LIMNOLOGY
and
OCEANOGRAPHY: METHODS

Limnol. Oceanogr.: Methods 2021
© 2021 The Authors. *Limnology and Oceanography: Methods* published by Wiley Periodicals LLC. on behalf of Association for the Sciences of Limnology and Oceanography.
doi: 10.1002/lom3.10413

Automatic plankton image classification—Can capsules and filters help cope with data set shift?

Rene-Marcel Plonus,[1,2]* **Jan Conradt**,[1] **André Harmer**,[1] **Silke Janßen**,[1] **Jens Floeter**[1]

[1]Institute of Marine Ecosystem and Fishery Science, Faculty of Mathematics, Informatics and Natural Sciences, University of Hamburg, Hamburg, Germany
[2]Present address: Institute of Marine Ecosystem and Fishery Science, Hamburg, Germany

Abstract

The general task of image classification seems to be solved due to the development of modern convolutional neural networks (CNNs). However, the high intraclass variability and interclass similarity of plankton images still prevents the practical identification of morphologically similar organisms. This prevails especially for rare organisms. Every CNN requires a vast amount of manually validated training images which renders it inefficient to train study-specific classifiers. In most follow-up studies, the plankton community is different from before and this data set shift (DSS) reduces the correct classification rates. A common solution is to discard all uncertain images and hope that the remains still resemble the true field situation. The intention of this North Sea Video Plankton Recorder (VPR) study is to assess if a combination of a Capsule Neural Network (CapsNet) with probability filters can improve the classification success in applications with DSS. Second, to provide a guideline how to customize automated CNN and CapsNet deep learning image analysis methods according to specific research objectives. In community analyses, our approach achieved a discard of uncertain predictions of only 5%. CapsNet and CNN reach similar precision scores, but the CapsNet has lower recall scores despite similar discard ratios. This is due to a higher discard ratio in rare classes. The recall advantage of the CNN decreases with increasing DSS. We present an alternative method to handle rare classes with a CNN achieving a mean recall of 96% by manually validating an average of 6.5% of the original images.

State-of-the-art sampling with towed optical devices provides anthropocenic marine planktologists with a wealth of data that even their most recent ancestors could only have dreamed off. Old-school planktologists had to spent hours sitting over the microscope hand-sorting net samples. They were rewarded with snapshots of plankton communities in space and time at the highest possible taxonomic level, sometimes even down to ontogenetic life stages, sex, and clutch sizes (Hansson et al. 1990; Ston et al. 2002; Johansson et al. 2004; Vuorio et al. 2005; Renz and Hirche 2006; Peters et al. 2013).

Modern plankton sampling devices provide information from the other ends of these scales: millions of images at a spatiotemporal resolution of cm and seconds (Davis et al. 1992; Wiebe and Benfield 2003; Benfield et al. 2007) sampled continously over transects 10s (Floeter et al. 2017) or even 100s (Davis and McGillicuddy 2006) of nautical miles long.

The subsequently necessary automatic plankton image classification has followed the trends in machine learning from Support Vector Machines (SVMs; Hu and Davis 2005; Sosik and Olson 2007), later on Neural Networks (NNs, Tang and Stewart 1996) to modern Random Forest (Bell and Hopcroft 2008; Orenstein et al. 2015; Faillettaz et al. 2016) and convolutional neural networks (CNNs; LeCun et al. 2015; Krizhevsky et al. 2017), though the use of manually engineered features such as in SVMs is still relatively common (e.g., Nanni et al. 2019). Since the year 2015, when the Microsoft Research Asia team (He et al. 2015) had won the annual ImageNet challenge (Russakovsky et al. 2015) by reaching an accuracy of 96.4% in classifying high-resolution color images into 1000 different categories, image classification seemed to be solved (Chollet 2017). At first sight, plankton images are no exception, because recent efforts have resulted in > 90% average classification accuracy (Al-Barazanchi et al. 2016; Luo et al. 2018).

However, the taxonomic resolution is also almost always diametrically opposed to the increasing scales, providing

*Correspondence: rene-marcel.plonus@uni-hamburg.des

Additional Supporting Information may be found in the online version of this article.

densities for coarse zooplankton groups such as "jellyfish" or "calanoid copepods" and reaching the family-, or for very distinct organisms the genus-level at best (e.g., *Pseudocalanus* spp. in the Baltic Sea; Möller et al. 2015; Pitois et al. 2018). This is certainly not sufficient for biodiversity monitoring (Batten et al. 2019). However, in many cases coarse groups are suitable for ecological process studies, especially targeting the mesoscale (Floeter et al. 2017) and microscale (Möller et al. 2012; Ohman et al. 2019).

Further on, the specific success of an automatic plankton image classification task depends on a number of factors: first on the desired taxonomic resolution, that is, the research question, and second on technical properties as the number of training images and their distribution among classes (e.g., Orenstein et al. 2015). Additionally, the image quality can have an effect (e.g., how many suspended particles have scattered the flashlight), as the GIGO Principle (Garbage In – Garbage Out) still prevails in times when machines are learning.

Some plankton classes are very abundant while others are scientifically more in focus but rare. Coupled with the usually high intraclass variability and interclass similarity, this leads to the first unsolved problem in real world applications of automatic plankton image classification: the correct identification of rare and/or morphologically similar organisms (Culverhouse et al. 2003; Benfield et al. 2007; Bell and Hopcroft 2008). The second remaining problem of plankton classification with machine learning methods in production mode applications is related to data set shift (DSS; Moreno-Torres et al. 2012), more specifically in form of "covariate shift" (Moreno-Torres et al. 2012; Webb et al. 2018). DSS can be a problem when, for example, a machine learning model fitted to images of one region such as the North Atlantic is applied in an apparently similar region in the adjacent North Sea (Webb et al. 2018). Covariate shift is a specification of DSS and can occur when a model that is fitted to images sampled from one plankton distribution needs to be applied to another plankton community sampled some weeks later at the same location.

One approach to cope with these challenges in the production mode application of machine learning methods in plankton image classification is the introduction of probability thresholds, which discards images with uncertain (i.e., likely erroneous) classifications (Faillettaz et al. 2016). This method leads to considerable improvements in average precision but simultaneously to high discard rates of 30–70% of the original images, which artificially changes their abundances (Luo et al. 2018). As some of the discarded images were correctly identified objects, the recall (i.e., the proportion of the true total number of objects of a class that are correctly predicted in that class) is reduced. The resulting key question is whether any subsequent analyses still yield ecological patterns that resemble the truth (Faillettaz et al. 2016; Luo et al. 2018). This is usually fulfilled for research questions that target common taxa at coarse spatial resolutions. When validated images in the order of magnitude of the test data set are easily obtainable for each new data set, a multiplication factor can be computed from the *F*-score based confusion matrix to calculate postfiltering corrected concentrations (Hu and Davis 2006; Briseño-Avena et al. 2020; Schmid et al. 2020).

However, when the scientific focus is on rare organisms or alpha biodiversity, recall is more important than mean precision and a filtering method may be impedimental.

The second main challenge is the consistency of model performance over time and space, that is, data set drift (Bell and Hopcroft 2008; Al-Barazanchi et al. 2016; González et al. 2017). The Capsule Network (CapsNet) is a recently developed machine learning architecture (Hinton et al. 2011; Sabour et al. 2017), which could overcome this issue. CapsNets group neurons into so called Capsules, which learn specific properties of an object or segment such as size or rotation. This makes the predictions of a CapsNet invariant to the viewpoint, that is, variations in position and orientation, and to variations in scale and lighting. It can theoretically improve the performance on overlapping objects, thus it could be useful to detect, for example, grazing interactions with marine snow particles (Möller et al. 2012). Instead of dropout layers, a CapsNet uses a reconstruction autoencoder for regularization. This autoencoder should be able to reconstruct an object of the predicted class based on the features learned for that class (Sabour et al. 2017; Xi et al. 2017). So far CapsNets have been successfully applied to "baseline" data sets such as MNIST or CIFAR10 (Sabour et al. 2017; Xi et al. 2017; Rajasegaran et al. 2019) but only to a limited number of "real-world" applications such as brain tumor recognition (Afshar et al. 2018).

The theoretical advantages of the CapsNet over a common CNN led us to the assumption that a CapsNet should be able to adapt better to changing field conditions and therefore yield better results in production mode applications. By following González et al. (2017) recommendations for the development of unbiased input data sets reflecting the class distribution in the field, we describe the whole training process for a deep learning CNN and a CapsNet to classify plankton images in 26 different classes. This includes preprocessing, classification, and postprocessing of the images. Subsequently, we apply our models in production mode, that is, without updating the training, to three different North Sea field data sets with increasing temporal and structural distance.

In our analysis, we demonstrate how the filtering method and a CapsNets can help coping with DSS in automatic plankton image classification. Specific research tasks typically focus on predicting broad scale plankton community properties in unseen samples or on classifying each image correctly, also for rare organisms.

To assess whether filtering methods and CapsNets can be customized to successfully cope with data set and covariate shift, we compare two different scenarios: a baseline scenario

(BL) without any filtering and a high precision scenario (P95) with probability filters aimed to maximize precision in a fully automated analyses of plankton communities. Second, we show how to customize the method to maximize the recall for classes, individually, supporting studies focusing on specific classes exclusively. To measure the potential advantage of the CapsNet, we compare the results of a simultaneously trained CNN with those of our CapsNet.

Materials and procedures

Description of instrument

We used a VPR (Seascan, Falmouth, Massachusetts, U.S.A.) mounted on a MacArtney TRIAXUS ROTV which was connected to a research vessel with a fiber optic cable to record high-resolution images of in situ plankton organisms. The ROTV was towed at a speed of 8 knots (4.1 m s^{-1}) with a three-degree lateral offset to lessen any disturbance from the vessels wake. During most transects, the ROTV was undulating with a vertical speed of 0.1 m s^{-1} from ~ 4 m below the sea surface to ~ 8 m above the seafloor. The VPR was equipped with a high-resolution digital camera (Pulnix TM-1040) that records up to 25 fps. A synchronized strobe (Seascan—20 W Hamamatsu xenon bulb) provided the illumination for the images at a pulse of 1 μs. The resulting images consist of 1392 × 1024 pixels with a size of 9.0 × 9.0 μm. The chosen field of view was 24 × 24 mm with a focal depth of ~ 60 mm at 246 mm from the lens. The image volume was thus 34.93 mL. Imaged particles were extracted as regions of interest (ROIs) by the Autodeck image analysis software (Seascan) and saved to the computer hard drive as TIFF files.

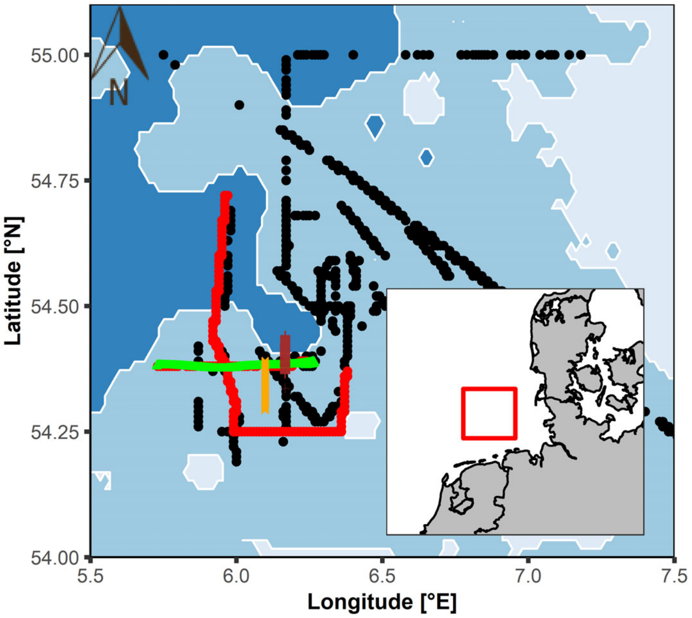


Fig 1. Core area of the sampling transects with the VPR in the North Sea. The red transect provided ~ 90% of the labeled training images. Black: Remaining training set; field sets—green: FS446 (#1; 2015; 55,302 images); brown: FS466 (#2; 2016; 7798 images); orange: FS534 (#3; 2019; 31,848 images). Blue shading: depth contours from 20 to 50 m.

Description of hardware and software

The German Climate Computing Center (DKRZ) provided computing time with the High Performance Computer System for Earth System Research (HLRE-3, Mistral), which consists of more than 3000 compute nodes, providing a peak compute performance of 3.6 PFLOPs and was used to train our models. We used two Mistral computing nodes (2x 18-core Intel Xeon [E5-2695 v4] with a single Nvidia Tesla V100 GPU and 512 GB RAM) for the training of our models (https://www.dkrz.de/up/systems/mistral/configuration, 04 August 2020. 16:35:40).

Training and application of deep learning models were done with a GPU supported Tensorflow (Abadi et al. 2015) backend for Keras (Chollet 2015) under Python 3.7 (Van Rossum and Drake 2009). Subsequent data analyses were done with the statistical package R (R Core Team 2020). Visualizations were created using ggplot2 (Wickham 2016) while data management was mainly done with dplyr (Wickham et al. 2020). We calculated the *t*-test modified by Dutilleul (Dutilleul et al. 1993) using SpatialPack (Osorio and Vallejos 2019). The Bray-Curtis (BC) dissimilarity (Bray and Curtis 1957) between the validated training set and the predicted field sets was calculated using the implementation in vegan (Oksanen et al. 2019).

Field sampling

We used ~ 124,000 hand sorted and labeled images to train our models, of which ~ 90% were sampled on the FS Heincke cruise HE446 on the 4th of June 2015 between 07:00 and 13:00 UTC. The remaining 10% of the images originate from the period June to August of 4 yr (2014–2017) and cover all 24 h of a day. Most of our images (94%) were sampled in the inner German Bight of the North Sea, including the three unlabeled field data sets (Fig. 1) which we used in our production mode analysis. The remaining 6% originated from the Baltic Sea and provided images for the classes “eggs” and “larvae” which were not represented otherwise.

Evaluation of the performance consistency of our final classifiers in production mode was done using the three field data sets. The similarity of the unvalidated classifications (i.e., predictions) of the field data sets and the training set (TS) was assessed calculating the BC dissimilarity. Field set 1 (FS446) originated from the same HE446 cruise in 2015 as the majority of our training images. However, the field set was sampled in the morning from 06:00 to 09:00 (UTC, 55,302 images). The second field set (HE466) was sampled in June 2016 between 18:00 and 19:00 (UTC, 7798 images). The third field set (HE534) was sampled in June 2019 from 11:00 to 12:00 (UTC, 31,848 images). All field set model predictions were manually checked and if necessary corrected by a human zooplankton expert to obtain the “true” classification.

Image preprocessing

As CNNs require equally sized images, based on our most common ROI size we chose a size of 240 × 240 pixel. Smaller images were extended by placing the original ROI image in the center and adding pixels. The new pixel values were set to the median of the ROI pixel values since most ROI images were mainly filled with black background. ROI images that were greater than 240 pixels in one or both dimensions were first squared by adding pixels with median values to keep the proportionality of the objects before rescaling them to 240 × 240 pixels. Initial experiments showed that classification accuracy did not benefit from colored images, so we transformed our images to grayscale by multiplying the RGB values with 0.299 (R), 0.587 (G), and 0.144 (B). The resulting matrix was replicated two times to create the required three channel image input format. When fed to the model, images greater than 240 pixels were reduced to 240 pixels.

Images were fed to the model in small batches using an Image Data Generator function provided by Keras. The batch size was adapted to the respective model and image set (1–40). Since deep learning models usually perform better with homogeneous, small values (Bishop 1995) all pixel values were divided by 255. Data augmentation was applied during the training but not in the validation and test step. Images were rotated, shifted in both directions, sheered, zoomed, and horizontally flipped randomly with fill mode set to "nearest." This was done to increase the generalization of the deep learning model by providing slightly altered images during each training cycle (= epoch).

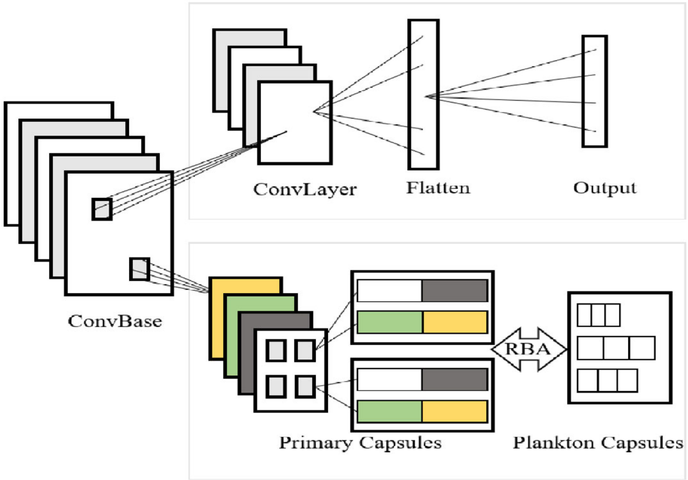


Fig 2. Our model architectures. Both models were based on the ConvBase of the Xception V1 (for details, see Chollet 2017). We changed the input size to 240 × 240, so the output of the ConvBase was a feature stack of 2048 filter maps with size 8 × 8. CNN: a convolutional layer with kernel size 3 × 3 returning 2560 filter maps followed by a flatten layer. The final softmax layer had length 26 (for 26 classes). CapsNet: a convolutional layer with 240 kernels of 6 × 6 and strides 2. The output was grouped into 80 primary capsules with 24 dimensions, each of which represented one property of the feature learned by the respective capsule. Plankton capsules returned 26 capsules (one for each class) with 16 dimensions (one for each property) after three cycles of routing-by-agreement.

While CapsNets do not necessarily need data augmentation to achieve the performance of similar CNNs trained with data augmentation (Jiménez-Sánchez et al. 2018), data augmentation nevertheless can increase the performance especially for small classes (Toraman et al. 2020). Thus, we also applied data augmentation during the training with the CapsNet.

Automated image classification

Workflow

We combined a two-step training procedure suggested by Lee et al. (2016) and the application of different filtering thresholds, as suggested by Faillettaz et al. (2016), to optimize our model performance. In step 1, the model was trained with a balanced data set and rated according to the performance with a balanced test set, both of which were subsets of the imbalanced labeled data set. On this basis, we continued to train the same model with an imbalanced training set, using the final weights from step 1 in the initialization and all available labeled images. As is common practice in deep learning, we split the entire data sets into training-, validation-, and test-subsets. Based on the predictions for the imbalanced test set, we calculated filter values which can be applied to tailor the results for specific research questions in production mode, that is, application to new field data sets without updating the training procedure.

Convolutional neural network

We used the convolutional base (ConvBase) of the Xception V1 model with weights pretrained on ImageNet available for download using the Tensorflow backend from Keras (https://keras.io/api/applications/xception/, 10 December 2019. 15:06:15). The input size was changed to 240 × 240 pixel from 299 × 299 pixel. We added an additional convolutional layer (ConvLayer—SeparableConv2D) using the Keras functional API before the flatten operation and the final Dense-Layer. The ConvLayer had a convolutional window with kernel size 3 × 3 and padding set to "same." We chose "Rectified Linear Unit" as activation function. The resulting filter stack of 2560 filter maps with size 8 × 8 was flattened and the final Dense-Layer with softmax activation was used to classify the images into 26 different classes. The final model had 30,385,218 parameters (Fig. 2).

Training the models

In a first training step, we used only 100 images of each class (2600 images in total) for the training set and 10 images for validation (260) and testing (259), respectively. The smallest class had only nine test images. The validation set is used to monitor the ability of the model to generalize during the training process, while the test set is a final evaluation prior to the application in production mode. ConvLayers "learn" by applying small weights to each input. Those weights store the "learned" information. We successively adapted more layers of the pretrained ConvBase to our images during training, starting with the topmost (last) layers, going deeper in each successive phase (Table 1). This is called "transfer learning" (Pan and Yang 2010; Kornblith et al. 2019). The Keras callback "ReduceLROnPlateau" was used with patience 2 and factor 0.6 and the weights of the best model achieved during training were saved by another callback "ModelCheckpoint." Using the Adam optimizer (Kingma and Ba 2014) and a categorical cross-entropy loss function, the model was trained with an initial learning rate of 2×10^{-5} (CNN) or 5×10^{-5} (CapsNet), using accuracy for evaluation.

In a second training step, the same model was initialized with the final weights from step 1 and trained on a heterogenous data set. The distribution of the training images represented the distribution observed in the labeled data set (Table 2). Eighty-four percent of all images in a class were used as training set and 8% as validation and test set, respectively. The smallest class (echinodermata) had 100 (0.1%) training images while the largest (marine snow) had 68,311 (65.7%). We used class weights (CW; Eq. 1) to account for this imbalance:

$$\mathrm{CW}_i = \log\left(\frac{N_{\max}}{N_i}\right) \tag{1}$$

The CW of class i was calculated as natural logarithm of the ratio of the maximum number of images over all classes ($N_{\max}$) and the number of training images of class i (N_i). The CW of the largest class marine snow was set to 1 and the CWs of the other 25 classes increased logarithmically with decreasing number of available training images up to a factor of 6.5 for the smallest class "echinodermata." Again, we used "transfer learning" to benefit from the features learned during the first training step, especially in less abundant classes.

The first training needed a computing time of ~ 20 min while second training required ~ 24 h (1440 min) for the CNN and ~ 21 h for the CapsNet.

As CNNs are a gradient-based method, the chosen starting point may be crucial for the final fit of the model, and one way of assessing and reducing the effect of start conditions are multistart approaches (Subbey 2018). We repeated the first training step 100 times with randomly changed image sequences fed into the CNN. The second training was only performed once based on the model from step 1 which achieved the best test accuracy. Repeating the second training step was not feasible due to long computing times.

The same best step 1 CNN model was used to train the CapsNet. Before we started the training with the heterogenous data set in the same way as described for the CNN, we repeated step 1 once in a reduced form (Fig. 3) to adjust the weights of the last three layers of the ConvBase to the new Capsule-Layers, which replaced the Dense- and Flatten-Layers used in the CNN (Fig. 2).

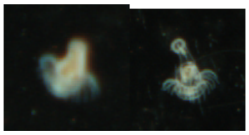
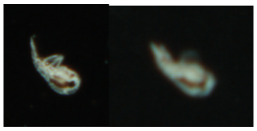
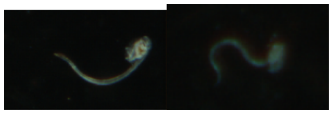
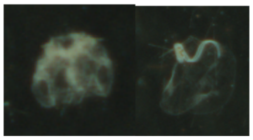
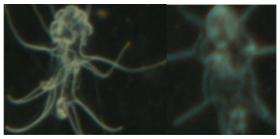
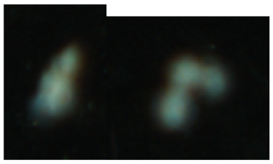
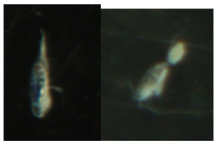
Model performance

To compare the different models, we calculated the classwise as well as average training-, validation-, and test-accuracies (Acc), which is the percentage of correctly classified images. Training-, validation-, and test-accuracies are related to the respective image subsets. In case of the balanced data set for the first training step that means 100 images per class for training and 10 images for each, validation and testing. While this is sufficient for homogenous data sets such as the one used during the first training, accuracy fails to account for the imbalance in a heterogenous data set as used during the

Table 1. Models were trained on 26 classes, including 2 classes with none-living objects ('marine snow' and 'rod') and 2 classes for unrecognized objects ('unknown' and 'blurry'). The numbers for the training- (TS) and field sets (FS) correspond to the 'true' distribution obtained by manual classifications.

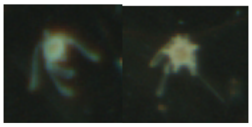
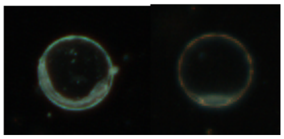
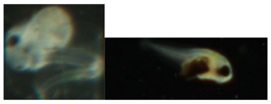
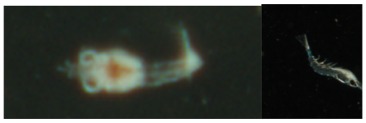
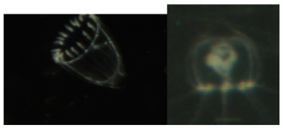
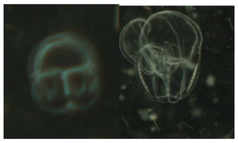
Model	Training step	Training phase	Epochs	ConvBase trainable layers
CNN	1	1	7	2
CNN	1	2	7	11
CNN	1	3	7	20
CNN	1	4	7	29
CNN	2	1	3	2
CNN	2	2	3	11
CNN	2	3	3	20
CNN	2	4	3	29
CNN	2	5	5	35
Cap	1	1	7	2
Cap	2	1	3	2
Cap	2	2	3	11
Cap	2	3	4	20
Cap	2	4	3	29

Table 2. Training procedure for the CNN and the CapsNet. Training the CNN was initialized using weights pre-trained on ImageNet, while the training of the CapsNet was initialized using the weights received at the end of step 1 with the CNN. Due to overfitting, the CapsNet was trained only for 4 phases in step 2, while the CNN was trained for 5 phases. The ConvBase had 40 layers in total.

Class	Label	TS (*N*)	FS446 (*N*)	FS466 (*N*)	FS534 (*N*)	Example image
Actinotrocha	act	208	0	5	11	
Amphipods	amp	241	0	0	0	
Appendicularia	app	545	28	147	198	
Appendicularia with house	app	837	270	204	899	
Bipinnaria	bip	473	79	11	8	
Blurry	blu	1187	101	1313	1014	
Copepods	cop	2258	151	627	2219	

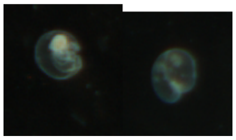
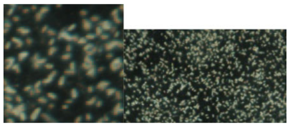
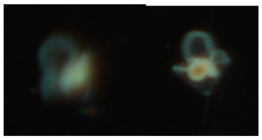
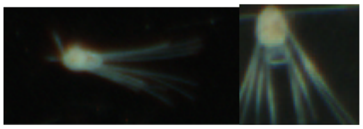
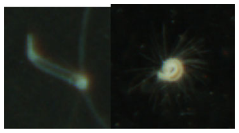
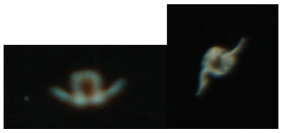
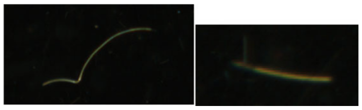
(Continues)

Table 2. Continued

Class	Label	TS (*N*)	FS446 (*N*)	FS466 (*N*)	FS534 (*N*)	Example image
Diatoms	dia	6984	3190	346	16	
Echinodermata	ech	100	0	0	0	
Eggs	egg	416	5	0	0	
Larvae	lar	230	15	7	1	
Malacostraca	mal	376	22	43	87	
Medusae	med	394	76	144	24	
Mnemiopsis	mne	739	144	1	7	

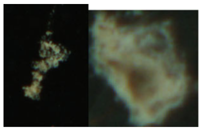
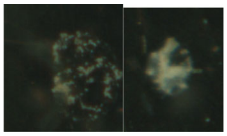
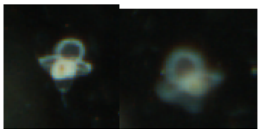
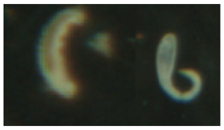
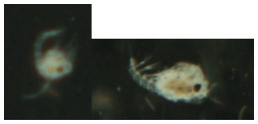
(Continues)

Table 2. Continued

Class	Label	TS (*N*)	FS446 (*N*)	FS466 (*N*)	FS534 (*N*)	Example image
Noctiluca	noc	834	348	20	3696	
Phaeocystis	pha	224	0	0	0	
Pilidium	pil	142	0	56	19	
Pluteus	plu	14,713	9861	212	1343	
Polychaeta	pol	802	363	16	22	
Pteropods	pte	587	0	0	0	
Rod	rod	264	2034	814	22	

(Continues)

Table 2. Continued

Class	Label	TS (*N*)	FS446 (*N*)	FS466 (*N*)	FS534 (*N*)	Example image
Marine snow	sno	68,311	37,675	3578	20,519	
Unknown	unk	509	208	176	758	
Veliger	vel	249	0	34	47	
Worms	wor	2103	705	37	913	
Zoea	zoe	274	27	7	25	

second training step (González et al. 2017). Therefore, we also calculated the F1 score (Eq. 2), which is the harmonic mean of the classification metrics precision (P—"purity"; Eq. 3) and recall (R—"completeness"; Eq. 4) and is more sensitive to wrong predictions in highly skewed data (He and Garcia 2009).

Precision and recall are calculated using true positives (TP), false positives (FP, type I error), and false negatives (FN, type II error). A correctly identified copepod image in the copepod class is a TP. A copepod identified as a diatom is a FN for the copepod class and at the same time a FP for the diatom class. FPs and FNs are class-specific and make sense only from the viewpoint of the respective class. Images which truly belong to a class, though they are sorted into other classes, count as FNs, while all images which do not belong to a class, though they are sorted into that class, count as FPs. Precision is the proportion of correctly classified objects in a predicted class and recall is the proportion of the true (i.e., manually labeled) number of objects of a class that are correctly predicted in that class.

$$\mathrm{F1}_i = \frac{2 \times P_i \times R_i}{P_i + R_i} \tag{2}$$

$$P_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i} \tag{3}$$

$$R_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} \,(= \mathrm{Acc}_i) \tag{4}$$

Probability filtering and top-k predictions

All 100 models of the first training step were used to create predictions for an identical test set of 259 images to assess the

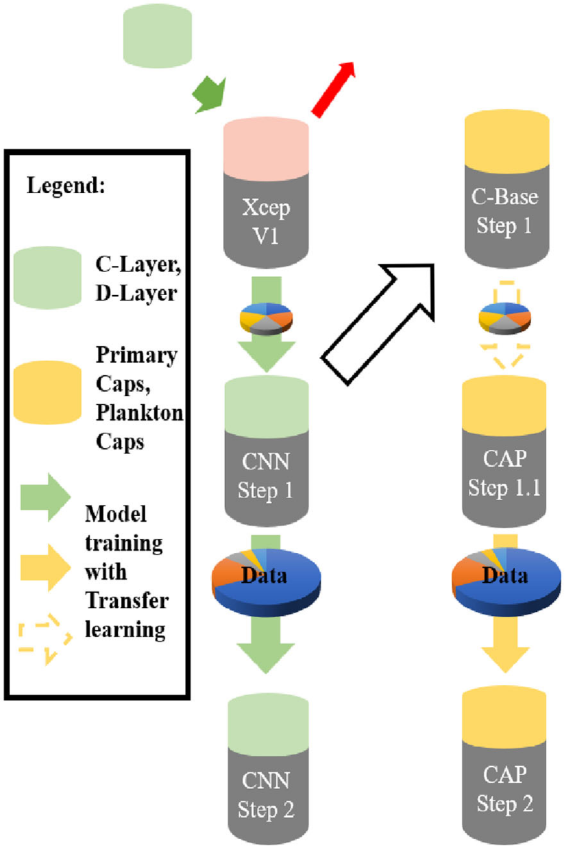


Fig 3. Schematic visualization of model training. Training step 1 was repeated for the capsule network in a reduced form (Table 1), which is indicated by the dashed arrow. Both our models shared the ConvBase with the Xception V1 model, but we replaced the final classifying layers with our own choice of layers as indicated by the colored part of the cylinders and the red and green arrow in the top left corner. CAP, capsule network; C-Base, convolutional base; C-Layer, convolutional-layer; CNN, convolutional neural network; D-Layer, dense-layer; Xcep V1, Xception V1.

final model performance. After the second training step, a labeled test set of 9903 images was used to validate the model performance on new, unseen data sets. The three unlabeled field sets were only predicted (i.e., classified) once by the final model after the second training step, since initial results suggested that field set predictions after step 1 were not meaningful. For a given image, CNNs compute a probability for each class. In an ideal case, the class with the highest probability resembles the true taxonomic class of the imaged object. The filtering method of Faillettaz et al. (2016) takes advantage of those probabilities and accepts only predictions above a user specified threshold. The assumption is that TPs have a higher probability than FPs and thus more wrong than correct predictions are discarded, ultimately increasing the precision. We used the labeled test set to calculate probability filters for each class individually. All images assigned to a class (TP + FP) were sorted in increasing order of their probability to belong to that class. All images with a lower probability than the chosen threshold but correctly classified as class *i* were then nevertheless treated as FNs and thus decreased the recall and subsequently the F1-score. Precision was calculated using only images with a higher probability than the chosen threshold, since the "purity" of a class can only be affected by images assigned to this class. Therefore, FPs with a lower probability than the chosen threshold had no influence on the calculated precision. This method can of course only be applied with data sets that have been manually validated and labeled to obtain the "true" classifications.

Tailoring to specific research questions

During the validation of the final model, the filters were stepwise increased from the lowest to the highest probability and the corresponding classification metrics were calculated. This enables the researchers to pick their favorite set of class-specific filters along the trade-off continuum between the best average precision and the best recall. In Luo et al. (2018), classes with $n < 25$ of 75,000 randomly drawn images were excluded to achieve a mean precision of 90.7%. This threshold ($n = 25$) divided classes into a "pure" (precision > 90%) and an "uncertain" (precision < 90%) group. In addition to the class-wise filters aiming to maximize the precision (P95), we chose for each model and field set a class unspecific threshold (*t*) of *n* images to separate a "pure" group of classes (mean precision > 90%) from a "uncertain" group of classes (mean precision < 90%). In a larger scale community distribution-oriented research question, this sorts classes classified on a human-like level into the "pure" group and leaves classes with poor performances in the "uncertain" group (Luo et al. 2018). The threshold was chosen based on the sum of the TPs and the FPs of the predictions of the three field sets (n = TP + FP). One hundred bootstraps were performed using randomly chosen 75% of the images that remained after P95-filtering to increase the confidence in the chosen threshold.

However, a reduced recall is problematic if rare taxa like fish larvae are specifically in the focus of the research question. Thus, we assessed whether the deep-learning practice of the Top-5-Accuracy can be used to increase the recall and significantly reduce the time needed for manual classification. We accepted the *k* highest predictions for each image, stepwise increasing *k* from 2 to 5, and treated an image as "correct classified" if the correct class was assigned within one of the top-*k* probabilities. Subsequently, the user has to manually classify all top-*k* images in the classes which are in focus of the research question. In this case, the trade-off between the recall and the number of images that have to be manually classified is of particular interest.

Representativeness of field set classifications

For research questions involving the detection of ecological patterns in high frequency data sets, particularly for common taxa, precision could be more important than recall (Faillettaz et al. 2016). This arises because the distribution of images could resemble the field plankton community even when large fractions of images that cannot be classified with sufficient certainty are discarded.

As in Faillettaz et al. (2016), we tested the spatial distributions of our filtered predictions against the spatial distribution of the manual classification using the *t*-test modified by Dutilleul et al. (1993). We aggregated our data in 1 m depth bins and by Latitude (0.01 decimal degree [DD] bins) for North–South transects or Longitude (0.01 DD bins) for West–East transects. Since the filtered predictions are per definition a subset of the original data set, we compared relative abundances instead of the absolute ones as Faillettaz et al. (2016) suggested.

Results

Model training

The dynamics of training and validation accuracy of the 100 models during the first CNN training step were slightly different each time, despite the fact that all were trained in the same way and with the same training images. At the end of step 1, the training accuracies (mean: 75%) usually slightly exceeded the validation accuracies (mean: 71%; Fig. 4). Further increasing the number of adjustable layers or training epochs only led to strong overdispersion, which indicates decreasing generalization of the model. Even though the general trends during the 100 training runs were similar, the final test accuracies ranged from 54.4% to 84.6%, indicating that different runs produced different convergence progressions and therefore different outcomes. The oppositional pattern of validation- and trainings-accuracy between phase 1 and all following phases was probably due to the fact that the model was trained to the verge of overfitting in each phase and thus was already close to overfitting when training started in phase 2 (and following phases).

The training progress during the second step using the heterogenous data set differed between the CapsNet and the CNN. The training and validation accuracy of the CNN increased gradually to ~ 95% (Acc_{Tr} = 95.2; Acc_{Val} = 94.9). The final test accuracy was 94.67%. The CapsNet instead deteriorated during the last of the five phases, even though more weights were released for training compared to the phase before. While training (Acc_{Tr} = 95.9%) and validation (Acc_{Val} = 94.3%) accuracy were similar at the end of phase 4, the validation accuracy decreased to Acc_{Val} = 89.5% in phase 5 while the training accuracy increased further to Acc_{Tr} = 96.3% (Fig. 5). The CapsNet reached a slightly lower test accuracy of 89.72%.

Test set predictions

Classes with a high mean test accuracy over all 100 models in step 1 had small confidence intervals (CIs; 95%), while classes with lower mean test accuracies could range from 0% to 80% correctly classified images, depending on the model run, even though all classes were trained with the same amount of 100 images (Fig. 6).

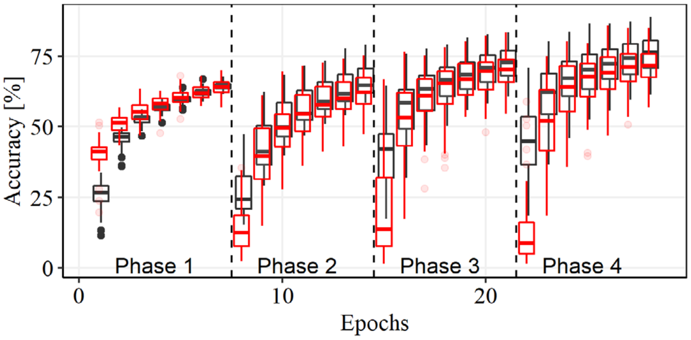


Fig 4. Convolutional neural network training step 1. The vertical lines separate the different training phases 1–4 during the first step (training with homogenous data set), where successively more layers were trained in each phase. Black boxplots: training accuracy of 100 models; red boxplots: validation accuracy of 100 models.

In step 2, classes with high abundances generally achieved high F1-scores, whereas the opposite was not true as low abundant classes could have low, medium or high F1-scores (Fig. 7). In general, the CNN achieved better results than the CapsNet after step 2. However, the CapsNet outperformed the CNN in four classes ("diatoms," "echinodermata," "noctiluca," and "pteropods"), at least in precision and the F1-score. Only the "marine snow" recall of the CapsNet was superior to the CNN. Both models had difficulties with the class "rod," which contains unidentified elongated objects. Another common weakness was the "unknown" class with low recall scores (Fig. 7). Most of the images labeled as "unknown" by a human are recognized as a specific class by both models, mainly as "marine snow" or "appendicularia with houses."

Image filtering

All classes shared a common pattern in regard to the assigned probability filters: at a high threshold, precision was high while recall was low. Thus, only correct classifications were accepted at the cost of discarding most of the correct, less confident classifications together with the wrong classifications. With decreasing probability filters, this was reversed at some point since more and more correctly identified images of the respective class were kept, while simultaneously the chance increased that incorrect classifications were kept as well. As long as the recall was close to 0, the F1-score tended to follow the recall. This was due to the fact, that the harmonic mean (F1) tends to be 0 as soon as one of the components is 0 (recall, Fig. 8). We selected class-specific filters aiming to achieve at least 95% precision in each class. All of the following results were based on these set of filters (P95).

The final filters varied from 36.44% to 92.36% (CapsNet) and from 21.17% to 98.65% (CNN), depending on the class. In general, after filtering the CNN was still superior to the

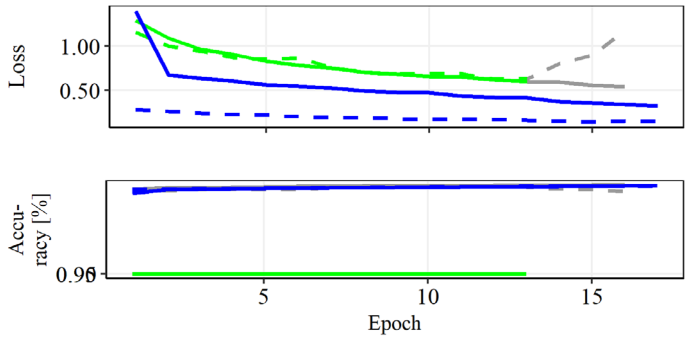


Fig 5. Training step 2, heterogenous data set. Upper panel shows the loss, which is an index of the difference between prediction and truth. The lower panel shows the accuracies as for training step 1. Solid: during training; Dashed: during validation; Green: CapsNet; Blue: CNN; Gray: period of overdispersion, the changes during these epochs were not included in the final model.

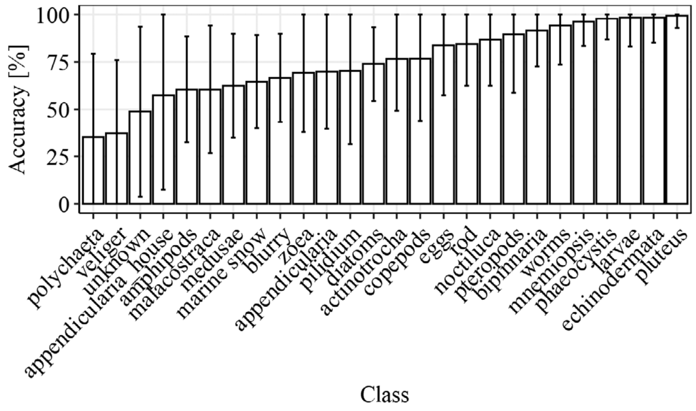


Fig 6. Increasingly sorted mean test accuracies for the 26 classes based on the predictions of the 100 models trained during step 1. Error bars: 95% CI.

CapsNet. However, for single classes the results of the CapsNet could overcome those of the CNN (Fig. 9).

Applying the filters to our test set increased the mean precision of the CNN by 14% from 84% to 98% and of the CapsNet by 15% from 78% to only 93%, as six classes did not achieve the target of 95% precision. Seven percent of all predictions had to be discarded using the CapsNet to maximize precision (5% for the CNN).

Field set predictions

The BC dissimilarity for the field set predictions, not the manually validated FSs, confirmed as expected, that FS446 (BC_{CNN} = 0.45; BC_{Cap} = 0.44) was closest to the TS as it was sampled in the same geographical region 12 h after the majority of our training images. FS534 (BC_{CNN} = 0.69; BC_{Cap} = 0.67) was closer to our training data and to FS446 than FS466 (BC_{CNN} = 0.92; BC_{Cap} = 0.92). Thus, DSS is highest for FS466, lowest for FS446, and in-between for FS534 (Supporting Information Fig. S1).

With increasing DSS, the threshold (t) to separate "pure" from "uncertain" classes for the CapsNet increased according to an exponential function of distance (Fig. 10):

$$t = a \times e^{(b \times BC)} \tag{5}$$

with a = 0.64 and b = 5.21. The simulated thresholds for FS466 followed a bimodal distribution. As the two groups were clearly separated, we chose to include only the higher group of thresholds in the estimation of the model. Therefore, it is less likely for the model to underestimate the true threshold. The observed thresholds (t_{446} = 5; t_{466} = 65; t_{534} = 25) were close to the average simulated thresholds (t_{446} = 3; t_{466} = 65; t_{534} = 23). No reasonable relationship could be established for the CNN and simulated and observed thresholds did not match either.

Filtering generally increased the mean precision and reduced the mean recall as expected. Excluding three nonebiological classes from the analyses, namely "blurry," "unknown," "rod," and additionally "marine snow," the thresholds between "pure" (mean precision > 90%) and "uncertain" classes were always approximately three times higher for the CNN compared to the CapsNet, for example, in FS446 all classes with five assigned images by the CapsNet already belonged to the "pure" group while the CNN had to assign at least 15 images to all classes to reach a mean precision > 90% in the "pure" group (Table 3).

Both models successfully detected a similar amount of classes in the field sets, but the CapsNet predictions had more classes contribution to the "pure" group compared to the CNN. So overall, the CapsNet was better in the generation of "pure" groups. The CapsNet predicted less images in classes, which were not occupied in the field set (n_{true} = 0 and n_{pred} > 0), thereby creating so-called empty classes with only FPs. In the predictions of the field set least similar to the training set (FS466), neither model achieved a mean precision > 90%. The selected threshold only maximized the mean precision to 87% for the CNN (one "pure" class) and to 76% for the CapsNet (three "pure" classes, Table 3).

The recall of the CapsNet was always lower compared to the CNN, but this CNN advantage was reduced by increasing DSS. Neither model dominated the other one regarding the discard ratio, that is, the number of images that had a lower probability than the class-specific filter for certain predictions (P95) compared to uncertain predictions. The discards ranged from 3% to 45% for the CNN and from 5% to 41% for the CapsNet (Table 3).

For illustrative purposes, we will give an interpretation of the first row in Table 3 (predictions of FS446 by the CNN). The field set included 18 classes that were also present in the training set: prior to P95-filtering, the model predicted 22 classes including four empty classes (TP = 0). Twelve classes included more than 15 predicted images (t = 15). Those 12

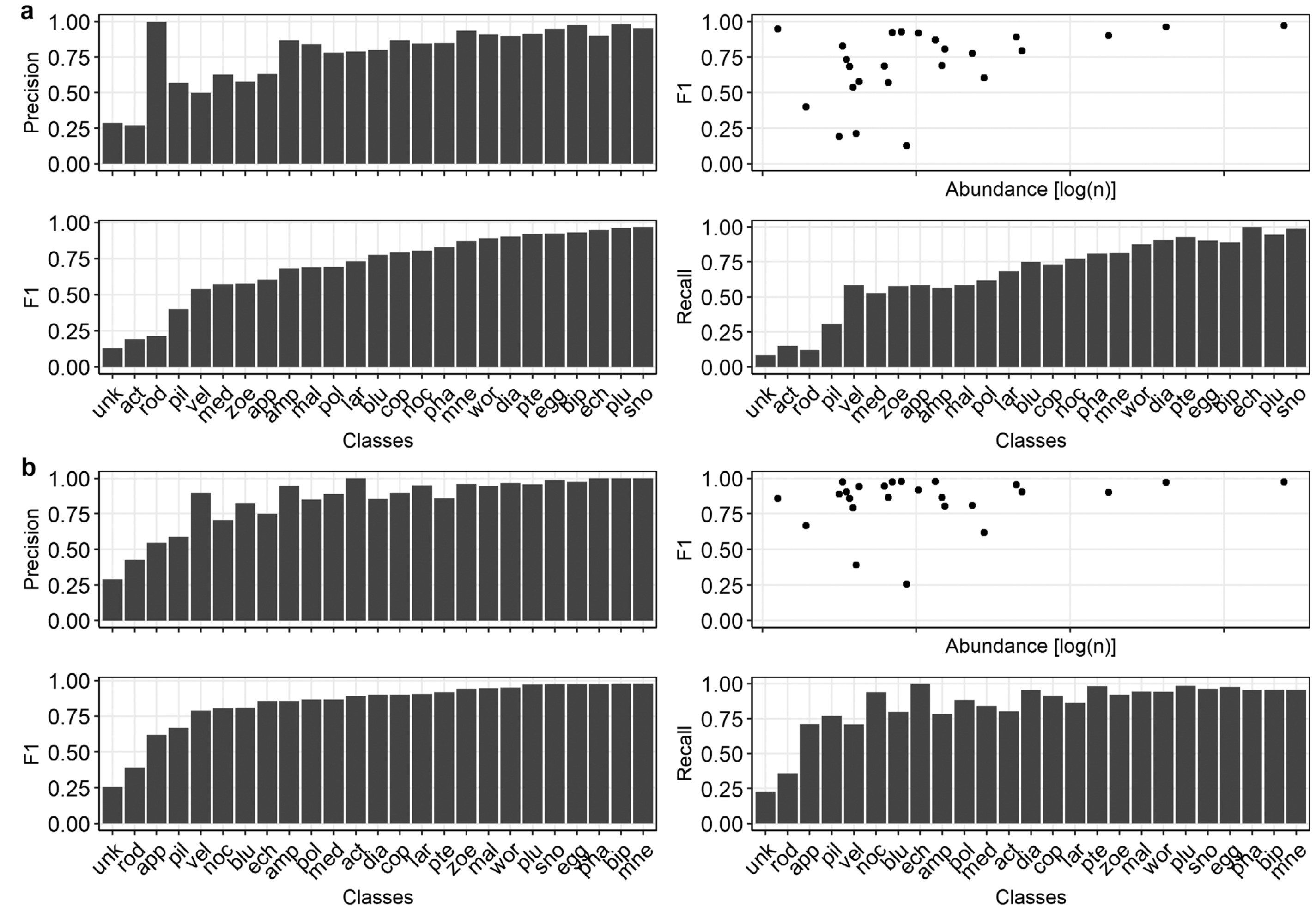


Fig 7. *F*-metrics (F1, precision, and recall) by class for the CapsNet (**a**) and the CNN (**b**). Classes were sorted by increasing F1-score from left to right. Class keys were presented in Table 2. The upper right panel in each plot presents the F1-score in relation to the abundance.

classes had a mean F1-score of 77%, a mean precision of 77%, and a mean recall of 80%. After applying the P95-filter set, 17 classes remained including now only three empty classes. The "pure" group of classes (TP + FP > 15) included eight classes of which none were empty (TP = 0). Those had a mean F1-score of 89%, a mean precision of 94%, and a mean recall of 87%. Only 3% of the images belonging to the 18 "true" classes (TP > 0) were discarded after filtering.

Top-*k* predictions

We investigated the relationship between *k* and the mean recall based on the predictions for FS446 (*n* = 55,302). The recall scores of the CNN always exceeded those of the CapsNet and simultaneously the number of images to validate manually was always lower. We therefore only present the results for the CNN. With *k* = 2 the mean recall increased from 63% (Supporting Information Table S1) to 93% (Table 4), while on average 7.8% of the images had to be validated. Only three classes required manual classification of more than 10% of the original data set images ("diatoms," "pluteus," and "snow"), but those were the most abundant classes. The majority (12 classes) required less than 3% of the original data set to be manually classified. With *k* = 3, the increase in recall (+2.9%) was similar to the increase in images (+3.3%), but further increasing *k* was less effective. We therefore selected *k* = 3 for all field sets.

The classes "blurry" and "unknown" included per definition a wide range of different, unidentified objects which made them scientifically uninteresting. They were therefore not included in the analyses. We also excluded the class "marine snow" because of the extraordinary size. For the remaining classes, the mean recall with *k* = 3 exceeded 90% for the low-shift data set (FS446; 96%, *n* = 17,318) and medium-shift data set (FS534; 95%, *n* = 9557), while the mean recall for the high-shift data set (FS466; *n* = 2731) was only 86%. The Supporting Information includes a complete table with all classes and

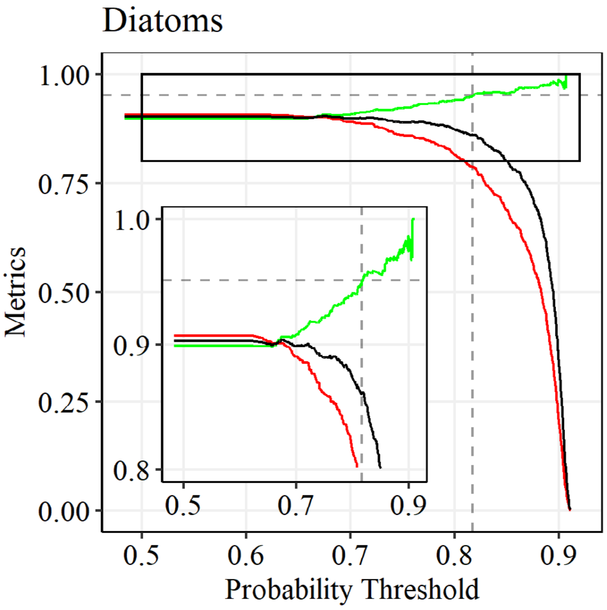


Fig 8. Example plot (class "diatoms") of a filter selection. Probability filters reflect the confidence of the model in the predicted class. The subpanel is the subsection from the whole plot where the metrics reach values ≥ 80%. Green: precision; Red: recall; Black: F1-score; Gray dashed lines: probability at which precision reaches 95%.

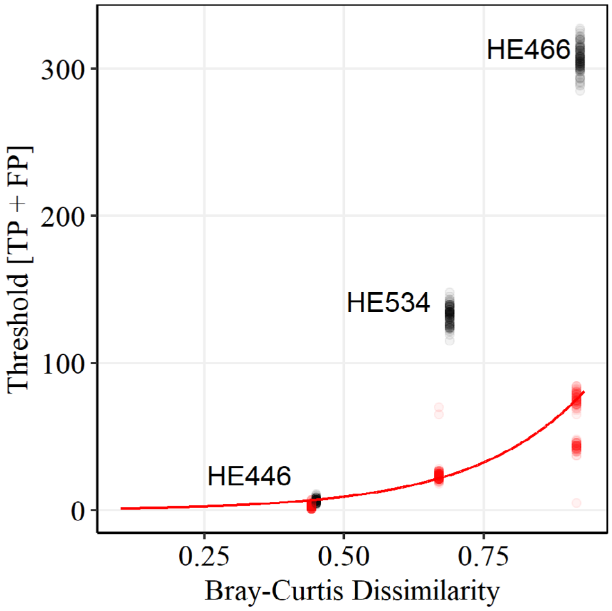


Fig 10. Thresholds to differentiate between "pure" and "uncertain" classes based on the BC dissimilarity between the three field sets and the training set. For each field set, 100 bootstraps were performed using randomly chosen 75% of the images remaining after P95-filtering. The labels might be oriented toward the black dots but are equally true for the red dots. Black: CNN; Red: CapsNet.

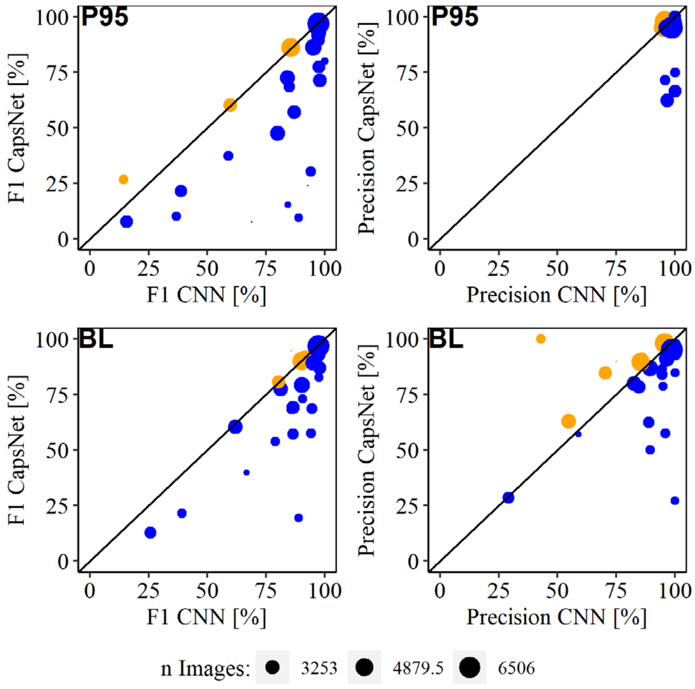


Fig 9. Comparing the F1-score and the precision between the CapsNet and the CNN with (P95) and without (BL) filtering the predictions. The size of each dot represents the logarithms of the abundance of the respective class in the labeled test set. Upper panels: with filtering; Lower panels: without filtering; Left panels: F1-score; Right panels: precision; Orange: CapsNet superior to CNN; Blue: CNN superior to CapsNet.

field sets (Supporting Information Table S2), here we described only the results for the first field set (FS446) in detail.

Only two classes had a recall below 90% ("malacostraca" and "medusae"). All other classes, even the rarest, had a recall above 95%. For especially rare classes like "eggs" (n = 5) and "larvae" (n = 15), less than 1% of the original data set needed manual validation to achieve a recall score of 100% at k = 3. However, rare classes usually had lower ratios between TPs and FPs compared to more abundant classes. An exception from this trend was the class "polychaeta" with n = 363 and a ratio between TP and FP of 1 : 29. Thus, ~ 20% of the original data set needed to be manually validated in order to achieve 97.5% recall for this class (Table 5).

Spatial distributions

We calculated Dutilleuls modified t-test to assess whether our P95 filtered model predictions were representative for the true plankton community in our field sets. While $p < 0.05$ was sufficient to accept the representativeness of a class prediction, we generally assumed the model with the lower p value to be superior. While the CapsNet was superior to the CNN in 11.1% of all classes in FS446 (low DSS), this increased to 21.1% in FS534 (medium DSS). However, the CNN was superior to the CapsNet in 50% of all classes in FS446 and in 36.8% classes in FS534. While this gave hope for a trend reversal in high DSS situations, in the high-shift field set FS466 the CNN is still superior in 55% of all classes and the CapsNet is

Table 3. Changes in model performance for biological classes induced by class-wise P95-filtering. The numbers give the actual result after filtering, while the numbers in the brackets give the difference from pre- to post-filtering. Mean F-scores (F1, precision, and recall) were calculated using only classes with TP + FP > t (group of ‘pure’ classes after filtering). True and empty refer to the respective number of classes predicted by the model for the respective FS. Discard is the percentage of images predicted with insufficient certainty for the P95-filter set.

		Classes		Pure classes						
Mod	**Data**	**True**	**Empty**	**True**	**Empty**	***t***	**Mean F1**	**Mean precision**	**Mean recall**	**Discard (%)**
CNN	446	14 (+0)	3 (−1)	8 (−3)	0 (−1)	15	0.89 (+0.12)	0.94 (+0.17)	0.87 (+0.07)	3
CNN	466	16 (+0)	4 (+0)	1 (−2)	0 (+0)	190	0.74 (+0.09)	0.87 (+0.29)	0.64 (−0.16)	33
CNN	534	15 (−1)	5 (+0)	5 (+0)	0 (−2)	85	0.68 (+0.05)	0.98 (+0.35)	0.61 (−0.03)	45
CAP	446	13 (−1)	1 (−1)	8 (−5)	0 (+0)	5	0.59 (−0.13)	0.94 (+0.06)	0.53 (−0.15)	5
CAP	466	16 (+0)	3 (−1)	3 (−1)	0 (−1)	60	0.55 (+0.05)	0.76 (+0.19)	0.5 (−0.02)	41
CAP	534	15 (−1)	3 (−2)	6 (−4)	0 (−2)	25	0.61 (+0.03)	0.91 (+0.28)	0.59 (−0.01)	33

Table 4. The development of the mean recall and the mean percentage of images to validate with increasing k for FS446, where k was the number of most likely predictions accepted for each image. If the true class belonged to the k accepted predictions, an image was counted as ‘true positive’. N images: 55,302.

***N k* = 2 (%)**	**Recall *k* = 2**	***N k* = 3 (%)**	**Recall *k* = 3**	***N k* = 4 (%)**	**Recall *k* = 4**	***N k* = 5 (%)**	**Recall *k* = 5**
7.8	93.4	11.1 (+3.3)	96.3 (+2.9)	14.5 (+3.5)	97.2 (+0.9)	18.6 (+4.1)	97.6 (+0.4)

only superior in 10% of the classes. Spatial distributions predicted by both models did not show any significant deviations from those of manually validated images, when two conditions were met: $n_{\text{true}} > 50$ images and recall > 20%, regardless of the level of DSS (Supporting Information Table S3). Figure 11 shows two exemplary distributions of copepods predicted by our models in the field sets FS446 and FS534, demonstrating the difficulties the CapsNet had with low abundant classes (Supporting Information Table S3).

Discussion

The intention of this study was to provide a guideline to efficiently process not only common, but also rare biotic taxa using automated analyses of in situ plankton images, which is even more challenging than laboratory imagery of plankton according to Faillettaz et al. (2016).

During the first step of the model training, we observed a great variability of the final test accuracies, even though the procedure was exactly the same each time, except for the sequence of the images. All classes were trained with the same number of images during step 1, but some classes had persistently lower mean test accuracies than others. However, the lower the mean test accuracy, the greater were the CIs. For those classes, for example, “polychaeta,” 100 images were clearly not sufficient to reflect the class variability of the full data set. Some of the 100 models probably learned more relevant patterns, most likely by chance (González et al. 2017). This highlights one of the drawbacks of gradient-based algorithms as described by Subbey (2018) and the importance of a vast amount of training images, especially if different classes contain similar organisms (like “amphipods” and “copepods”) and additionally one or more similar classes have high intraclass variability (e.g., already due to frontal, dorsal, or lateral viewpoints). For example, images of “veliger” were frequently misinterpreted as “pilidium” and in case of the CapsNet even vice versa (Supporting Information Table S4).

For ecological studies, it is sometimes more important that an image is correctly classified into a certain group rather than the exact class (González et al. 2017). For example, one way to cope with the high intraspecific variability in plankton classes is to divide images of a single species into multiple classes according to morphological distinctions (Luo et al. 2018). In this study, initial experiments with our classifier showed that the separation of the images with appendicularians in “appendicularia” and “appendicularia with house” yielded much better results compared to the classification of a single, combined class. Since both classes were treated as one in the subsequent analyses, a misclassification of an “appendicularia” into “appendicularia with house” was ultimately a correct classification, thus increasing the performance of our model in a way of a Top-2 accuracy. This method is probably even more relevant for more detailed image sets that allow for a higher taxonomic resolution than our images (e.g., flowcam images), but is somewhat limited by the number of available training images. Within this context, different sizes of plankton can

Table 5. Results of the Top-k-method with the CNN and k = 3 applied to FS446. The variable ‘n’ is the number of images that need to be manually validated to maximize the recall. The variable ‘N’ is the number of images in FS446 (55,302) including the three classes ‘unknown’, ‘blurry’, and ‘marine snow’. When recall is empty, the model sorted images in a class which was not occupied in the field data set (‘correct classified’ = 0). The ratio ‘correct classified’:‘false classified’ (CC:FC) provides the number of false images to be manually sorted for one true image found.

Class	Recall	CC	*n*	*n*/*N* (%)	CC : FC
amp		0	4824	8.7	
app	0.997	298	6331	11.4	1:20.2
bip	0.987	79	171	0.3	1:1.2
cop	0.98	151	3601	6.5	1:22.8
dia	1	3190	11,172	20.2	1:2.5
ech		0	321	0.6	
egg	1	5	346	0.6	1:68.2
lar	1	15	85	0.2	1:4.7
mal	0.773	22	122	0.2	1:4.5
med	0.789	76	4353	7.9	1:56.3
mne	0.986	144	184	0.3	1:0.3
noc	0.971	348	5197	9.4	1:13.9
plu	0.998	9861	11,675	21.1	1:0.2
pol	0.975	363	10,896	19.7	1:29
pte		0	81	0.1	
rod	0.954	2034	5684	10.3	1:1.8
vel		0	90	0.2	
wor	0.991	705	2731	4.9	1:2.9
zoe	1	27	77	0.1	1:1.9
Mean	0.96±0.07	1155	3576	6.46	1:15
Median	0.987	79	2731	4.9	1:13.9

have different biological meaning, like ontogenetic stages, and thus could be worth including. However, the true size information is unfortunately not available from VPR-images as the distance to the lens is unknown.

Filtering

Our approach reduced the discard of uncertain predicted images, which are removed by filtering, from 35.7% in Luo et al. (2018) to 5%. In Luo et al. (2018), classes with $n < 25$ of 75,000 randomly drawn images were excluded to achieve a mean precision of 90.7%. This threshold ($n = 25$) divided classes into a “pure” (precision > 90%) and an “uncertain”’”(precision < 90%) group. However, an evaluation using n is probably misleading since some classes in the field set had a true $n = 0$, which was below the threshold of $n < 5$, and simultaneously had FP > 5, which was above that threshold. Thus, without human interference (i.e., validation), classes with $n < 5$ could be erroneously categorized as “pure” (and vice versa). Therefore, we used the sum of TPs and FPs instead to divide between “pure” (i.e., trustworthy) and “uncertain” classes (i.e., classes that need to be validated). The major advantage of this approach is that this threshold is applicable without knowledge of the true distribution of the classes since it is based on the predictions instead of the true abundance. Furthermore, we found a correlation between the threshold and the BC dissimilarity that separates the distribution of the TS from the distribution of the new field set. Remarkably, this new method increased the threshold from $n < 5$ to TP + FP < 15 for the CNN while it decreased the threshold for the CapsNet ($n < 25$; TP + FP < 5). This threshold to separate “pure” from “uncertain” classes using the CapsNet was below the threshold of the CNN and therefore the CapsNet is superior in extended production mode applications. Summarizing, the CapsNet had similar discard ratios but lower mean recall scores compared to the CNN. Thus, while it produced more “pure” classes, the drawback was a stronger filter pressure on rare classes.

However, each optical sampler is designed to target a different component of the zooplankton (Owens et al. 2013) which encompasses organisms that vary greatly in terms of size, shape, and behavior (Pitois et al. 2018). This and varying environmental conditions and ecosystem compositions may affect the difficulty of classification tasks (Luo et al. 2018) and might contribute to the differences found between different studies.

The fraction of “pure” classes decreased for the CapsNet from 57% in FS446 (CNN: 57%) with a low amount of DSS to 38% (CNN: 31%) in FS534 with the medium amount of DSS. The high amount of DSS in FS466 overcharged both models similarly, as no class actually reached > 90% precision. This trend observed between the threshold “*t*” and DSS (measured as BC dissimilarity) was not reflected in the *F*-metrics. No obvious pattern for biological classes was observed regarding the *F*-score, recall, or precision, indicating that the community structure, and thus the difficulty of the classification task, is probably an evenly strong driver of model performance as DSS. Considering all classes, the CapsNet could not achieve as high scores as the CNN under a low amount of DSS, but the decrease with increasing DSS was also lower. However, for the CapsNet to overcome the CNN, an amount of DSS is probably necessary that precludes a practical application of either model.

The performance of regionally trained classifiers tends to decline with increasing environmental dissimilarity, whereas a globally trained classifier achieved similar results in all areas, but at the cost of lower accuracies for rare taxa (Chang et al. 2012). Continually increasing the training set and adapting the model to new situations could therefore help coping with unknown community structures and keeping the amount of DSS at a lower level, but this was not investigated in this study.

Spatial distributions

Both models accurately predicted the spatial distributions of filtered classes with $n_{\text{true}} > 50$ and recall > 20%, regardless of the amount of DSS. Due to better recall scores, the CNN could predict the spatial distributions of more classes compared to

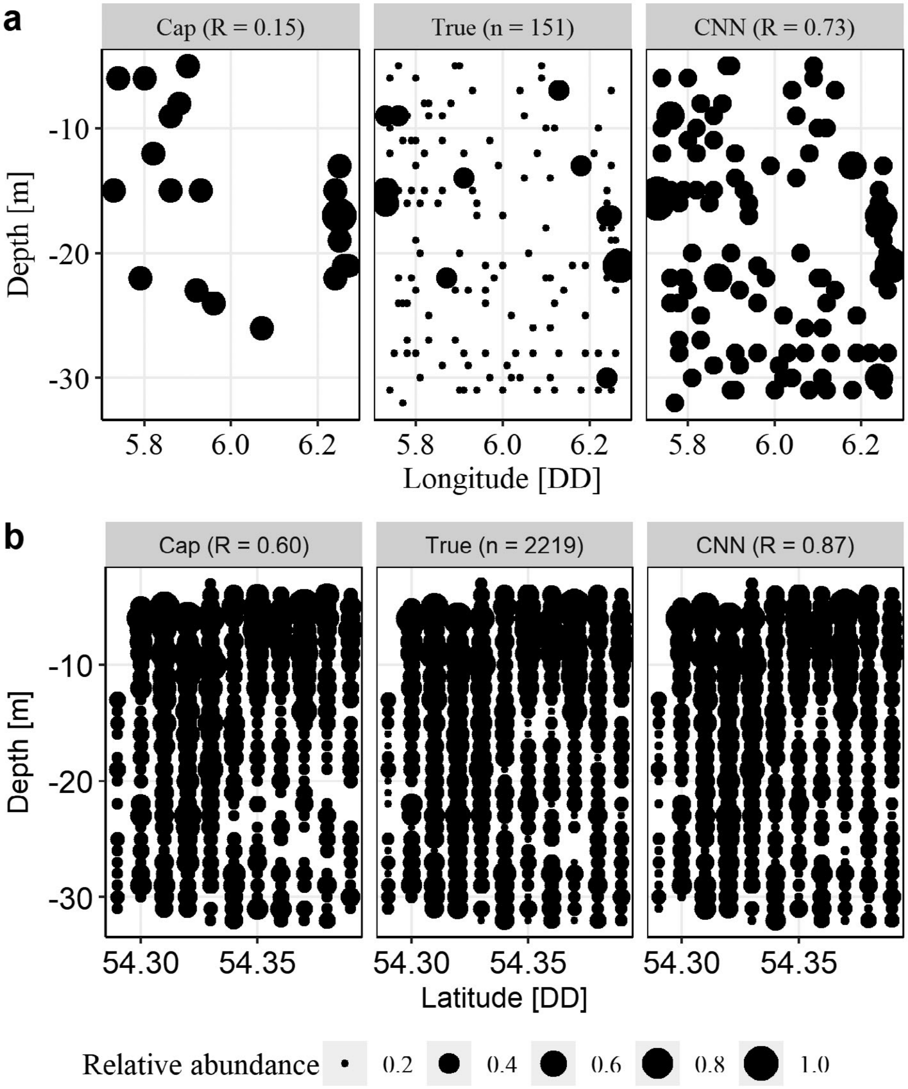
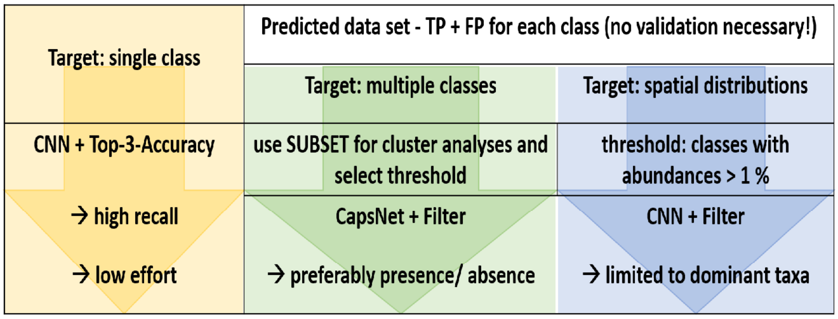


Fig 11. Spatial distribution of the relative copepod abundances in FS446 (**a**) and FS534 (**b**) aggregated by longitude/latitude (0.01 DD) and depth (m). The predictions of the CapsNet (left panel) and the CNN (right panel) were compared to the manual validation (central panel). *n*, absolute abundance; *R*, recall.



Target: single class

Predicted data set - TP + FP for each class (no validation necessary!)

Target: multiple classes

Target: spatial distributions

CNN + Top-3-Accuracy

use SUBSET for cluster analyses and select threshold

threshold: classes with abundances > 1 %

→ high recall

CapsNet + Filter

CNN + Filter

→ low effort

→ preferably presence/ absence

→ limited to dominant taxa

Fig 12. Suggested workflows for three research target specific automated plankton image analysis methods.

the CapsNet, especially more smaller classes. Most classes which spatial distributions were correctly predicted belonged to the "pure" group. However, the opposite was not given. Due to low recall scores not all predictions for classes from the "pure" group reflected the spatial distribution in the field, as some "pure" classes had a high precision but a low recall (e.g., CapsNet FS446 "appendicularia": precision = 100%, recall < 1%). Thus, a categorization in trustworthy and misleading predicted spatial distributions generally requires knowledge of the recall and therefore manual validation.

Top-*k* predictions

The application of filters enables a user to automatically detect a wide range of taxa with a high precision. However, less abundant classes are still difficult to predict, especially if DSS occurs. Since filters are per definition not appropriate to increase the recall and make it more likely to detect rare classes, we instead employed the concept of the Top-5-Accuracy used in machine learning. Here, the correct class of an image does not have to have the highest probability. Instead, the correctly predicted class is among the five highest probabilities. Using already the Top-3-Accuracy, we increased the mean recall by 33.3–96.3% and reduced the average number of images that needed manual validation to 6.5% of the original amount. This method significantly reduces the required human efforts if the research focuses on rare classes, like fish larvae, that are most likely not detected at a sufficient rate using only the highest probability. Such an approach is certainly limited by the size of a potential data set and the number of classes, but so far it is probably the most effective way if spatial distributions are equally important as total abundances.

Comments and recommendations

The effectivity of a classifier is not solely determined by the final model performance. Particularly, the specific objectives of a research task need to be considered to tailor the best model. Research targeting rare classes usually requires a quantitative classification (i.e., high recall) rather than a qualitative classification (i.e., high precision) which is more important for a study of community structures and biodiversity. The assessment of spatial distributions requires qualitative and quantitative classifications which is, without manual validation, currently limited to dominant classes (> 1% of the whole data set). As a general guideline, we propose the following scheme (Fig. 12): either model can be used to classify a given data set using the P95-filter set. Subsequently, a cluster analyses to estimate the similarity of the new data set (predictions) and the training data set (validated) is needed. As we have not investigated how the thresholds behave for increasing data set sizes, currently a subset of the original data set of 30,000–50,000 images is recommended to estimate the threshold for "pure" classes based on Eq. 5. Recall scores of the CapsNet fluctuate less strong with varying levels of DSS and the threshold "*t*" can be adapted dynamically, which is important for the comparison of different samples. The spatial distributions of a particular sample should be investigated using a CNN due to better recall scores in general, but without manual validation this is limited to the dominant taxa. Rare taxa should be targeted using a CNN and the Top-3-Accuracy to maximize the recall at a limited amount of human effort. Thus, the only benefit in using a CapsNet in fact arises under the presumption of DSS.

Data availability statement

All manually classified images from the full training set and test sets (124 K probability filtering set and 94 K random field sets), as well as text files containing predicted and validated classes for all test sets will be available on Zenodo.org (doi: 10.5281/zenodo.4431509).

References

Abadi, M., and others. 2015. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. Available from https://www.tensorflow.org/. (accessed 30 March 2020).

Afshar, P., A. Mohammadi, and K. N. Plataniotis. 2018. Brain tumor type classification via capsule networks. arXiv: 1802.10200 [cs]. Available from http://arxiv.org/abs/1802.10200. (accessed 02 April 2019).

Al-Barazanchi, H., A. Verma, and S. X. Wang. 2016. Intelligent plankton image classification with deep learning. Int. J. of Computational Vision and Robotics **8**: 561–571. doi:10.1504/IJCVR.2018.095584.

Batten, S. D., and others. 2019. A global plankton diversity monitoring program. Front. Mar. Sci. **6** 321. 10.3389/fmars.2019.00321

Bell, J. L., and R. R. Hopcroft. 2008. Assessment of ZooImage as a tool for the classification of zooplankton. J. Plankton Res. **30**: 1351–1367. doi:10.1093/plankt/fbn092

Benfield, M., and others. 2007. RAPID: Research on automated plankton identification. Oceanography **20**: 172–187. doi:10.5670/oceanog.2007.63

Bishop, C. M. 1995. Neural networks for pattern recognition. Oxford Univ. Press.

Bray J. Roger, Curtis J. T. 1957. An ordination of the upland forest communities of southern wisconsin. Ecological Monographs. **27** (4):325–349. doi:10.2307/1942268.

Briseño-Avena, C., M. S. Schmid, K. Swieca, S. Sponaugle, R. D. Brodeur, and R. K. Cowen. 2020. Three-dimensional cross-shelf zooplankton distributions off the Central Oregon Coast during anomalous oceanographic conditions. Prog. Oceanogr. **188**: 102436. doi:10.1016/j.pocean.2020.102436

Chang, C.-Y., P.-C. Ho, A. R. Sastri, Y.-C. Lee, G.-C. Gong, and C.-H. Hsieh. 2012. Methods of training set construction: Towards improving performance for automated mesozooplankton image classification systems. Cont. Shelf Res. **36**: 19–28. doi:10.1016/j.csr.2012.01.005

Chollet, F. 2015. Keras. GitHub. Available from https://github.com/fchollet/keras

Chollet, F. 2017. Deep learning with python, 1st Edition. Manning Publications.

Culverhouse, P., R. Williams, B. Reguera, V. Herry, and S. González-Gil. 2003. Do experts make mistakes? A comparison of human and machine identification of dinoflagellates. Mar. Ecol. Prog. Ser. **247**: 17–25. doi:10.3354/meps247017

Davis, C. S., S. M. Gallager, M. S. Bermann, L. R. Haury, and J. R. Strickler. 1992. The video plankton recorder (VPR): Design and initial results. Arch. Hydrobiol. **36**: 67–81.

Davis, C. S., and D. J. McGillicuddy. 2006. Transatlantic abundance of the n2-fixing colonial cyanobacterium trichodesmium. Science **312**: 1517–1520. doi:10.1126/science.1123570

Dutilleul, P., P. Clifford, S. Richardson, and D. Hemon. 1993. Modifying the t test for assessing the correlation between two spatial processes. Biometrics **49**: 305–314. doi:10.2307/2532625

Faillettaz, R., M. Picheral, J. Y. Luo, C. Guigand, R. K. Cowen, and J.-O. Irisson. 2016. Imperfect automatic image classification successfully describes plankton distribution patterns. Methods Oceanogr. **15–16**: 60–77. doi:10.1016/j.mio.2016.04.003

Floeter, J., and others. 2017. Pelagic effects of offshore wind farm foundations in the stratified north sea. Prog. Oceanogr. **156**: 154–173. doi:10.1016/j.pocean.2017.07.003

González, P., E. Álvarez, J. Díez, Á. López-Urrutia, and J. J. del Coz. 2017. Validation methods for plankton image classification systems. Limnol. Oceanogr.: Methods **15**: 221–237. doi:10.1002/lom3.10151

Hansson, S., U. Larsson, and S. Johansson. 1990. Selective predation by herring and mysids, and zooplankton community structure in a Baltic Sea coastal area. J. Plankton Res. **12**: 1099–1116. doi:10.1093/plankt/12.5.1099

He, H., and E. A. Garcia. 2009. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**: 1263–1284. doi:10.1109/TKDE.2008.239

He, K., X. Zhang, S. Ren, and J. Sun. 2015. Deep residual learning for image recognition. arXiv:1512.03385 [cs]. [accessed 2020 March 26]. Available from http://arxiv.org/abs/1512.03385

Hinton, G. E., A. Krizhevsky, and S. D. Wang. 2011. Transforming auto-encoders, p. 44–51.*In* T. Honkela, W. Duch, M. Girolami, and S. Kaski [eds.], Artificial neural networks and machine learning – ICANN 2011. Springer Berlin.

Hu, Q., and C. Davis. 2005. Automatic plankton image recognition with co-occurrence matrices and support vector machine. Mar. Ecol. Prog. Ser. **295**: 21–31. doi:10.3354/meps295021

Hu, Q., and C. Davis. 2006. Accurate automatic quantification of taxa-specific plankton abundance using dual classification with correction. Mar. Ecol. Prog. Ser. **306**: 51–61. doi: 10.3354/MEPS306051

Jiménez-Sánchez, A., S. Albarqouni, and D. Mateus. 2018. Capsule networks against medical imaging data challenges, p. 150–160. arXiv:1807.07559 [cs], 11043. doi: 10.1007/978-3-030-01364-6_17

Johansson, M., E. Gorokhova, and U. Larsson. 2004. Annual variability in ciliate community structure, potential prey and predators in the open northern Baltic Sea proper. J. Plankton Res. **26**: 67–80. doi:10.1093/plankt/fbg115

Kingma, D. P., and J. Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980 [cs]. [accessed 2019 April 11]. Available from http://arxiv.org/abs/1412.6980

Kornblith, S., J. Shlens, and Q. V. Le. 2019. Do better ImageNet models transfer better?, p. 2656–2666. *In* 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/CVPR.2019.00277.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM **60**: 84–90. doi:10.1145/3065386

LeCun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. Nature **521**: 436–444. doi:10.1038/nature14539

Lee, H., M. Park, and J. Kim. 2016. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning, p. 3713–3717. *In* 2016 IEEE International Conference on Image Processing (ICIP). doi: 10.1109/ICIP.2016.7533053

Luo, J. Y., J.-O. Irisson, B. Graham, C. Guigand, A. Sarafraz, C. Mader, and R. K. Cowen. 2018. Automated plankton image analysis using convolutional neural networks. Limnol. Oceanogr.: Methods **16**: 814–827. doi:10.1002/lom3.10285

Möller, K. O., M. St John, A. Temming, J. Floeter, A. Sell, J. Herrmann, and C. Möllmann. 2012. Marine snow, zooplankton and thin layers: Indications of a trophic link from small-scale sampling with the video plankton recorder. Mar. Ecol. Prog. Ser. **468**: 57–69. doi:10.3354/meps09984

Möller, K. O., and others. 2015. Effects of climate-induced habitat changes on a key zooplankton species. J. Plankton Res. **37**: 530–541. doi:10.1093/plankt/fbv033

Moreno-Torres, J. G., T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera. 2012. A unifying view on dataset shift in classification. Pattern Recogn. **45**: 521–530. doi:10.1016/j.patcog.2011.06.019

Nanni, L., S. Brahnam, S. Ghidoni, and A. Lumini. 2019. Bioimage classification with handcrafted and learned features. IEEE Computer Society Press; [accessed 2020 March 26]. Available from https://doi.org/10.1109/TCBB.2018.2821127

Ohman, M. D., R. E. Davis, J. T. Sherman, K. R. Grindley, B. M. Whitmore, C. F. Nickels, and J. S. Ellen. 2019. Zooglider: An autonomous vehicle for optical and acoustic sensing of zooplankton. Limnol. Oceanogr.: Methods **17**: 69–86. doi:10.1002/lom3.10301

Oksanen, J., and others. 2019. Vegan: Community ecology package. Available from https://CRAN.R-project.org/package=vegan. (accessed 09 September 2020).

Orenstein, E. C., O. Beijbom, E. E. Peacock, and H. M. Sosik. 2015. WHOI-plankton—a large scale fine grained visual recognition benchmark dataset for plankton classification. arXiv:1510.00745 [cs]. Available from http://arxiv.org/abs/1510.00745. (accessed 25 April 2019).

Osorio, F., and R. Vallejos. 2019. Tools for assessment the association between two spatial processes. Available from http://spatialpack.mat.utfsm.cl. (accessed 06 April 2020).

Owens, N. J. P., G. W. Hosie, S. D. Batten, M. Edwards, D. G. Johns, and G. Beaugrand. 2013. All plankton sampling systems underestimate abundance: Response to "continuous plankton recorder underestimates zooplankton abundance" by J.W. Dippner and M. Krause. J. Mar. Syst. **128**: 240–242. doi:10.1016/j.jmarsys.2013.05.003

Pan, S. J., and Q. Yang. 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**: 1345–1359. doi:10.1109/TKDE.2009.191

Peters, J., J. Dutz, and W. Hagen. 2013. Trophodynamics and life-cycle strategies of the copepods temora longicornis and *Acartia longiremis* in the Central Baltic Sea. J. Plankton Res. **35**: 595–609. doi:10.1093/plankt/fbt004

Pitois, S. G., J. Tilbury, P. Bouch, H. Close, S. Barnett, and P. F. Culverhouse. 2018. Comparison of a cost-effective integrated plankton sampling and imaging instrument with traditional systems for mesozooplankton sampling in the Celtic Sea. Front. Mar. Sci. **5**: 5. doi:10.3389/fmars.2018.00005

R Core Team. 2020. R: A language and environment for statistical computing. R Foundation for Statistical Computing. Available from https://www.R-project.org/. (accessed 30 March 2020).

Rajasegaran, J., V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo. 2019. DeepCaps: Going deeper with capsule networks. arXiv:1904.09546 [cs]. Available from http://arxiv.org/abs/1904.09546. (accessed 29 January 2020).

Renz, J., and H.-J. Hirche. 2006. Life cycle of *Pseudocalanus acuspes* giesbrecht (copepoda, calanoida) in the Central Baltic Sea: I. Seasonal and spatial distribution. Mar. Biol. **148**: 567–580. doi:10.1007/s00227-005-0103-5

Russakovsky, O., and others. 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**: 211–252. doi:10.1007/s11263-015-0816-y

Sabour, S., N. Frosst, and G. E. Hinton. 2017. Dynamic routing between capsules. arXiv:1710.09829 [cs]. Available from http://arxiv.org/abs/1710.09829. (accessed 01 April 2019).

Schmid, M. S., R. K. Cowen, K. Robinson, J. Y. Luo, C. Briseño-Avena, and S. Sponaugle. 2020. Prey and predator overlap at the edge of a mesoscale eddy: Fine-scale, in-situ distributions to inform our understanding of oceanographic processes. Sci. Rep. **10**: 921. doi:10.1038/s41598-020-57879-x

Sosik, H. M., and R. J. Olson. 2007. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry: Phytoplankton image classification. Limnol. Oceanogr.: Methods **5**: 204–216. doi:10.4319/lom.2007.5.204

Ston, J., A. Kosakowska, M. Lotocka, and E. Lysiak-Pastuszak. 2002. Pigment composition in relation to phytoplankton community structure and nutrient content in the Baltic Sea. Oceanologia **44**: 419–437.

Subbey, S. 2018. Parameter estimation in stock assessment modelling: Caveats with gradient-based algorithms. ICES J. Mar. Sci. **75**: 1511–1511. doi:10.1093/icesjms/fsy060

Tang, X., and W. K. Stewart. 1996. Plankton image classification using novel parallel-training learning vector quantization network, p. 1227–1236. *In* OCEANS 96 MTS/IEEE Conference Proceedings. The Coastal Ocean - Prospects for the 21st Century. V. 3. doi:10.1109/OCEANS.1996.569077

Toraman, S., T. B. Alakus, and I. Turkoglu. 2020. Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from x-ray images using capsule networks. Chaos Solitons Fractals **140**: 110122. doi:10.1016/j.chaos.2020.110122

Van Rossum, G., and F. L. Drake. 2009. Python 3 reference manual. CreateSpace.

Vuorio, K., A. Lagus, J. M. Lehtimäki, J. Suomela, and H. Helminen. 2005. Phytoplankton community responses to nutrient and iron enrichment under different nitrogen to phosphorus ratios in the northern Baltic Sea. J. Exp. Mar. Biol. Ecol. **322**: 39–52. doi:10.1016/j.jembe.2005.02.006

Webb, G. I., L. K. Lee, B. Goethals, and F. Petitjean. 2018. Analyzing concept drift and shift from sample data. Data Min. Knowl. Discov. **32**: 1179–1199. doi:10.1007/s10618-018-0554-1

Wickham, H. 2016. Ggplot2: Elegant graphics for data analysis. Springer-Verlag. Available from https://ggplot2.tidyverse.org. (accessed 30 March 2020).

Wickham, H., R. François, L. Henry, and K. Müller. 2020. Dplyr: A grammar of data manipulation. Available from https://CRAN.R-project.org/package=dplyr. (accessed 30 March 2020).

Wiebe, P. H., and M. C. Benfield. 2003. From the hensen net toward four-dimensional biological oceanography. Prog. Oceanogr. **56**: 7–136. doi:10.1016/S0079-6611(02)00140-4

Xi, E., S. Bing, and Y. Jin. 2017. Capsule network performance on complex data. arXiv:1712.03480 [cs, stat]. Available from http://arxiv.org/abs/1712.03480. (accessed 08 April 2019).

Acknowledgments

Many thanks to Ángel López-Urrutia for his insightful comments and helpful discussions. We thank Mathis Rosenhauer for the introduction in how to operate the high performance computing system “Mistral” of the German Climate Computing Center and Inga Hense for sharing valuable computing hours of her project. The three RV Heincke research cruises were supported by Grant AWI_HE446_00, AWI_HE466_00, and AWI_HE534_00. Finally, we would like to thank an anonymous reviewer whose comments helped to improve our manuscript. Open Access funding enabled and organized by ProjektDEAL.

Conflict of Interest

None declared.

Submitted 11 September 2020
Revised 11 December 2020
Accepted 24 December 2020

Associate editor: Malinda Sutor