]. While some loss mechanisms such as grazing are well known, the contribution of loss mechanisms like parasitism remains poorly considered and largely understudied in many aquatic systems. Phytoplankton are susceptible to a wide variety of parasites, such as viruses, bacteria, protists, and fungi. Such parasites can cause mortality of certain phytoplankton species, thereby altering the phytoplankton bloom dynamics and changing the cycling of matter and flow of energy in aquatic ecosystems [3–5].

Zoosporic or nanoflagellate parasites that infect phytoplankton comprise a highly diverse functional group of eukaryotic protist and fungal species [6]. They have in common the production of free-living motile stages as their infective propagules, which attach to a phytoplankton host cell and develop either inside (endobiotic) or outside (epibiotic) the host cell using host resources for their growth and reproduction. Due to their inconspicuous nature, phytoplankton parasites are difficult to identify, and objects that are difficult to identify typically tend to be overlooked or neglected. Consequently, although the presence and potential importance of these phytoplankton parasites are increasingly recognized, quantitative data of their occurrence in nature are extremely scarce.

An additional challenge to study of phytoplankton parasites is the need to capture rapid infection dynamics on a relevant temporal and spatial scale (e.g., days). Obtaining quantitative information about parasite infections using traditional methods is labor-intensive and time-consuming, which limits the spatial and/or temporal coverage of many studies investigating phytoplankton–parasite interactions [7].

Recent technological advances in imaging instruments have made it possible to collect large volumes of plankton image data for study of plankton populations, thus opening new research possibilities [8]. The possibility of high-frequency sampling enabled by imaging instruments can potentially result in better understanding of phytoplankton dynamics and their potential interactions with parasites [7]. However, while methods for automatic recognition of phytoplankton classes have been widely developed, methods for automatic recognition of phytoplankton parasite infections remain underdeveloped. The absence of an effective approach for parasitic infection recognition is likely associated with challenges related to obtaining sufficient volumes of image data of plankton parasites, which requires screening of huge amounts of raw image data. Such tasks are best addressed with automated solutions.



**Fig. 1** Anomalous sample of the Centrales plankton species: **a** Original, **b** encoded space, **c** reconstruction, and **d** difference image

The scarcity of plankton parasite images is a major challenge for the development of deep learning-based computer vision methods for parasite detection. While object detection methods such as Faster R-CNN [9] and YOLO [10] have been shown to achieve high accuracy on various detection tasks, including parasite detection (see, e.g., [11]), they struggle when the amount of training data is limited. Therefore, a more promising approach is to formulate parasite detection as an anomaly detection task. Here, the idea is to train the model with images of healthy plankton and detect images that deviate from the data on which the models were trained. Due to the availability of large amounts of plankton image data without parasites for training and relatively small intra-class variation among healthy samples, images that deviate notably from the training data can be expected to contain potential parasites.

This work investigates automated image-based phytoplankton parasite detection. The problem is formulated as an anomaly detection problem and solved using an autoencoder. The proposed method consists of a vector-quantized variational autoencoder (VQVAE) [12] that encodes the input image into a compressed latent representation and uses the compressed representation to reconstruct the original image. The rationale is that when the autoencoder is trained only on images of healthy phytoplankton, the autoencoder fails to reconstruct the parasites, which allows them to be detected from the difference image (see Fig. 1). The proposed method further employs the HardNet [13] feature extractor and Local Outlier Factor [14] to distinguish between healthy plankton and plankton with parasites.

In the experimental part of the work, an extensive set of different backbone convolutional neural networks (CNNs), autoencoder architectures, feature extractors, and classifiers are systematically evaluated on challenging phytoplankton image data to find the best combination and to demonstrate the performance of the proposed method. In addition, we compare the autoencoder-based anomaly detection method to a Faster R-CNN-based object detector. The results show that the proposed method achieves comparable accuracy to the state-of-the-art Faster R-CNN object detector while requiring no images with parasites for training. Consequently, the autoencoder-based method can be considered a promising approach for utilization in plankton image analysis where

the collection of large training data of plankton with parasites is infeasible.

The main contributions of our paper are the development of a novel anomaly detection framework and its application to phytoplankton parasite detection. High-frequency imaging data coupled to automatic presorting of potentially infected plankton allow to capture and quantify infection dynamics on relevant temporal and spatial scales. This is an essential step toward understanding the role of parasites in shaping phytoplankton community dynamics and ecosystem processes. The proposed framework is general and can be applied to other anomaly detection task such as industrial fault control.

## 2 Related work

Anomaly detection is a data classification technique in which a detector models the representation of samples within a specification (OK) and classifies all samples that deviate from the specification as anomalous (NOK). This problem can be challenging because of potentially high diversity within the NOK samples, imbalance between the number of samples in the OK and NOK groups, and irregularity of the NOK class. A comprehensive overview describing anomaly detection problems, techniques, and categorization is presented in [15].

First introduced for image data in [16], autoencoder (AE) models are now widely used in computer vision. The use of insufficient generalization ability on out-of-training data for an AE model with the aim of detecting anomalies in synthetic and real-world data, in the case studied, telemetry data, was first demonstrated in [17]. The results showed that such AEs can be used to detect previously unseen anomalous samples. The concept was further enhanced and used on image data in, for example, [18] and [19]. A comprehensive overview of AE techniques can be found in [20].

Plankton anomaly detection has been previously studied in the context of open-set recognition, i.e., image classification with the presence of previously unseen classes (plankton species). In [21], the authors presented an unsupervised approach to classify a plankton sample and detect potential significant differences (i.e., anomalies) with respect to the detected class. Image features were extracted using classical computer vision methods utilizing geometrical, moment-based, and other traditional features.

In [22], a CNN trained on OK samples and artificial NOK samples derived from the OK data by common data augmentation techniques such as blurring and noise addition was used as the feature extractor. An anomaly score was then computed from these features and used together with the trained feature extractor to distinguish between the OK plankton samples and anomalies. In the work, air bubbles and non-plankton water particles were considered as anomalies.

In [23], the authors used a parallel network of custom statistical classifiers called TailDeTect (TDT) to discover previously unseen plankton species. Each of the TDT classifiers was trained on one particular species, and a sample was considered as unknown if none of the classifiers was able to detect it. Unknown samples were collected and validated by experts. Feature extraction and the concept itself were based on work presented in [21].

In [24], open-set recognition plankton recognition was addressed using a similarity learning approach. Metric learning with angular margin loss was applied to obtain image embedding vectors that model the similarity between images. The anomalies (images from previously unseen classes) were detected by setting threshold values for the similarity.

Faster R-CNN [9] is a popular deep learning (DL) algorithm that has been successfully applied to various domains and tasks, including object detection and anomaly detection [25]. Anomaly detection using Faster R-CNN involves training the model on abnormal images to learn the features of abnormal instances. Then, during classification, the model is used to detect abnormal samples that deviate from the expected outcome. For instance, in industrial manufacturing, abnormal behavior can include machine malfunctions, while in medical diagnosis, it can take the form of unusual patterns in medical images.

An example of anomaly detection is presented in [25], where an improved Faster R-CNN was used to detect defects in steel plates. The algorithm was trained on a dataset of abnormal regions on steel plate images and was able to accurately detect anomalies such as cracks and holes in test images. Using a similar approach, a subtle modification of Faster R-CNN for detection of anomalies in CT images of lungs was considered in [26].

Object detection methods have also been successfully used for parasite detection. For example, in [11], where a YOLOv5 object detector is used to detect a parasitic mite on the body of a honey bee. An overview of other object detection techniques and commonly used datasets can be found, for example, in [27].

In plankton research, Faster R-CNN has been widely adopted for segmentation and object detection. Several object detection approaches, including Faster R-CNN, were utilized in [28] to evaluate a synthetically augmented dataset. Similar work is presented in [29], where a plankton dataset from a darkfield microscope was compiled and then tested with various object detection methods, including YOLOv3 [30], R-CNN [31], and SSD [32].

### 3 Proposed methods for phytoplankton anomaly detection

In this work investigating detection of phytoplankton samples with anomalies, we primarily employ an unsupervised autoencoder-based approach, followed by the use of different feature extractors and one-class classifiers. Supervised object detection based on the Faster R-CNN [9] is utilized to compare the results of our proposed method with a state-of-the-art approach.

#### 3.1 Autoencoder-based approach

The proposed method to detect anomalous plankton samples is constructed on the framework available in [33]. This implementation allows various combinations of different AE architectures without consideration of the convolutional layers (i.e., fully connected AE, variational AE and others) termed as AE cores, convolutional layers architectures, feature extractors, and one-class classifiers to be tested. In the approach used in this paper, we combined five AE cores, six convolutional encoders and decoders, six feature extractors, and four classifiers (720 combinations in total). The processing pipeline is shown in Fig. 2 and described in more detail in the sections below.

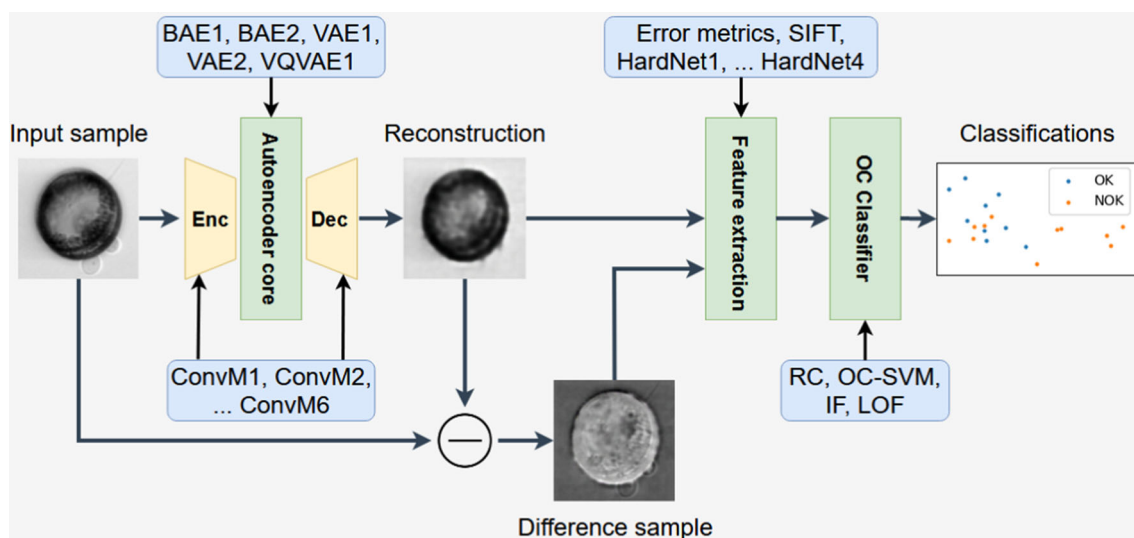
The anomaly detection is based on comparison between the original data and the autoencoder-reconstructed data, followed by feature extraction and one-class classification.

#### 3.1.1 Autoencoder architectures and convolutional layers

As the first step of anomaly detection, we use AE models trained only on OK data to reconstruct unknown input samples of both OK and NOK classes. On account of the non-optimal generalization of the AE models and training only on the OK class of data, we hypothesize that data from the NOK class will be reconstructed worse than data from the OK class, as described in [17].

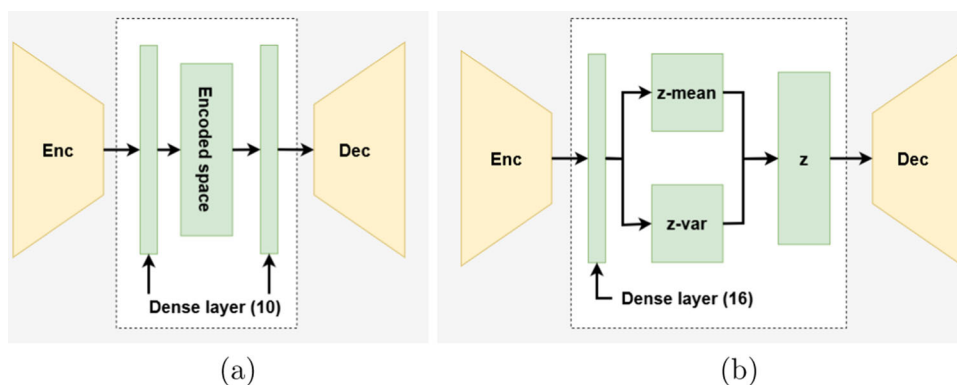
To better understand the effect of the AE architecture's core and the complexity of the convolutional encoding and decoding layers, we decided to build our implementation such that the core of the model could be combined with the selected convolutional pairs of the encoders and the decoders. The proposed structure allows us to analyze the contributions of the selected architecture and convolutional layers separately.

We evaluated five different options for the AE cores. As the first alternative, we used implementations of basic convolutional AE [34] as the BAE1 core, convolutional variational AE [35] as the VAE1 core, and vector-quantized AE [12] as the VQVAE1 core. As well as using these cores, we tried to further reduce the features extracted by an encoder by inserting fully connected layers to the basic convolutional AE as the BAE2 core [36] and to the variational AE as the VAE2 core. These modifications to the autoencoder cores are shown in Fig. 3.



**Fig. 2** Processing pipeline of the proposed autoencoder-based anomaly detection method

**Fig. 3** Schemes of the modified autoencoder cores: **a** BAE1 core, **b** VAE2 core



We expect that the basic convolutional AE is going to be surpassed in performance by both the variational and the vector-quantized cores because of their non-probabilistic encoding space, which allows leak of more input image’s anomalous parts to the encoded space and reconstructed image. The quality of the reconstructed images should be better in the case of the basic and vector-quantized cores than with variational cores, which typically produce blurry outputs [37]. When training on different classes, the best results are expected from the vector-quantized core, which should create separable clusters for each class in the encoded space.

Besides the AE cores described above, we also consider six pairs of convolutional encoding and decoding layer architectures, whose structure is described in the complementary tables: Table 5 for encoders, and Table 6 for decoders. Each convolutional layer or block described in these tables is complemented with the batch normalization layer. The activation function was set as Leaky ReLu by the ConvM1 architecture and as ReLu for the other architectures.

The tested convolutional layers go from the more complex ConvM2 architecture, suggested for anomaly detection in [19], and ConvM1 architecture, where we expect the ability to reconstruct fine features and details, to the simpler architectures ConvM5, ConvM4 and ConvM3. By using the simpler architectures, we expect that fine features and smaller image structures will be suppressed and the architecture might thus perform better on shape or structure anomalies. The last architecture, ConvM6, is unsymmetrical, as suggested in [38], and uses the more complex encoder of the ConvM5 architecture and the simpler decoder of the ConvM4 architecture. Using this architecture, we expect that anomalies that are propagated to the encoded space will be further suppressed by the decoder reconstruction.

In the optimal case, anomalous areas of the original image are removed during the image reconstruction as shown in Fig. 11. A difference image between the original sample and the reconstructed sample is then computed and used in the feature extraction.

### 3.1.2 Feature extraction

The second step of the framework applies feature extractors to analyze the reconstructions. The features are based on comparison between the original and reconstructed data (Error metrics, HardNet3 and HardNet4) or analysis of the difference image (SIFT feature extraction, HardNet1 and HardNet2).

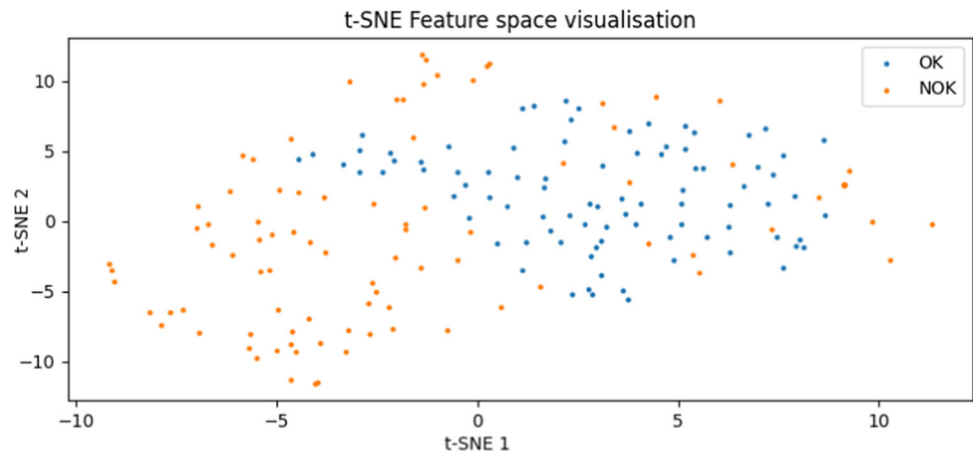
The first feature extraction approach (Error metrics) creates a low-dimensional feature vector for each image by computing selected error metrics between the original and reconstructed images. The L2 and SSIM metrics applied in [36] are complemented with the Average hash and mean-squared error metrics.

The second feature extraction method (SIFT feature extraction) uses scale and metrics properties of the image keypoints found by the SIFT method. It is a direct reimplementation the approach presented in [39]. The method uses difference images between the original and reconstructed data.

The last four feature extraction methods (HardNet1, HardNet2, HardNet3 and HardNet4) are all based on the batch similarity metric presented in [13]. HardNet1 is the simplest method where each sample is described by the HardNet (HN) feature vector of the original image resized to the size of  $32 \times 32$  as required by the original HN implementation. Since such resizing might not be optimal for small anomalies, HardNet2 splits the image of the original size to blocks of  $32 \times 32$  and computes the HN feature vector for each such block. The resulting feature vector consists of the norms over those vectors. HardNet3 splits the original and reconstructed images to  $32 \times 32$  blocks as in the HardNet2 method, but the resulting feature vector is computed as a cosine similarity between the HN feature vectors of the corresponding blocks of the original and decoded images. HardNet4 uses the same technique, but the cosine similarity is supplemented by the logarithm, which is supposed to emphasize smaller differences of the HardNet3 feature vector.

A 2D visualization of the resulting feature space obtained by the ConvM5-BAE2 autoencoder over the *Aphanizomenon*

**Fig. 4** Example feature space of the *Aphanizomenon* plankton species



plankton species using the HardNet2 feature extractor is shown in Fig. 4. The OK samples form an elliptical cluster, and most of the NOK samples are separate from that cluster.

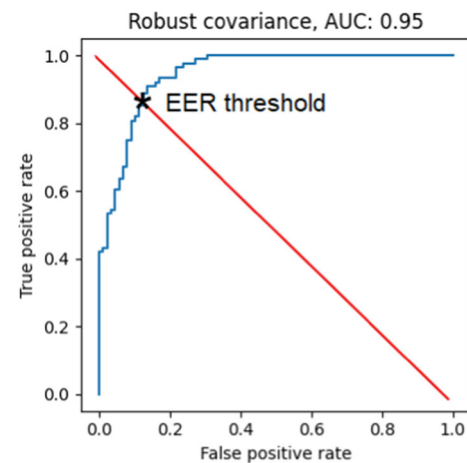
### 3.1.3 One-class classification

For the classification part, we used the following one-class classifiers:

- Robust covariance (RC) [40]: The RC classifier assumes the same distribution for all OK samples and fits an elliptic envelope to the central data point. The anomaly score is computed using the distribution estimations and Mahalanobis distance.
- One-class SVM (OC-SVM) [41]: The OC-SVM classifier utilizes the support vector machine (SVM) and a nonlinear kernel to create a separating hyperplane of the training data from the origin of the feature space. Samples on the other side of this hyperplane are considered as anomalies.
- Isolation Forest (IF) [42]: The IF classifier uses random feature selection and splitting to isolate observed samples. The anomaly score is based on the total number of splits. Anomalies are supposed to have a smaller number of splits as it should be easier to separate them.
- Local Outlier Factor (LOF) [14]: The LOF classifier is based on the local density deviation of the observed point with respect to its  $k$ -nearest neighbors. The density of the anomalies should be lower in comparison with the OK samples, which are considered to create denser clusters.

The fraction of anomaly samples for the OC-SVM, IF and LOF was set to 1% since this value is the minimum value of common implementations. Based on the normal distribution, we should also assume that even some OK samples might slightly differ from the majority. All classifiers are fit on the dataset containing only OK samples.

Input features for the one-class classification are normalized using robust scaling, which normalizes the median



**Fig. 5** Illustration of equal-error-rate (EER) threshold selection criterion on the ROC curve

and the interquartile range, as suggested in [43]. This normalization should be more robust to outliers than simple normalization approaches such as min–max normalization or standardization.

To select the optimal decision threshold for anomaly detection, we use the equal error rate (EER) over the ROC curve of the classifier as shown in Fig. 5. All classifiers are fit only on the OK data, and the ROC curve is obtained from the test dataset.

## 3.2 Object detection-based approach

The Faster R-CNN [9] algorithm is composed of three main components: a base feature extractor network, a region proposal network (RPN) for extracting the regions of interest, and a detector that uses the region proposals and respective feature maps to classify the detected objects as shown in Fig. 6. The first component is the feature extractor responsible for generating feature maps from the input image. This module is usually a CNN such as VGG-16 [44] or ResNet-50 [45].

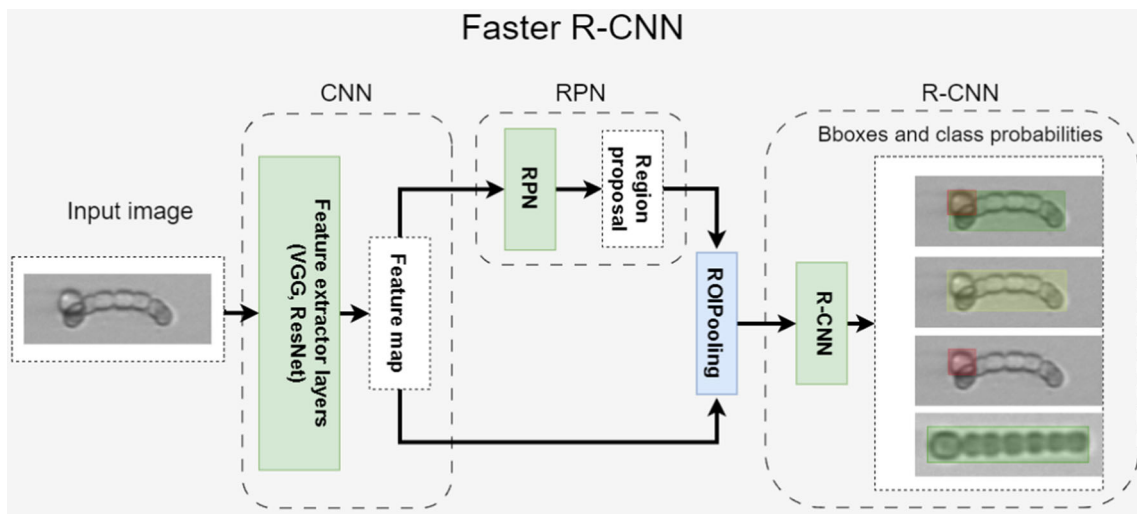


Fig. 6 Faster R-CNN architecture

Fig. 7 The Faster R-CNN approach for anomaly detection

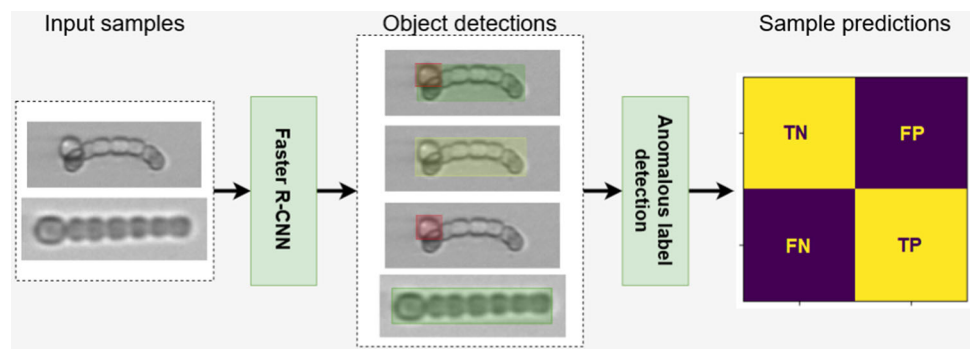
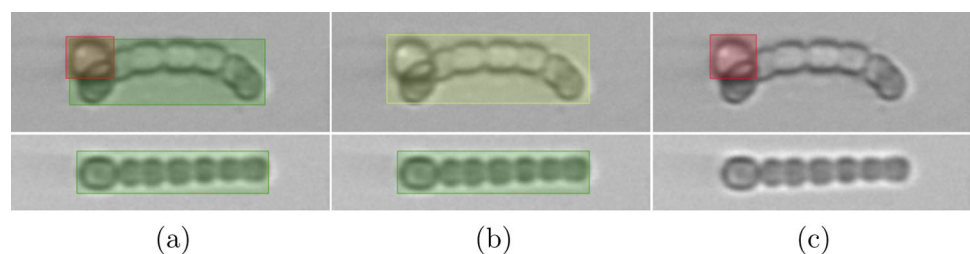


Fig. 8 Object detection tasks using the Faster R-CNN approach: **a** Plankton versus Anomalies; **b** Plankton versus Anomalous Plankton; **c** Anomalies only. The NOK samples are shown in the top row and the OK samples in the bottom row



The RPN is a kind of fully convolutional network that takes the feature maps from the previous step and returns a set of region proposals that guide the detector on where to find the objects in the image. The proposals and corresponding feature maps from the CNN are then utilized to yield candidate objects with bounding boxes and fixed-length feature vectors using the ROI pooling layer. Finally, these outputs are passed to the R-CNN network. The R-CNN network uses the proposed feature maps to classify each bounding box as an object or background and predict final class scores with the bounding boxes.

For our object detection experiments, we used the Faster R-CNN implementation available from [46] based on the ResNet-50 backbone presented in [47]. To employ an anomaly detection task in the Faster R-CNN baseline, the

architecture is supplemented by a one-class classification module based on the predicted object labels, as shown in Fig. 7.

Since anomalies such as parasites are relatively small compared to the image size, it is important to consider the anchor generator which is a part of the region-proposal network. Anchors define regions of an image, usually of different aspect ratios and sizes, that are used as references to detect objects. The anchor generator creates a set of anchors for each location in a feature map; then, for each region of interest, the model predicts which anchor box best encloses the object. The choice of an anchor generator mostly depends on the type of detection task. For example, if we want to detect small objects, then a smaller anchor size should be used. On the other hand, if the task is to detect objects of various sizes,

a range of anchor sizes should be defined [9]. Additionally, the aspect ratios of the anchors should match the aspect ratios of the objects in the image.

As suggested in [11], three separate object detectors are considered, each trained on different ground truth: (1) plankton and anomalies, (2) plankton (clean) and anomalous plankton, and (3) anomalies only (see Fig. 8). In the first column, we can see that the model detects a plankton sample in both cases and an anomaly in the top row. The second column shows detection of a plankton sample with anomaly in the top row and detection of a clean sample in the bottom row, and finally, the third column shows detection of an anomaly in the top row only.

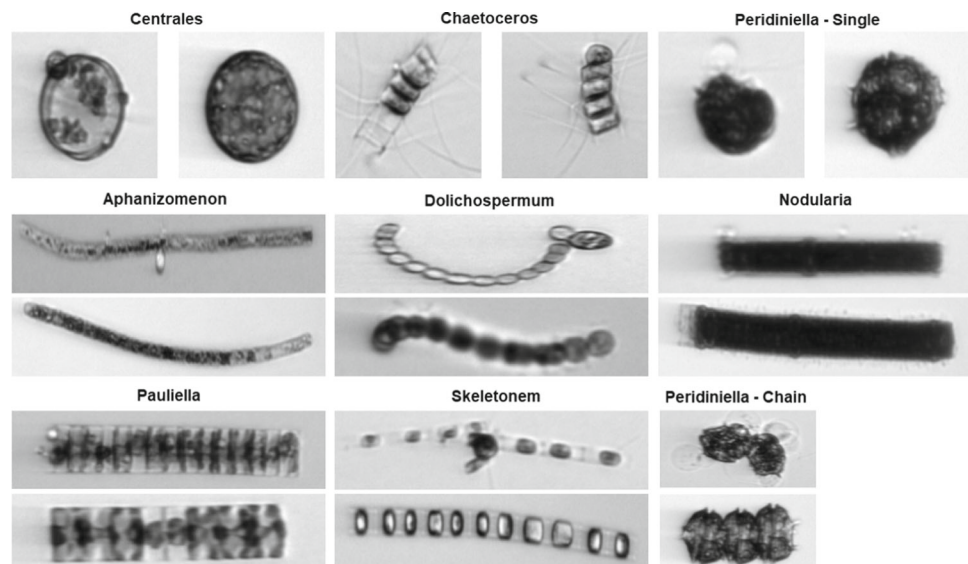
## 4 Experiments

In this section, we describe the datasets used, the evaluation metrics, and the results of the autoencoder-based experiments and the object detection-based experiments.

**Table 1** Species-specific statistics of the plankton anomaly dataset

Plankton class	OK samples count	NOK samples count
Aphanizomenon	830	140
Centrales	400	57
Dolichospermum	515	406
Chaetoceros	606	371
Nodularia	118	357
Pauliella	160	433
Peridiniella Chain	183	31
Peridiniella Single	459	63
Skeletonema	769	419

**Fig. 9** Anomalous (left column, or upper row) and non-anomalous samples (right column, or lower row) from all dataset classes of the used dataset



## 4.1 Phytoplankton anomaly dataset

Natural Baltic Sea phytoplankton communities are continuously imaged with an Imaging FlowCytobot (IFCB) [48] deployed at Utö Atmospheric and Marine Research Station, Finland (59°46.84' N, 21°22.13' E). The IFCB is connected to the station flow-through system, which receives water pumped from an ~5 m deep inlet located 250 m offshore, representative of the sub-surface layer. At Utö, IFCB takes a 5-ml sample nearly every 20 min and the system is set to trigger based on the detection of chlorophyll, i.e., targeting phytoplankton cells rather than non-living particles. The research station and IFCB deployment at Utö are described in detail in [49] and [50].

The phytoplankton data from Utö IFCB can be currently classified near real-time into 50 different classes, as described by [51]. Putative parasite infection images were manually annotated by experts based on other Utö data, collected between February and August 2021, using phytoplankton data from nine classes. These classes were selected based on their importance during the spring or summer blooms in the Baltic Sea.

In our experiments, we used a phytoplankton anomaly dataset derived from the annotated images used to train the classifier described above with OK samples from the dataset published in [51] and NOK samples from unpublished 2021 Utö data. The anomaly dataset contains over 6200 manually annotated and expert-validated samples for 9 plankton classes with known anomalies, as shown in Table 1. Non-anomalous and anomalous samples of each class are shown in Fig. 9. As an annotation tool, we used the free version of the Label Studio available at [52]. The annotated dataset is available online at [53] in both COCO and YOLO formats.



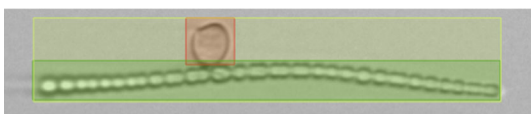


Fig. 10 Example of the annotation bounding boxes

### 4.1.1 Dataset annotations

When annotating the dataset, we used three different labels to define a separate species set:

- The label *Anomaly* marks the parasite or other anomalies on the plankton sample.
- The label *PlanktonSpecies\_Anomaly* marks plankton species with the attached parasite.
- The label *PlanktonSpecies\_Clean* marks plankton species with or without the parasite.

The last two labels could overlap, but whenever it was possible, the *PlanktonSpecies\_Clean* label does not cover the sample part with parasite. To distinguish between the OK and NOK samples, the *PlanktonSpecies\_Clean* label should be removed if it overlaps with the *PlanktonSpecies\_Anomaly* one.

An example of the annotation over a *Dolichospermum* plankton species sample is shown in Fig. 10. The red color marks a plankton anomaly and, in this case, the darker green marks the clean sample and the lighter green marks the sample with an anomaly.

### 4.1.2 Derived dataset for autoencoder-based experiment

For the purposes of the autoencoder-based experiment, we used the above-described dataset to derive a one-class dataset with no NOK samples and 70% of the OK sample in the training set. Test and validation datasets always contain a balanced number of OK and NOK samples. The experiment with all plankton species contains all available training samples, 10 validation samples, and 10 test samples from each species.

In order to help the AE model to learn more robust features, we added salt-and-pepper noise to the image samples used during the training with a clean sample used as a label as suggested in [54]. Besides this noise augmentation, we also use random flipping, contrast, saturation, brightness, inversion, and hue augmentation.

Because the HardNet-based feature extractors work correctly only with image sizes of multiples of 32, all samples were resized with respect to the major aspect ratio of each class (1:4 for five classes, 1:1 for three classes and 1:2 for one class) as can be seen in Fig. 9. For the experiment over all classes, we chose the aspect ratio of 1:2 as a compromise.

### 4.1.3 Derived dataset for object detection-based experiment

For the object detection experiment, the model was trained in a supervised manner. The split ratios were set as 70%, 10%, and 20% for the training, validation, and test subsets, respectively. Training and validation sets do not include clean samples, whereas a test set contains a balanced number of anomalies and clean images.

Additionally, we applied the following augmentation techniques: horizontal and vertical flip with a probability of 30%, and random brightness, contrast and saturation adjustment with a probability of 10%.

### 4.2 Performance metrics

To compare the results of the autoencoder and object detection experiments, we need to evaluate the predictions of the models with respect to the ground-truth labels. To do so, we can define true-positive (TP) and true-negative (TN) predictions, where the model correctly classifies OK and NOK samples, together with false-positive (FP) and false-negative (FN) predictions, where the model misclassifies NOK samples as OK in the FP case and OK samples as NOK in the FN case.

Precision, Recall and F1 score metrics are used for comparison of the different variations of autoencoders and object detection methods. The metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

Similarly to precision and recall, we can also define specificity as:

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

In the autoencoder experiment, we complemented the metrics with the area under the curve (AUC) score. This parameter is defined as the area under the receiver operator characteristics (ROC) curve, an example of which is shown in Fig. 5. This curve is obtained by changing the decision threshold of a binary classifier by a defined step and plotting the resulting specificity on the x-axis and recall on the y-axis for each threshold step. Each point of the ROC curve then corresponds to one threshold setting.

### 4.3 Autoencoder-based approach

Due to the high number of combinations in our framework (five autoencoder types, six pairs of convolutional layers, six feature extractors and four classifiers), we decided to split the experimental results into two parts. The first part describes the optimal combination of model, feature extractor and classifier trained on all datasets together with its selection criteria, and the second part describes the optimal results achieved per plankton species. The optimal combination of autoencoder model, convolutional layers, feature extractor and one-class classifier was determined based on the maximum-achieved F1 score because the detection results appeared to be more consistent in comparison with the AUC metric. This metric is also advantageous because of the easier comparison with the results of the object detector-based approach. The whole implementation was built using the TensorFlow 2 platform [55] and Scikit-learn library [56].

#### 4.3.1 Optimal model for anomaly detection

To select an optimal anomaly detection model, we analyzed the results of all model combinations for the experiment with all plankton species. The best results were achieved with the autoencoder ConvM2-VQVAE1, HardNet1 as the feature extractor, and Local Outlier Factor as the one-class classifier. These results are reported in Table 2, and illustrative examples of the results are shown in Fig. 11.

To further demonstrate that the selected anomaly detection model performs best, we evaluated the F1 score over: (1) model combinations with a fixed feature extractor and classifier (Table 7), (2) feature extractor combinations with a fixed model and classifier (Table 10), and (3) classifier combinations with a fixed model and feature extractor (Table 11). The highest F1 score was 0.75 consistently over all described combinations.

**Table 2** Species-specific results with the optimal combination anomaly detection model over all plankton species based on the highest F1-score (ConvM2-VQVAE1, HardNet1, Local Outlier Factor)

Plankton class	AUC score	F1 score	Prec	Rec
Aphanizomenon	0.89	0.83	0.83	0.83
Centrales	0.64	0.60	0.60	0.60
Dolichospermum	0.87	0.81	0.83	0.80
Chaetoceros	0.64	0.61	0.61	0.61
Nodularia	0.70	0.67	0.67	0.67
Pauliella	0.73	0.70	0.70	0.70
Peridiniella Chain	0.73	0.69	0.69	0.69
Peridiniella Single	0.90	0.83	0.83	0.83
Skeletonema	0.80	0.76	0.76	0.76
Plankton all	0.80	0.75	0.75	0.74

#### 4.3.2 Optimal anomaly detection model per plankton class

Results of the optimal anomaly detection models per plankton class are shown in Table 12. The detection results are approximately 10% better than when using the optimal anomaly detection model trained on all datasets described above, which could be particularly important for the *Centrales* and *Chaetoceros*, which showed the lowest values on the performance metrics. This performance gain nevertheless comes at the cost of the need for a separately trained anomaly detection model for each plankton class.

### 4.4 Object detection-based approach

Results of the *Plankton versus Anomalies*, *Plankton versus Anomalous Plankton* and *Anomalies Only* experiments over all samples are shown in Table 3. The Plankton versus Anomalies experiment contains both large bounding boxes of plankton annotations and small bounding boxes of anomalies. Therefore, the anchor generator was set up such that the sizes of feature map were 16, 32, 64, 128, 256 and 512. In the Plankton versus Anomalous Plankton experiment, only the large bounding boxes were used, and the sizes were 64, 128, 256, 512, and 1024. For the Anomalies only experiment, the sizes were 4, 8, 16, 32, 64, and 128. The scales and the aspect ratios of sizes for each experiment were the same: 0.5, 1.0, 1.5, 2.0, 3.0.

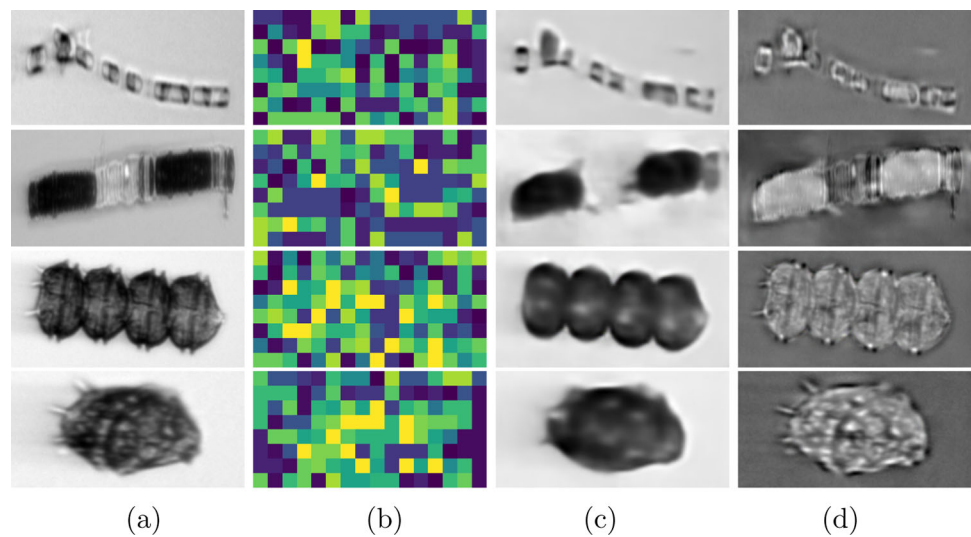
The highest F1 score was achieved with the Plankton versus Anomalies experiment and the lowest one with the Anomalies only experiment. We were able to reach a high F1 score also in the Plankton versus Anomalous Plankton experiment, but the resulting plankton labels were often misleading in this case.

For the object detection approach, one major issue is that the model is incapable to distinguish between plankton parts and anomalies, as shown in Fig 12. This major drawback of the Faster R-CNN object detection method originates from the architecture itself and could not be solved by anchor modifications or other parameter tuning.

To have a better comparison with Table 12, we also performed an anomalies only experiment trained on species-specific data, whose results are shown in Table 4. In this experiment, the approach performed worse than the species-specific autoencoder experiment and also worse than the universal model on average.

We also provide supplementary material as Table 14, Table 13 and Table 15, which show the results of the Plankton versus Anomalies, Plankton versus Anomalous Plankton and Anomalies only experiments with respect to the individual plankton class.

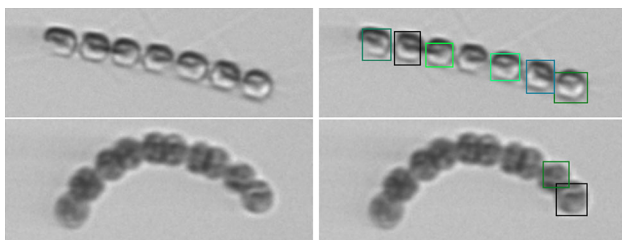
**Fig. 11** Illustration of the results with the ConvM2-VQVAE1 autoencoder model: **a** original, **b** encoded space, **c** reconstruction and **d** difference image



**Table 3** Faster-RCNN detection results

Configuration	F1 score	Prec	Rec
Plankton versus Anomalies	<b>0.86</b>	0.94	0.79
Plankton versus Anomalous Plankton	0.83	0.82	0.85
Anomalies only	0.75	0.76	0.74

Bold value indicates the best performing models and the best achieved results



**Fig. 12** Plankton species without any anomalies recognized as an anomaly by Faster RCNN. The model is confused by a chain of cells and cannot distinguish between plankton parts and anomalies

**Table 4** Species-specific dataset Faster-RCNN detection results

Plankton class	F1 score	Prec	Rec
Aphanizomenon	0.66	0.63	0.68
Centrales	0.67	0.50	1.00
Dolichospermum	0.38	0.41	0.36
Chaetoceros	0.43	0.38	0.43
Nodularia	0.79	0.72	0.87
Pauliella	0.75	0.69	0.83
Peridiniella Chain	0.67	0.50	1.00
Peridiniella Single	0.67	0.50	1.00
Skeletonema	0.53	0.48	0.61
Average	0.62	0.53	0.75

### 5 Discussion

Learning to detect parasites from phytoplankton images is a challenging problem due to large variation in the appearance of the plankton cells and parasites, the small size of parasites combined with the limited spatial resolution of the images, as well as the scarcity of training data because of the relative rarity of plankton cells with parasites. Even for an expert, it is often impossible to confirm with certainty the parasitic nature of all the attached (non-host) structures from the images alone. For example, spherical structures that are attached to the host cell and have a different appearance to the phytoplankton cell are typically parasites, but they can also be loosely attached free living (i.e., non-parasitic) cells or phytoplankton-cell derived organelles expelled from the cell due to stress. Complexity of those structures makes it infeasible to collect annotated training and test data just on phytoplankton parasites. Therefore, we formulated the problem as anomaly detection where the goal is to detect the phytoplankton cells that deviate from “healthy” cells. The method can be seen as an anomaly detector of putative parasite infections allowing screening of large volumes of plankton image data to obtain a subset of interesting images for further analysis.

A large dataset of phytoplankton cell images with and without anomalies was collected for the study. Untypically for most anomaly detection studies, the collected dataset contains a relatively large number of images with putative parasites (NOK samples). This made it possible to train also supervised object detectors for the task and to compare unsupervised anomaly detection methods with object detector-based methods. We evaluated two approaches to detect whether an image contains anomalies or not: (1) an autoencoder-based anomaly detection approach and (2) a Faster R-CNN-based object detector for anomalies.

For the autoencoder-based approach, a full pipeline consists of the CNN-based autoencoder architecture, feature extraction from the reconstruction and difference image, and one-class classifier was proposed. Various methods were considered for each part of the pipeline and extensively evaluated. The best overall accuracy (F1 score) was obtained using the combination of vector-quantized variational autoencoder (VQVAE) [12] architecture with the CNN backbone by [19], the HardNet feature extractor [13] and the Local Outlier Factor classifier [14]. The F1 score for the best combination varied between different classes from 0.6 (*Centrales*) to 0.83 (*Peridiniella Single*) with an overall F1 score of 0.75 over all classes. The ablation study (Tables 8, 9, 10 and 11) demonstrated the superiority of the proposed combination over the other alternatives. The accuracy could be further improved by optimizing the method for each phytoplankton classes separately, as can be seen from Table 12 with F1 scores varying from 0.73 to 0.94. While fine-tuning class-specific models reduces the generalizability of the method, these are promising results for studying parasites on individual plankton species.

The limited amount of training data is a notable challenge for supervised object detectors. To properly learn the large variation in the appearance of anomalies, the training stage would require a sufficient number of example images for each class. The difficulty of the detection task is further emphasized by the fact that the anomalies are typically attached to the host cell and are often very small compared to the plankton cell. These challenges are apparent when observing the Faster R-CNN results (Table 3), where the accuracies are not as high as commonly seen in object detection problems. Three configurations for the R-CNN-based method were evaluated: (1) detection of anomalies only, (2) detection of both anomalies and plankton cells, and (3) detection of healthy plankton cells and plankton cells with anomalies as separate classes. Based on the results, it is evident that learning how normal plankton cells look like is beneficial for the R-CNN. The model trained only on anomalies tends to often detect parts of the phytoplankton itself as anomalies. It was further noticed that when trained on healthy plankton cells and plankton cells with anomalies as separate classes, the detector often fails to correctly detect the bounding boxes in the presence of anomalies but still produces correct classification results (OK vs. NOK). This raises questions about the generalizability of the method.

The Faster R-CNN-based method (model trained on both anomalies and plankton cells) achieved higher accuracy (F1 score: 0.86) than the autoencoder-based method (F1 score: 0.75). This is understandable as the autoencoder-based method did not have access to the images with anomalies during the training stage. The Faster R-CNN-based method is more suitable when enough annotated training data is available. This, however, is not typically the case in plankton

anomaly and parasite detection due to the reasons discussed above. The autoencoder-based approach has some notable advantages over supervised object detectors: (1) no training data with anomalies are needed, (2) no annotated bounding boxes are needed, and (3) the method works also with previously unseen anomalies. These advantages together with the comparable accuracy to the R-CNN-based method make the proposed autoencoder method a more promising approach for plankton anomaly detection on new datasets.

Being able to screen large volumes of plankton image data for anomalies has potential to noticeably reduce manual work and allows more extensive research on parasitic infections. Ecologically speaking, separating cells with anomalies is interesting and can lead to new research avenues in the future. Anomaly detections can give a valuable first hint of putative parasite infections (or physiologically stressed phytoplankton). However, further method development is needed to make it possible to distinguish between the different types of anomalies and relate them with more certainty to parasites. An interesting future direction would be to apply clustering methods for anomalies to identify different types of anomalies. The classification of anomaly types could be validated by a wider community effort with expertise on different parasite groups, epibionts and symbionts. In combination with classifying different types of anomalies, different datasets could also be collected from known parasites/epibionts/host-derived anomalies from culture systems. Detected parasites and their presence could be further qualitatively confirmed in parallel by additional methods such as microscopy or eDNA-based approaches.

## 6 Conclusion

This paper presents an unsupervised anomaly detection approach to detect anomalies, which was tested over nine phytoplankton classes. Although studies exist on plankton anomaly detection in the context of open-set recognition, i.e., detecting previously unseen plankton classes and non-plankton particles, our paper is according to our best knowledge the first one focusing on the detection of small anomalies such as potential phytoplankton parasites or infections in a known set of plankton classes.

We propose an anomaly detection pipeline consisting of a vector-quantized variational autoencoder (VQVAE) in combination with the eight-layer deep convolutional architecture (ConvM2), the HardNet feature extractor and the Local Outlier Factor classifier. With this pipeline, we achieved an average F1 score of 0.75 for all nine analyzed phytoplankton classes. We also suggest that the achieved anomaly detection results could be further improved by optimizing the components of the pipeline for each class separately.

The results achieved with this approach were compared to a supervised Faster R-CNN object detector for experiments in three configurations: (1) detection of anomalies only, (2) detection of anomalies and plankton cells, and (3) detection of plankton cells with and without anomalies. The best results were achieved with the second configuration with an average F1 score of 0.86. Although this score is higher than the one achieved by the autoencoder proposed in this paper, our approach is more universal because it can detect also previously unseen data and it needs no anomalous samples for its training. These benefits make the proposed autoencoder approach more promising for plankton research, where annotated anomaly datasets are not available or their collection is infeasible. The code and dataset used in this study have been made publicly available as a part of this paper.

To identify parasitic infections in phytoplankton, manual scrutiny of the data is still needed as our knowledge of them is still very poor. Our proposed method, however, provides a solution to reduce the workload of manual checking enormously by pointing out the proportion of images an expert potentially needs to go through. This opens a possibility to begin understanding the important role that parasites play in the phytoplankton community dynamics.

**Acknowledgements** The research was carried out in the FASTVISION and FASTVISION-plus projects funded by the Academy of Finland (Decision Numbers 321980, 321991, 339612, and 339355). This study utilized research infrastructure as part of FINMARI (Finnish Marine Research Infrastructure consortium). LH was supported by OBAMA-NEXT (Grant Agreement No. 101081642), funded by the European Union under the Horizon Europe program. SVdW and JK were supported by the Academy of Finland (Decision Numbers 340659 and 346387). The work was further supported by the Grant Number FEKT-S-23-8451 "Research on advanced methods and technologies in cybernetics, robotics, artificial intelligence, automation and measurement" from the Internal science fund of Brno University of Technology.

**Funding** Open Access funding provided by LUT University (previously Lappeenranta University of Technology (LUT)).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix A: Convolutional encoders and decoders

See Tables 5 and 6.

**Table 5** Overview of the tested convolutional encoder models

Encoder name	Layer name	Filters	Kernel size	Stride
ConvM1	ConvE1	32	3 × 3	2
	ConvE2	64	3 × 3	2
	ConvE3	64	3 × 3	2
	ConvE4	64	3 × 3	2
	ConvE5	64	3 × 3	2
ConvM2	ConvE1	32	4 × 4	2
	ConvE2	32	4 × 4	2
	ConvE3	32	3 × 3	1
	ConvE4	64	4 × 4	2
	ConvE5	64	3 × 3	1
	ConvE6	128	4 × 4	2
	ConvE7	64	3 × 3	1
	ConvE8	32	3 × 3	1
ConvM3	ConvE1	32	3 × 3	2
	ConvE2	64	3 × 3	2
ConvM4	ConvE1	8	5 × 5	1
	MaxPool	–	2 × 2	–
	ConvE2	4	3 × 3	1
ConvM5	MaxPool	–	2 × 2	–
	ConvE1	16	3 × 3	1
	MaxPool	–	2 × 2	–
	ConvE2	8	3 × 3	1
	MaxPool	–	2 × 2	–
	ConvE3	4	3 × 3	1
	MaxPool	–	2 × 2	–

**Table 6** Overview of the tested convolutional decoder models

Decoder name	Layer name	Filters	Kernel size	Stride
ConvM1	ConvT1	64	3 × 3	2
	ConvT2	64	3 × 3	2
	ConvT3	64	3 × 3	2
	ConvT4	32	3 × 3	2
	ConvT5	Image channels	3 × 3	2
ConvM2	ConvD1	16	3 × 3	1
	ConvD2	64	3 × 3	1
	Upsampling	-	2 × 2	-
	ConvD3	128	4 × 4	1
	Upsampling	-	2 × 2	-
	ConvD4	64	3 × 3	1
	Upsampling	-	2 × 2	-
	ConvD5	64	4 × 4	1
	Upsampling	-	2 × 2	-
	ConvD6	32	3 × 3	1
	Upsampling	-	2 × 2	-
	ConvD7	32	4 × 4	1
	Upsampling	-	2 × 2	-
	ConvD8	32	4 × 4	1
	Upsampling	-	2 × 2	-
ConvD8	Image channels	8 × 8	1	
ConvM3	ConvD1	64	3 × 3	1
	ConvD2	32	3 × 3	1
	Upsampling	-	2 × 2	-
ConvM4	ConvD3	Image channels	3 × 3	1
	ConvD1	4	3 × 3	1
	Upsampling	-	2 × 2	-
ConvM5	ConvD2	8	3 × 3	1
	Upsampling	-	2 × 2	-
	ConvD3	Image channels	3 × 3	1
	ConvD1	4	3 × 3	1
	Upsampling	-	2 × 2	-
ConvM5	ConvD2	8	3 × 3	1
	Upsampling	-	2 × 2	-
	ConvD3	16	3 × 3	1
	Upsampling	-	2 × 2	-
	ConvD4	Image channels	3 × 3	1

## Appendix B: Complementary results of the autoencoder-based experiment

See Tables 7, 8, 9, 10, 11, 12

**Table 7** Results over all plankton species for fixed feature extractor (HardNet1) and classifier (Local Outlier Factor)

Model name	AUC score	F1 score	Prec	Rec
ConvM1-BAE1	0.73	0.70	0.70	0.70
ConvM1-BAE2	0.81	0.72	0.72	0.72
ConvM1-VAE1	0.74	0.66	0.66	0.66
ConvM1-VAE2	0.72	0.65	0.65	0.64
ConvM1-VQVAE1	0.72	0.64	0.64	0.64
ConvM2-BAE1	0.73	0.67	0.67	0.67
ConvM2-BAE2	0.79	0.72	0.72	0.72
ConvM2-VAE1	0.72	0.67	0.67	0.67
ConvM2-VAE2	0.79	0.70	0.71	0.70
ConvM2-VQVAE1	0.80	<b>0.75</b>	0.75	0.74
ConvM3-BAE1	0.68	0.60	0.60	0.60
ConvM3-BAE2	0.69	0.62	0.62	0.62
ConvM3-VAE1	0.72	0.67	0.67	0.67
ConvM3-VAE2	0.69	0.63	0.63	0.63
ConvM3-VQVAE1	0.71	0.65	0.66	0.64
ConvM4-BAE1	0.70	0.61	0.62	0.61
ConvM4-BAE2	0.72	0.64	0.65	0.63
ConvM4-VAE1	0.69	0.64	0.64	0.64
ConvM4-VAE2	0.71	0.64	0.64	0.64
ConvM4-VQVAE1	0.74	0.71	0.71	0.71
ConvM5-BAE1	0.74	0.69	0.69	0.69
ConvM5-BAE2	0.69	0.64	0.64	0.64
ConvM5-VAE1	0.66	0.57	0.57	0.58
ConvM5-VAE2	0.74	0.68	0.68	0.68
ConvM5-VQVAE1	0.72	0.68	0.69	0.68
ConvM6-BAE1	0.70	0.64	0.64	0.64
ConvM6-BAE2	0.73	0.68	0.68	0.68
ConvM6-VAE1	0.72	0.69	0.69	0.69
ConvM6-VAE2	0.75	0.70	0.70	0.70
ConvM6-VQVAE1	0.71	0.67	0.67	0.67

Bold value indicates the best performing models and the best achieved results

**Table 8** Results over all plankton classes for fixed convolutional layer architecture (ConvM2), feature extractor (HardNet1) and classifier (Local Outlier Factor)

Model architecture	AUC score	F1 score	Prec	Rec
BAE1	0.73	0.67	0.67	0.67
BAE2	0.79	0.72	0.72	0.72
VAE1	0.72	0.67	0.67	0.67
VAE2	0.79	0.70	0.71	0.70
VQVAE1	0.80	<b>0.75</b>	0.75	0.74

Bold value indicates the best performing models and the best achieved results

**Table 9** Results over all plankton classes for fixed model architecture (VQVAE1), feature extractor (HardNet1) and classifier (Local Outlier Factor)

Conv. architecture	AUC score	F1 score	Prec	Rec
ConvM1	0.72	0.64	0.64	0.64
ConvM2	0.80	<b>0.75</b>	0.75	0.74
ConvM3	0.71	0.65	0.66	0.64
ConvM4	0.74	0.71	0.71	0.71
ConvM5	0.72	0.68	0.69	0.68
ConvM6	0.71	0.67	0.67	0.67

Bold value indicates the best performing models and the best achieved results

**Table 10** Results over all plankton classes for fixed model (ConvM2-VQVAE1) and classifier (Local Outlier Factor)

Feature extractor	AUC score	F1 score	Prec	Rec
ErrMetrics	0.50	0.51	0.51	0.51
SIFT	0.49	0.43	0.43	0.43
HardNet1	0.80	<b>0.75</b>	0.75	0.74
HardNet2	0.64	0.60	0.60	0.60
HardNet3	0.75	0.68	0.70	0.67
HardNet4	0.77	0.73	0.73	0.72

Bold value indicates the best performing models and the best achieved results

**Table 11** Results over all plankton classes for fixed model (ConvM2-VQVAE1) and feature extractor (HardNet1)

Classifier	AUC score	F1 score	Prec	Rec
Robust covariance	0.66	0.61	0.62	0.61
One-Class SVM	0.59	0.56	0.56	0.57
Isolation Forest	0.54	0.53	0.53	0.52
Local Outlier Factor	0.80	<b>0.75</b>	0.75	0.74

Bold value indicates the best performing models and the best achieved results

**Table 12** Overview of the per-plankton-classes optimal anomaly detection combinations and their results

Plankton class	Model name	Feature extractor	Classifier	AUC score	F1 score	Prec	Rec
Aphanizomenon	ConvM5-BAE1	HardNet2	LOF	0.95	0.93	0.91	0.94
Centrales	ConvM2-VAE1	HardNet2	RC	0.74	0.73	0.72	0.72
Dolichospermum	ConvM6-BAE1	HardNet1	LOF	0.93	0.90	0.92	0.88
Chaetoceros	ConvM1-BAE1	HardNet1	RC	0.72	0.72	0.71	0.72
Nodularia	ConvM3-BAE1	HardNet1	RC	0.95	0.94	1.00	0.89
Pauliella	ConvM3-VAE1	HardNet3	RC	0.83	0.86	0.81	0.91
Peridiniella Chain	ConvM1-VAE1	HardNet4	IF	0.93	0.88	0.88	0.88
Peridiniella Single	ConvM6-BAE1	HardNet1	LOF	0.90	0.86	0.84	0.89
Skeletonema	ConvM6-VAE2	HardNet4	RC	0.90	0.84	0.84	0.84

## Appendix C: Complementary R-CNN results

See Tables 13, 14 and 15.

**Table 13** Faster-RCNN detection results for Plankton vs Anomalies experiment

Plankton class	F1 score	Prec	Rec
Aphanizomenon	0.98	1	0.96
Centrales	0.85	0.73	1
Dolichospermum	0.89	0.97	0.81
Chaetoceros	0.80	0.89	0.73
Nodularia	0.76	0.98	0.62
Pauliella	0.86	0.99	0.77
Peridiniella Chain	0.60	0.75	0.5
Peridiniella Single	0.76	1	0.62
Skeletonema	0.98	0.97	1
Plankton & Anomalies	0.87	0.95	0.81

**Table 14** Faster-RCNN detection results for Plankton vs Anomalous Plankton experiment

Plankton class	F1 score	Prec	Rec
Aphanizomenon	0.66	0.58	0.75
Centrales	0.60	0.47	0.82
Dolichospermum	0.85	0.82	0.88
Chaetoceros	0.74	0.71	0.78
Nodularia	0.78	0.96	0.66
Pauliella	0.79	0.76	0.83
Peridiniella Chain	0.67	0.67	0.67
Peridiniella Single	0.88	0.92	0.85
Skeletonema	0.94	0.92	0.94
Plankton & Anomalous plankton	0.83	0.82	0.85

**Table 15** Faster-RCNN detection results for Anomalies experiment

Plankton class	F1 score	Prec	Rec
Aphanizomenon	1	1	1
Centrales	0.77	0.67	0.91
Dolichospermum	0.38	0.87	0.25
Chaetoceros	0.67	0.66	0.68
Nodularia	0.85	0.88	0.82
Pauliella	0.85	0.78	0.94
Peridiniella Chain	0.44	0.67	0.33
Peridiniella Single	0.89	0.86	0.92
Skeletonema	0.83	0.75	0.93
Anomalies	0.75	0.76	0.74

## References

- Falkowski, P.G., Barber, R.T., Smetacek, V.: Biogeochemical controls and feedbacks on ocean primary production. *Science* **281**(5374), 200–206 (1998). <https://doi.org/10.1126/science.281.5374.200>
- Reynolds, C.S.: *The Ecology of Phytoplankton*, Ecology, Biodiversity and Conservation. Cambridge University Press, Cambridge (2006). <https://doi.org/10.1017/CBO9780511542145>
- Suttle, C.A., Chan, A.M., Cottrell, M.T.: Infection of phytoplankton by viruses and reduction of primary productivity. *Nature* **347**(6292), 467–469 (1990). <https://doi.org/10.1038/347467a0>
- Klawonn, I., Van den Wyngaert, S., Parada, A.E., et al.: Characterizing the “fungal shunt”: Parasitic fungi on diatoms affect carbon flow and bacterial communities in aquatic microbial food webs. *Proc. Natl. Acad. Sci.* **118**(23), e2102225,118 (2021). <https://doi.org/10.1073/pnas.2102225118>
- Klawonn, I., Van den Wyngaert, S., Iversen, M.H., et al.: Fungal parasitism on diatoms alters formation and bio-physical properties of sinking aggregates. *Commun. Biol.* **6**(1), 206 (2023). <https://doi.org/10.1038/s42003-023-04453-6>
- Scholz, B., Guillou, L., Marano, A.V., et al.: Zoosporic parasites infecting marine diatoms—a black box that needs to be opened. *Fungal Ecol.* **19**, 59–76 (2016). <https://doi.org/10.1016/j.funeco.2015.09.002>
- Peacock, E.E., Olson, R.J., Sosik, H.M.: Parasitic infection of the diatom *guinardia delicatula*, a recurrent and ecologically important phenomenon on the new england shelf. *Mar. Ecol. Prog. Ser.* **503**, 1–10 (2014). <https://doi.org/10.3354/meps10784>
- Lombard, F., Boss, E., Waite, A.M., et al.: Globally consistent quantitative observations of planktonic ecosystems. *Front. Mar. Sci.* **6**, 196 (2019). <https://doi.org/10.3389/fmars.2019.00196>
- Ren, S., He, K., Girshick, R., et al.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **28** (2015)
- Jocher, G.: YOLOv5 by Ultralytics. <https://doi.org/10.5281/zenodo.3908559>, <https://github.com/ultralytics/yolov5> (2020)
- Bilik, S., Kratochvila, L., Ligocki, A., et al.: Visual diagnosis of the varroa destructor parasitic mite in honeybees using object detector techniques. *Sensors* **21**(8), 2764 (2021). <https://doi.org/10.3390/s21082764>
- Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: Guyon, I., Luxburg, U.V., Bengio, S., et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates Inc, New York (2017)
- Mishchuk, A., Mishkin, D., Radenovic, F., et al.: Working hard to know your neighbor’s margins: Local descriptor learning loss. In: Guyon, I., Luxburg, U.V., Bengio, S., et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates Inc, New York (2017)
- Breunig, M.M., Kriegel, H.P., Ng, R.T., et al.: Lof: Identifying density-based local outliers. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA, SIGMOD’00, pp. 93–104, (2000) <https://doi.org/10.1145/342009.335388>
- Pang, G., Shen, C., Cao, L., et al.: Deep learning for anomaly detection: a review. *ACM Comput. Surv.* **54**(2), 1–38 (2021). <https://doi.org/10.1145/3439950>
- Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786), 504–507 (2006). <https://doi.org/10.1126/science.1127647>
- Sakurada, M., Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory*



- Data Analysis. Association for Computing Machinery, New York, NY, USA, MLSDA'14, pp. 4–11, (2014) <https://doi.org/10.1145/2689746.2689747>
18. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. *Spec. Lect. IE* **2**(1), 1–18 (2015)
  19. Bergmann, P., Löwe, S., Fauser, M., et al.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, INSTICC*. SciTePress, pp. 372–380 (2019). <https://doi.org/10.5220/0007364503720380>
  20. Charte, D., Charte, F., García, S., et al.: A practical tutorial on autoencoders for nonlinear feature fusion: taxonomy, models, software and guidelines. *Inf. Fusion* **44**, 78–96 (2018). <https://doi.org/10.1016/j.inffus.2017.12.007>
  21. Pastore, V.P., Zimmerman, T.G., Biswas, S.K., et al.: Annotation-free learning of plankton for classification and anomaly detection. *Sci. Rep.* **10**(1), 12,142 (2020). <https://doi.org/10.1038/s41598-020-68662-3>
  22. Pu, Y., Feng, Z., Wang, Z., et al.: Anomaly detection for in situ marine plankton images. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 3661–3671 (2021)
  23. Pastore, V.P., Megiddo, N., Bianco, S.: An anomaly detection approach for plankton species discovery. In: *Sciaroff S, Distanto C, Leo M, et al (eds) Image Analysis and Processing – ICIAAP 2022*. Springer International Publishing, Cham, pp. 599–609 (2022) [https://doi.org/10.1007/978-3-031-06430-2\\_50](https://doi.org/10.1007/978-3-031-06430-2_50)
  24. Badredeen Bdawy Mohamed, O., Eerola, T., Kraft, K., et al.: Open-set plankton recognition using similarity learning. In: *Bebis G, Li B, Yao A, et al (eds) Advances in Visual Computing*. Springer International Publishing, Cham, pp. 174–183 (2022). [https://doi.org/10.1007/978-3-031-20713-6\\_13](https://doi.org/10.1007/978-3-031-20713-6_13)
  25. Zhao, W., Chen, F., Huang, H., et al.: A new steel defect detection algorithm based on deep learning. *Comput. Intell. Neurosci.* **2021**, 1–13 (2021). <https://doi.org/10.1155/2021/5592878>
  26. Su, Y., Li, D., Chen, X.: Lung nodule detection based on faster r-cnn framework. *Comput. Methods Programs Biomed.* **200**(105), 866 (2021). <https://doi.org/10.1016/j.cmpb.2020.105866>
  27. Horak, K., Sablatnig, R.: Deep learning concepts and datasets for image recognition: overview 2019. In: *Hwang JN, Jiang X (eds) Eleventh International Conference on Digital Image Processing (ICDIP 2019)*, International Society for Optics and Photonics, vol 11179. SPIE, p. 111791S (2019). <https://doi.org/10.1117/12.2539806>
  28. Li, Q., Sun, X., Dong, J., et al.: Developing a microscopic image dataset in support of intelligent phytoplankton detection using deep learning. *ICES J. Mar. Sci.* **77**(4), 1427–1439 (2019). <https://doi.org/10.1093/icesjms/fsz171>
  29. Chen, T., Li, J., Ju, W., et al.: Object detection and abundance analysis for fountain-flow imaging of marine plankton. In: *OCEANS 2021: San Diego - Porto*, pp. 1–9 (2021). <https://doi.org/10.23919/OCEANS44145.2021.9705862>
  30. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. (2018) arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
  31. Girshick, R.: Fast r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448 (2015)
  32. Liu, W., Anguelov, D., Erhan, D., et al.: Ssd: Single shot multibox detector. In: *Leibe B, Matas J, Sebe N, et al (eds) Computer Vision – ECCV 2016*. Springer International Publishing, Cham, pp. 21–37 (2016) [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
  33. Bilik, S.: Ae-reconstruction-and-feature-based-ad. <https://github.com/boortel/AE-Reconstruction-And-Feature-Based-AD>, open source software available from <https://github.com/boortel/AE-Reconstruction-And-Feature-Based-AD> (2022)
  34. Masci, J., Meier, U., Cireşan, D., et al.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: *Honkela T, Duch W, Girolami M, et al (eds) Artificial Neural Networks and Machine Learning – ICANN 2011*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 52–59, (2011) [https://doi.org/10.1007/978-3-642-21735-7\\_7](https://doi.org/10.1007/978-3-642-21735-7_7)
  35. Pu, Y., Gan, Z., Henao, R., et al.: Variational autoencoder for deep learning of images, labels and captions. In: *Lee, D., Sugiyama, M., Luxburg, U., et al. (eds.) Advances in Neural Information Processing Systems*, vol. 29. Curran Associates Inc, New York (2016)
  36. Bilik, S.: Feature space reduction as data preprocessing for the anomaly detection. In: *Proceedings I of the 27th Conference STUDENT EEICT 2021*, pp. 415–419. Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno (2021)
  37. Hou, X., Shen, L., Sun, K., et al.: Deep feature consistent variational autoencoder. In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp 1133–1141, (2017) <https://doi.org/10.1109/WACV.2017.131>
  38. Makhzani, A., Frey, B.J.: Winner-take-all autoencoders. In: *Cortes, C., Lawrence, N., Lee, D., et al. (eds.) Advances in Neural Information Processing Systems*, vol. 28. Curran Associates Inc, New York (2015)
  39. Bilik, S., Horak, K.: Sift and surf based feature extraction for the anomaly detection. In: *Proceedings I of the 28th Conference STUDENT EEICT 2022 General Papers*, pp. 459–464. Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno (2022)
  40. Rousseeuw, P.J., Driessen, K.V.: A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **41**(3), 212–223 (1999). <https://doi.org/10.1080/00401706.1999.10485670>
  41. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., et al.: Estimating the Support of a High-Dimensional Distribution. *Neural Comput.* **13**(7), 1443–1471 (2001). <https://doi.org/10.1162/089976601750264965>
  42. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422 (2008). <https://doi.org/10.1109/ICDM.2008.17>
  43. Iglewicz, B.: Robust scale estimators and confidence intervals for location. In: *Understanding Robust and Exploratory Data Analysis*, 1st edn., pp. 405–431. Wiley-Interscience, New York (2000)
  44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. (2014) arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) <https://doi.org/10.48550/arXiv.1409.1556>
  45. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 770–778 (2016)
  46. Li, Y., Xie, S., Chen, X., et al.: Faster R-CNN (ResNet50). (2021a) [https://pytorch.org/vision/main/models/generated/torchvision.models.detection.fasterrcnn\\_resnet50\\_fpn\\_v2.html#torchvision.models.detection.fasterrcnn\\_resnet50\\_fpn\\_v2](https://pytorch.org/vision/main/models/generated/torchvision.models.detection.fasterrcnn_resnet50_fpn_v2.html#torchvision.models.detection.fasterrcnn_resnet50_fpn_v2)
  47. Li, Y., Xie, S., Chen, X., et al.: Benchmarking detection transfer learning with vision transformers (2021) arXiv preprint [arXiv:2111.11429](https://arxiv.org/abs/2111.11429) <https://doi.org/10.48550/arXiv.2111.11429>
  48. Olson, R.J., Sosik, H.M.: A submersible imaging-in-flow instrument to analyze nano- and microplankton: Imaging flowcytobot. *Limnol. Oceanogr. Methods* **5**(6), 195–203 (2007). <https://doi.org/10.4319/lom.2007.5.195>
  49. Laakso, L., Mikkonen, S., Drebs, A., et al.: 100 years of atmospheric and marine observations at the finnish utö island in the baltic sea. *Ocean Sci.* **14**(4), 617–632 (2018). <https://doi.org/10.5194/os-14-617-2018>
  50. Kraft, K., Seppälä, J., Hällfors, H., et al.: First application of ifcb high-frequency imaging-in-flow cytometry to investigate bloom-forming filamentous cyanobacteria in the baltic sea. *Frontiers*

- in *Marine Science* **8**, (2021). <https://doi.org/10.3389/fmars.2021.594144>
51. Kraft, K., Velhonoja, O., Eerola, T., et al.: Towards operational phytoplankton recognition with automated high-throughput imaging, near-real-time data processing, and convolutional neural networks. *Frontiers in Marine Science* **9**, (2022). <https://doi.org/10.3389/fmars.2022.867695>
  52. Tkachenko, M., Malyuk, M., Holmanyuk, A., et al.: Label Studio: Data labeling software. (2020-2022) <https://github.com/heartexlabs/label-studio>, open source software available from <https://github.com/heartexlabs/label-studio>
  53. Bilik, S., Baktrakhonov, D., Eerola, T., et al.: Ifcb phytoplankton anomaly dataset (ifcb-pad). (2023) <https://doi.org/10.23729/08b2ac4a-a80d-4e54-85e0-ab3ea46085ec>
  54. Vincent, P., Larochelle, H., Bengio, Y., et al.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. Association for Computing Machinery, New York, NY, USA, ICML'08, pp. 1096–1103, (2008) <https://doi.org/10.1145/1390156.1390294>
  55. Abadi, M., Agarwal, A., Barham, P., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems. (2015) <https://www.tensorflow.org/>, software available from tensorflow.org
  56. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Simon Bilik** received the M.Sc. degree in technical cybernetics from the Department of Control and Instrumentation, Brno University of Technology, Czech Republic, in 2019. He is currently a double degree PhD student with the Machine Vision Group, BUT University and the Computer Vision and Pattern Recognition Laboratory, LUT University. His research field includes applied machine vision and machine learning.

**Daniel Batrakhonov** attained his M.Sc. degree in Computational Engineering and Technical Physics from the School of Engineering Science at LUT University, Lappeenranta, Finland in 2021. Currently, he is a Junior Researcher at the LUT Computer Vision and Pattern Recognition Laboratory, his focus areas include machine learning, image processing, and data analysis.

**Tuomas Eerola** received the M.Sc. and Ph.D. degrees in information processing from the Department of Information Technology, Lappeenranta University of Technology, Finland, in 2006 and 2010, respectively. He is currently an Associate Professor with the Computer Vision and Pattern Recognition Laboratory, LUT University. His research interests include digital image processing, applied computer vision, and deep learning.

**Lumi Haraguchi** is an oceanographer specialized in phytoplankton and microzooplankton ecology. She received her PhD in Biosciences from Aarhus University (DK) in 2018 and is currently working at the Finnish Environmental Institute.

**Kaisa Kraft** works in Marine Ecology Measurements unit of the Finnish Environment Institute. Her work includes high-frequency measurements of phytoplankton community with main focus on imaging. She works on implementation of the imaging technology to Baltic Sea phytoplankton community and the consequent image analysis. Her special interest is in phytoplankton bloom dynamics and the related environmental forcing.

**Silke Van den Wyngaert** is an aquatic microbial ecologist and received her Ph.D. degree from the Swiss Federal Institute of Technology (ETHZ), Switzerland, in 2013. She is currently an Academy Research fellow at the Department of Biology, University of Turku, Finland. Her research interests include plankton ecology, in particular diversity of aquatic fungi and their interactions with phytoplankton.

**Jonna Kangas** graduated as a MSc in Environmental Sciences in 2022 from University of Helsinki. Her thesis project focused on the effects of salinity change on the microbial loop in the Baltic Sea.

**Conny Sjöqvist** graduated as a PhD in Marine Biology in 2015 from Åbo Akademi University (ÅAU), Finland. He is currently working as a researcher in plankton ecology at ÅAU. He was awarded the Title of Docentship in Molecular and Evolutionary Marine Ecology in 2022.

**Karin Madsén** graduated as a MSc. in Environmental and Marine Biology in 2023 from Åbo Akademi University. Her thesis project focused on detection of phytoplankton parasites using imaging flow cytometry.

**Lasse Lensu** is a professor of machine vision and data analysis at Lappeenranta-Lahti University of Technology LUT, Finland. He received his D.Sc. (Tech.) degree in computer science and engineering in 2002 from the Department of Information Technology of LUT. His research interests include machine/computer vision, pattern recognition with machine learning and data analysis. Prof. Lensu is a member of the Computer Vision and Pattern Recognition Laboratory of the Department of Computational Engineering at LUT, and he has contributed to the technology transfer to three spin-off companies from the university.

**Heikki Kälviäinen** is a full professor of Computer Science and Engineering at the Lappeenranta-Lahti University of Technology LUT, School of Engineering Science, Finland. Prof. Kälviäinen is a head of the Computer Vision and Pattern Recognition Laboratory at the Department of Computational Engineering. His research interests include computer vision, pattern recognition, machine learning, and especially applications of digital image processing and analysis. Besides LUT, Prof. Kälviäinen has been a Professor of Computing at Monash University Malaysia, and a Visiting Professor at Brno University of Technology in Czech Republic, Czech Technical University at Prague, and University of Surrey in UK.

**Karel Horak** received the M.Sc. and Ph.D. degrees in technical cybernetics from the Department of Control and Instrumentation, Brno University of Technology, Czech Republic, in 2004 and 2008, respectively. He is currently an Associate Professor with the Machine Vision Group, BUT University. His research interests include machine vision, image processing and machine learning.