MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
1 November 2022

REVISED
9 June 2023

ACCEPTED FOR PUBLICATION
4 July 2023

PUBLISHED
14 July 2023

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Robust detection of marine life with label-free image feature learning and probability calibration

Tobias Schanz[1,3,*], **Klas Ove Möller**[2], **Saskia Rühl**[2] and **David S Greenberg**[1,*]

[1] Helmholtz-Center Hereon, Institute of Coastal Systems—Analysis and Modeling, Model-Driven Machine Learning, Geesthacht, Germany
[2] Helmholtz-Center Hereon, Institute of Carbon Cycles, Biological Carbon Pump, Geesthacht, Germany
[3] International Max Planck Research School on Earth System Modeling, Hamburg, Germany
[*] Authors to whom any correspondence should be addressed.

E-mail: tobias.machnitzki@hereon.de and david.greenberg@hereon.de

Keywords: classification, semi-supervised, plankton, neural networks, SimCLR, Bayesian correction, multi-label classification

Abstract

Advances in *in situ* marine life imaging have significantly increased the size and quality of available datasets, but automatic image analysis has not kept pace. Machine learning has shown promise for image processing, but its effectiveness is limited by several open challenges: the requirement for large expert-labeled training datasets, disagreement among experts, under-representation of various species and unreliable or overconfident predictions. To overcome these obstacles for automated underwater imaging, we combine and test recent developments in deep classifier networks and self-supervised feature learning. We use unlabeled images for pretraining deep neural networks to extract task-relevant image features, allowing learning algorithms to cope with scarcity in expert labels, and carefully evaluate performance in subsequent label-based tasks. Performance on rare classes is improved by applying data rebalancing together with a Bayesian correction to avoid biasing inferred *in situ* class frequencies. A divergence-based loss allows training on multiple, conflicting labels for the same image, leading to better estimates of uncertainty which we quantify with a novel accuracy measure. Together, these techniques can reduce the required label counts ∼100-fold while maintaining the accuracy of standard supervised training, shorten training time, cope with expert disagreement and reduce overconfidence.

1. Introduction

Improvements in imaging technology have strongly impacted marine science, with large image datasets becoming an important asset across species, locations and imaging modalities (Möller *et al* 2015, Vilgrain *et al* 2021). The ability to observe individual organisms *in situ*, measure spatio-temporal density variations, and identify co-occurrence of species, particles, and environmental factors has unlocked previously inaccessible research directions.

However, before underwater images can provide useful information for marine science, they must generally be 'labeled' with image classes corresponding to species or particle types. With large data volumes, manual annotation by qualified human experts presents an analysis bottleneck, so that ongoing improvements in the imaging rate, sensor quality, depth of field or number of cameras no longer increase the rate at which relevant information is obtained. For example, a single underwater camera can easily acquire 10 000 plankton images every hour without interruption, while a human expert spending 5 s per image could label only 720 images in an hour of focused work.

As an alternative to manual annotation, deep neural networks have shown considerable success in image-based classification, segmentation and regression tasks, approaching or exceeding human capabilities (Gu *et al* 2018, Alzubaidi *et al* 2021, Caron *et al* 2021). The most common approach is *supervised learning*, in which a neural network is trained on a fixed set of image-label pairs, with the aim of quickly and accurately labeling unseen images without human intervention. Several recent studies have successfully applied

© 2023 The Author(s). Published by IOP Publishing Ltd

supervised deep learning to underwater image classification (Dai *et al* 2016, Lumini and Nanni 2019), while others have used additional techniques including principal component analysis, support vector machines and random forests (Di Mauro *et al* 2011, Li *et al* 2014, Zheng *et al* 2017).

However, real-world performance of image classifiers lags behind the impressive results on standard benchmark datasets, due to several key differences. Foremost among these is label scarcity: some images in the training data must be 'labeled' by human experts. Benchmarks can have millions (Lin *et al* 2014, Russakovsky *et al* 2015) of images or more (Sun *et al* 2017), but even 100 000 labeled images is high for a marine biology study (Sosik and Olson 2007, Gorsky *et al* 2010) since the required expertise is itself a scarce resource. Another difficulty arises from class imbalance: performance is degraded for species or particle types that occur rarely. While high accuracy can be obtained on 'rebalanced' data with equal representation of all classes (Branco *et al* 2015), this is not representative of real-world performance. Furthermore, imperfect visibility and image focus can lead to expert disagreement, so that training data exhibit noise in their labels (Zhu and Wu 2004).

To address these challenges, we present an approach to image classification with neural networks that incorporates recent advances to go beyond simple supervised deep learning. Our study makes three main contributions:

First, to address class imbalance, we carefully investigate its effects on how classifiers are trained and evaluated. We rebalance our data to improve recognition of rare classes while incorporating a Bayesian correction to avoid bias. We quantify the effects of these techniques on overall accuracy and for individual image classes.

Second, we pretrain our networks using the SimCLR algorithm (Chen *et al* 2020a) to cope with label scarcity. SimCLR is an unsupervised method—it trains the network to image features relevant for classification without requiring labels for its input images, and we show that it effectively reduces the number of labeled training examples required to achieve accurate classification. We compare and evaluate multiple techniques for combining label-free feature learning with subsequent label-driven classification.

Third, to account for the possibility of inaccurate or conflicting labels, we apply a divergence-based loss function to train on images with multiple expert labels. We show that this increases robustness and reduces overconfidence in the presence of label noise. We also introduce a novel accuracy measure for multi-labeled data that assesses both the classifier's accuracy and its quantification of uncertainty.

We demonstrate and evaluate these techniques using two *in situ* datasets from Cape Verde and Helgoland, observing significant benefits to accuracy, data efficiency and computation time.

2. Related work

Several previous studies have addressed label scarcity in aquatic imaging using transfer learning, in which supervised training for one task or dataset is used to initialize network weights for another. (Lumini and Nanni 2019) trained convolutional networks on many labeled datasets, then tuned the networks further on the task of interest. Other studies instead initialized via supervised training on ImageNet data (Kyathanahally *et al* 2021, Le *et al* 2022). Here we will take a different approach using semi-supervised learning. The relative merits of semi-supervised vs. transfer learning depend on the number of unlabeled images available for pretraining, the number of labeled images available for transfer learning, and the relatedness of the different datasets and tasks.

Below, we describe our semi-supervised learning framework based on the SimCLR method (Chen *et al* 2020a). Among alternative semi-supervised approaches, the most successful is Momentum Contrast (MoCo) (He *et al* 2020), which uses the same NT-Xent loss (Sohn 2016) but avoids large batch sizes by building a dictionary queue. Both SimCLR and MOCO have been significantly improved since their introduction, and unsupervised pretraining remains an active area of research (Chen *et al* 2020b, 2020c, Bardes *et al* 2022, Garrido *et al* 2022, Yeh *et al* 2022).

While high accuracy can be obtained on 'rebalanced' test data with equal representation of all classes (Branco *et al* 2015, Kraft *et al* 2022), this is not representative of real-world performance. (Bochinski *et al* 2019) handle an imbalanced dataset by first training on a balanced subdivision of the complete dataset, only sampling as many images for each class as are available for the least present class. They then make predictions for unlabeled data, identify images for which the network is most uncertainty for further expert labeling. A fundamentally different way of dealing with class imbalance is by synthesizing new images, either by data augmentations (Luo *et al* 2018) or with generative modeling (Wang *et al* 2017). In some cases, strong performance has been demonstrated on unbalanced data (Kyathanahally *et al* 2021). To our knowledge, ours is the first study to carefully examine the effects of Bayesian correction on deep learning of plankton classification.

A number of plankton classification studies have used probability filtering, in which automatic classification results are discarded when the maximum inferred probability is below some (possibly

class-specific) threshold (Luo *et al* 2018, Guo *et al* 2021, Kraft *et al* 2022). This technique reduces the number of incorrect images, but leaves some images unclassified, possibly biasing the estimated class frequencies or relationships with environmental drivers. In (Luo *et al* 2018) 35.7% of images were discarded, which for our CPICS imaging data would mean up to 3500 unclassified images per hour. While we chose not to employ probability filtering, its desirability ultimately depends on the scientific questions pursued.

The problem of disagreement among expert labels has so far received limited attention, especially in plankton imaging. (Koller *et al* 2022) examined using the KL-divergence loss for such a problem in a land-use classification task. Unlike their study, we cannot use majority votes to find one true label for accuracy calculations, since we only have three equally valid 'votes' per labeled image. We will therefore introduce a novel way for assessing accuracy in the presence of multiple target labels.

3. Data

We use two image datasets (figure 2(a)), collected at Cape Verde and at the coast of Helgoland.

Cape Verde data were collected on a 21-day research cruise (19.5–26° W, 14–19° N, 24 November–17 December 2019), using a Continuous Particle Imaging Classification System (CPICS) deployed in 73 vertical hauls attached to the Rosette frame (500–4231 m depth). A set of 15 class labels were used to manually and automatically classify these data (figure 1). 204,787 images were labeled by a single human expert (example images in figure A1), and a further 107 532 unlabeled images were available as well.

Helgoland data were also generated using a CPICS, but instead of sporadic vertical hauls, it was deployed on a stationary lander north of the island of Helgoland (54.11° N, 7.55° E). Data were collected between 0:00 and 5:30 UTC on 28 July 2021, with a lander frame capable of remote-controlled vertical profiling between seafloor and surface. The lander frame ascended passively through the water column (0–10 m depth) using the buoyancy of swimmers attached to three vertical struts, and descended actively using a winch system to an anchored base. Helgoland data were assigned to 8 classes, and the same 1000 images were labeled by three human experts independently (example images in figure A2). From this location, an additional 123 147 unlabeled images were available.

While the Cape Verde data can be characterized as open water observations, the conditions near Helgoland are distinctly coastal.

Since the images provided by the CPICS system vary in height and width vary from 36×42 pixel over 28×180 and up to 2576×2156 pixel, we resized them to a fixed height and width using bilinear interpolation. We also experimented with alternative approaches, such as first appending zero-valued pixels on the images' shorter side before resizing them afterward to maintain the original aspect ratio, but found these to yield poorer performance. This coincides with a study from (Lumini *et al* 2020) and thus reinforces our decision.

4. Methods

Our overall aim is classifying each *in situ* image in our datasets as belonging to one of several dataset-specific image classes. These classes correspond roughly to biological taxa, but also include marine snow and blurry images. Thus, for every image x, we wish to assign a class label y.

In this study, we report results from both *supervised learning*, in which training data consists of both images x and labels y, as well *unsupervised learning*, in which training data consists of images alone. We also carry out *semi-supervised learning*, in which network parameters are first initialized by unsupervised learning and then further optimized by supervised learning, allowing us to use all labels and images in our datasets. As the main innovations we introduce here involve going beyond simple supervised learning to investigate potential benefits of class rebalancing, semi-supervised learning and a novel multi-label loss, we present supervised learning results primarily as a baseline for further comparison.

We first review and formalize supervised, unsupervised and semi-supervised learning and strategies. Next, we describe class balancing and Bayesian correction. We then explain the difficulty in training on multi-labeled data and why the standard accuracy metric is not sufficient in the multi-label scenario.

4.1. Neural architecture

For all experiments, we use the ResNet50 network, an instance of a residual convolutional neural architecture (He *et al* 2015). Each layer of a residual network updates an additive update to an internal state, making it easier to train deeper models (e.g. with > 10 layers). The ResNet50's 'backbone' consists of 50 layers containing convolutional filters and pointwise rectifying nonlinearities, with three input channels (red, green, and blue color channels of the input images) and a 2048-dimensional feature vector as output. A complete ResNet50 consists of this backbone along with an additional 'head' that takes a feature vector as input and

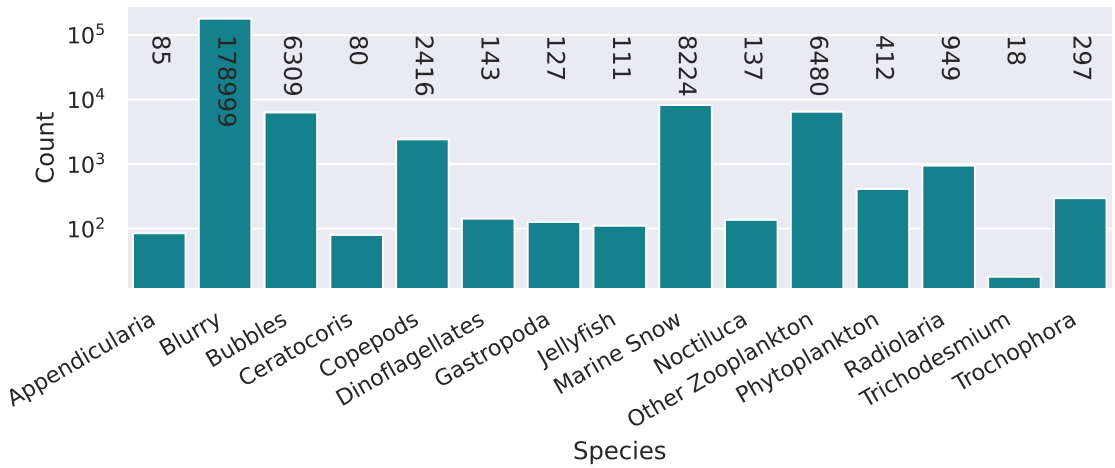

Figure 1. Number of labeled images for each class in the Cape Verde training dataset.

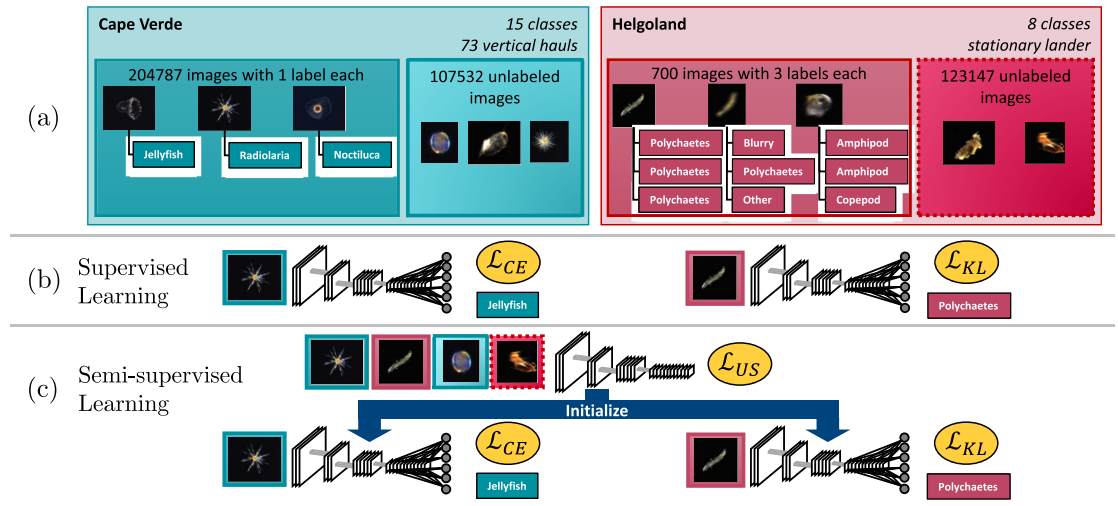

Figure 2. (a) We use two datasets collected during different campaigns, both of which contain more unlabeled than labeled images. (b) In supervised learning only labeled images are used, and a different ResNet50 network is trained for each task using a cross-entropy loss and randomly initialized parameters. (c) In semi-supervised learning, an unsupervised SimCLR-based step is first used to train the 'backbone' of the ResNet50 using both labeled and unlabeled data from both datasets. This pretraining step (upper) does not require or make use of any labels. In a subsequent tuning step (lower), the parameters obtained from the pretraining step are used to initialize supervised training for each dataset and task as previously, using cross-entropy loss and random initialization of classifier 'head' parameters.

produces task-specific outputs such as probabilities for each image class. Full details on the ResNet50 can be found in (He *et al* 2015). We chose ResNet50 since it is a 'modern' architectural standard (Jiao *et al* 2019) that provides close to state-of-the-art classification performance, while remaining conceptually simple and computationally efficient (Shafiq and Gu 2022). Furthermore, ResNet50 is implemented in all major machine learning software frameworks (Chollet *et al* 2015, TorchVision-maintainers and contributors 2016).

4.2. Supervised learning

For supervised learning (figure 2(b)), we used a ResNet50 backbone with a classification head consisting of a fully connected layer and soft-max function. We trained all network parameters by minimizing a standard cross-entropy loss. For a batch of N labeled images from the training data, this loss is defined by

$$\mathcal{L}_{\mathrm{CE}}(\phi) = -\sum_{n=1}^{N} y_n \cdot \log(\hat{y}_\phi(x_n)). \tag{1}$$

For the nth image-label pair, the label y_n is a 'one-hot-encoding' vector that is 1 at the position corresponding to the expert-assigned class, and 0 elsewhere. $\hat{y}_\phi(x_n)$ is the vector of class probabilities by the neural network with the labeled image x_n as input. The logarithm is applied to each output probability individually and ϕ

Table 1. Data augmentations used at different stages of the training. The values for the color jittering mean that for each jitter operation a factor was uniformly chosen from $[max(0, 1 - factor), 1 + factor]$. Only the jittering for the hue factor was uniformly chosen from $[-factor, factor]$.

pretraining	Fine-tuning and supervised baseline
1. Random sized cropping 2. Bilinear resizing to 128 × 128 pixel 3. Random horizontal flips (50 % chance) 4. Random color jittering (80 % chance of being applied. If applied the following factors were used: brightness = 0.8, contrast = 0.8, saturation = 0.8, hue = 0.2) 5. Random gray scaling (20 % chance) 6. Rescaling values of all three color channels to [0, 1]	1. Random sized cropping, 2. Bilinear resizing to 128 × 128 pixel 3. Random horizontal flips (50 % chance) 4. Random vertical flips (50 % chance) 5. Gaussian blurring (kernel size of 5) 6. Random auto-contrast adjustment (50 % chance) 7. Rescaling values of all three color channels to [0, 1]

denotes all trainable network parameters. Minimizing this loss means maximizing expected log probability for the expert-assigned image class. We minimized $\mathcal{L}_{\mathrm{CE}}$ using the stochastic optimization algorithm ADAM (Kingma and Ba 2014) with an initial learning rate of 0.01 and 64 samples per batch.

Both datasets were divided into 70% training, 10% validation and 20% test data. Cape Verde data were split class-wise so that these ratios applied to each image class as well (details in 4.5), while for the multi-labeled Helgoland data we verified that at least one instance of each class label was present in the test data. Only training data were used to minimize the loss over ϕ, but the loss was also measured on validation data during training to assess whether the learned predictions generalize to new data or were overfitting on the training data by 'memorizing' labels for each image. We also used validation data to tune hyperparameters like the learning rate, momentum, batch-size, and augmentations. Test data were used only for calculating the final results reported for each experiment. Supervised learning in this work is used as a baseline for comparison to all further experiments.

4.3. Unsupervised learning

Unsupervised learning uses unlabeled images to train some or all parameters of a neural network. Here, we follow the common two-step pattern in which unsupervised *pretraining* initializes network parameters before further supervised *tuning* (figure 2(c)). Unsupervised learning optimizes a loss function $\mathcal{L}_{\mathrm{US}}$ that depends on network parameters ϕ and images x, but not on labels y, and measures the network's ability to extract 'useful' image features without involving labels. Pretraining replaces the simpler initialization used in purely supervised learning, which is setting the network parameters to values drawn from a random distribution.

We pretrained networks using the SimCLR algorithm (Chen *et al* 2020a), which we adapted for plankton classification. SimCLR's premise is that network outputs should change for each input image, but remain fixed under image transformations such as cropping, rescaling, flipping or color manipulations. Thus, the network should learn to extract features expressing what is 'essential' to each image, in the sense that they distinguish it from other images but are not sensitive to the chosen transformations. Critical details include the choice of image transformations, referred to as 'data augmentations,' and the loss function used to express the concept of same vs. different outputs.

SimCLR duplicates each image x in a training batch, and both copies undergo random augmentations as described in table 1. Next, both transformed copies of each image are passed through the modified ResNet50 to produce feature vectors, so that x_k is used to produce z_{2k-1} and z_{2k}.

Once a batch of N images has been processed to produce $2N$ feature vectors, a cosine similarity score is computed between all feature vector pairs:

$$\mathrm{sim}(z_i, z_j) = \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|}. \tag{2}$$

These scores are then used to calculate the SimCLR loss function:

$$\mathcal{L}_{\mathrm{US}} = \frac{1}{2N} \sum_{k=1}^{N} (\ell_{2k-1,2k} + \ell_{2k,2k-1}) \tag{3}$$

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(z_i, z_k)/\tau)} \tag{4}$$

where the 'temperature' τ is a fixed hyperparameter, which was set to 0.5 throughout all experiments, following (Chen *et al* 2020a). $\mathcal{L}_{\mathrm{US}}$ is minimized when the feature vectors extracted from two transformed

Table 2. Naming and configurations for 3 semi-supervised experiments as well as fully supervised baseline. Hidden classifier layers had 1000 neurons each and ReLU activations.

Method Name	Pretrained Using SimCLR	Optimizer for Tuning	Optimized Backbone Weights	Hidden Layers
Linear Evaluation	Yes	SGD	No	0
Fine-tune SGD	Yes	SGD	Yes	0
Fine-tune Frozen	Yes	Adam	No	1
Supervised	No	Adam	Yes	0

copies of the same image are more similar to each other (numerator in equation (4)) than to other images in the batch (denominator in equation (4)).

In contrast to the batch size of 64 used for supervised learning, SimCLR requires much large batch sizes to achieve sufficient diversity of the pair-wise comparisons of feature vectors. We therefore used a batch size of $N = 8000$ images, and the LARS optimizer (You *et al* 2017) intended for large N. As in (Chen *et al* 2020a), the learning rate was increased during the first 10 epochs from 0.0 to 1 and then decreased with a cosine decay back to 0.0 to a maximum of 800 training epochs. Unsupervised pretraining used all unlabeled images from both datasets, and all labeled images from their training fractions (figure 2(c)). Pretraining used 10 compute nodes with 4 A100 GPUs each over 16.7 h.

4.4. Supervised tuning after unsupervised pretraining

After pretraining, we used labeled images to further train network weights on the classification task (figure 2(c), bottom). The SimCLR head was replaced by a fully connected classifier network that took the 2048-dimensional output vector of the ResNet50 backbone as input. For this second training phase, we evaluated three different modes (table 2). In 'linear evaluation' mode, the classifier consists of a single affine transform followed by a softmax layer, and the ResNet50 backbone parameters remain fixed. 'Fine-tuning frozen' mode uses the same configuration, except that the classifier consists of two fully connected layers with 1000 units in the hidden layer and a ReLU activation after the first. 'Fine-tune SGD' mode uses the same single-layer classifier as linear evaluation mode, but trains all parameters, including those of the ResNet50 backbone. 'Supervised' mode is the baseline from which we measured improvements derived from pretraining: a ResNet50 without the SimCLR step. Further details of SimCLR can be found in (Chen *et al* 2020a, 2020b).

Together, the unsupervised and supervised learning steps comprise a *semi-supervised learning* approach. The three supervision modes present a trade-off between complexity and speed, with 'fine-tune SGD' the most powerful but slowest to train, 'linear evaluation' the fastest but least powerful, and 'fine-tune frozen' a compromise between the two.

4.5. Sub-sampling of training data

We explored how training on a fraction v of labeled training images affects performance of supervised and semi-supervised learning. For Cape Verde data, we sub-sampled each image class: if the kth class has N_k labeled images, we randomly select $\lceil vN_k \rceil$ of these (where $\lceil \cdot \rceil$ denotes rounding up to the next integer). Since Helgoland images may have conflicting class labels, for this data we subsample $\lceil v * N \rceil$ from the full set of labeled images.

4.6. Class rebalancing and Bayesian correction

The imbalanced distribution of marine organisms results in very different label counts for various classes (figure 1). For our datasets, most images belong to the blurry or bubbles classes and are not useful for further analyses. These 'nuisance' classes can make up >90% of images, depending on the instrument used to collect the images and environmental factors such as turbidity. In many studies, these images are manually removed before the supervised learning, and the classifier is trained and evaluated only on relevant classes (Möller *et al* 2012, Faillettaz *et al* 2016). This approach simplifies imbalance problems considerably, but cannot measure accuracy for real-world applications. Furthermore, these systems cannot be deployed to classify incoming streams of 'live' data, as they are incapable of dealing with the sorted out classes.

4.6.1. Class rebalancing

Training a classifier directly on class-imbalanced data emphasizes performance on over-represented classes, while performance on rare classes is degraded. To overcome this problem in the single-label (Cape Verde dataset) experiments, we rebalanced our dataset by sampling images from rare classes more frequently, such that each class contributed labeled images to training batches at the same rate. This was our default strategy

both for fully supervised learning and for tuning after SimCLR-based pretraining. Balancing was not performed on validation data or testing data used to quantify accuracy. We did not rebalance when images were multiply labeled (Helgoland dataset) since 'true' class counts and class memberships were ill-defined.

Since our rebalancing strategy involves training on the same images more frequently for rare classes, it could cause overfitting for classes with few available images. To prevent this, we randomly transformed input images (table 1, right column), similar to the augmentations used for pretraining.

4.6.2. Bayesian correction

Rebalancing data introduces statistical bias into the class probabilities inferred by the network. Suppose a neural network trained on rebalanced data infers that an image is equally likely to have come from two different classes, but one class is 100 times more frequent in the original data. Concluding that the two classes are equally likely would then be incorrect, as the true class frequencies are not considered. To correct this, we use Bayes' theorem:

$$p(y|x) = p(x|y)\frac{p(y)}{p(x)} = p(x|y)\frac{p_r(y)}{p(x)}\frac{p(y)}{p_r(y)} = Cp_r(y|x)p(y) \tag{5}$$

where $p(y)$ are Bayes-corrected class frequencies, $p(y|x)$ are probabilities for each class given the image and the true class frequencies and $p_r(y|x)$ are probabilities inferred by the network given the image and the assumption of equal class frequencies (that is, $p_r(y) = 1/C$).

4.7. Training on multi-labeled data

Since Helgoland data were labeled by multiple experts, cross-entropy loss (equation (1)) is not directly applicable. Instead, we minimize Kullback-Leibler divergence (D_{KL}) between the network's inferred distribution over classes and the empirical distribution of expert labels (Cover and Thomas 1991).

$$D_{\mathrm{KL}}(P||Q) = \sum_k P(k)\log\left(\frac{P(k)}{Q(k)}\right) \tag{6}$$

where $P(k)$ and $Q(k)$ are the empirical and inferred probabilities for the class k.

4.8. Accuracy measures

In all experiments, we measured accuracy, defined as the fraction of correctly classified images. Specifically, the *micro accuracy* (also known as *overall accuracy*) is calculated over all labeled images together:

$$\mathrm{Acc}_{\mathrm{micro}} = \frac{1}{N}\sum_i^N \mathbb{1}_{\hat{y}_i = y_i} \tag{7}$$

where $\hat{y}$ is the prediction and y the target label.

We also measured the *macro accuracy* (or *detection rate*), which is an average class-specific accuracy over C classes:

$$\mathrm{Acc}_{\mathrm{macro}} = \frac{1}{C}\sum_{j=1}^{C}\frac{1}{N_j}\sum_i^N \mathbb{1}_{\hat{y}_{i,j} = y_{i,j}} \tag{8}$$

where N_j images have been expert-labeled as class j, and $\hat{y}_{i,j}$ denotes the ith estimated class for images labeled with class j. The quantity $\frac{1}{N_j}\sum_i^N \mathbb{1}_{\hat{y}_{i,j}=y_{i,j}}$ is the *recall rate* for class j.

Macro and micro accuracy are the same for class-balanced data, but different otherwise. Macro accuracy assigns the same importance to each class, and is sensitive to misclassification of rare classes. Micro accuracy instead assigns the same importance to each image.

For evaluation on multi-labaled data, we defined a custom accuracy measure. We first defined the network's 'prediction' as the class j_{pred} with the highest inferred probability. We then defined accuracy as $P(j_{\mathrm{pred}})/\max_j(P(j))$ where $P(j)$ is the empirical distribution of expert-assigned labels. Thus, accuracy is defined as the number of experts who agreed with the network, normalized by the maximum number of experts who agreed on any class. The intuition behind this accuracy measure is that the network is considered 100% accurate when its prediction matches the maximum possible number of experts.

4.9. Implementation details

We implemented all algorithms and training procedures in PyTorch (Paszke *et al* 2019) with the Lightning library for distributed training (Falcon and team 2019), and the Hydra package for reproducible scripting (Yadan 2019). All learning algorithms and network architectures employed 32 bit precision. Catalyst (Kolesnikov 2018) data samplers were used for balancing data during training. Code is available at https://github.com/m-dml/plankton-classifier, and trained network weights are available upon request.

5. Experiments and analyses

5.1. Bayes-corrected rebalancing recovers rare aquatic objects

We first trained classifiers to label 15 species and particle types on *in situ* images from Cape Verde, using fully supervised learning without pretraining (figure 2(a), left). The aim of these experiments was to identify classification challenges arising from class imbalance, to assess potential means of overcoming these challenges, and to establish a clear baseline for assessing further techniques.

The Cape Verde data exhibited class imbalance (figure 1): for example, nuisance classes were ∼90% of training data while Trichodesmium, a nitrogen-fixing cyanobacteria important to ecosystem function (Hewson *et al* 2009), was 0.02%. Since rare classes contribute negligibly to the training loss, classifiers trained on unbalanced data frequently identified common classes, but failed to detect rare classes. This resulted in high overall accuracy (91.7%, figure 3(a) top, red), but low per-class recall (16.2%, figure 3(a) bottom, red). Furthermore, class densities estimated by the trained network were highly biased, and only the nuisance and marine snow classes were assigned the highest probability for any image in the testing data (figure 3(b), red). These class-specific errors hinder effective automatic classification for scientific studies, as neglecting rarely observed species and objects could significantly distort the emerging picture of an ecosystem or its coupling to environmental factors.

To address this issue, we applied class rebalancing during training (He and Garcia 2009), randomly sampling images from underrepresented classes with increased frequency but different random augmentations so that all classes were encountered equally often as the network learned. This resulted in improved detection rates and more accurate densities for most classes (figures 3(a) and (b), blue), although detection rates for overrepresented classes are reduced. Class rebalancing thus allows the network to learn image features relevant to rare classes during training, but inevitably introduces biases due to different class frequencies in the training and testing data. To reduce these biases while maintaining the ability to identify rare classes, we used Bayes' rule to correct the class probabilities output by the network (section 4.6.2). This Bayesian correction improved overall accuracy to 95.0%, while reducing the average accuracy over image classes by only 5.3%, and further improved estimated class frequencies (figures 3(a) and (b), yellow). We therefore used class-balancing during training and Bayesian correction during inference on test data in all further experiments and analyses. We emphasize that all our reported accuracy values are calculated on held-out test data *without* rebalancing, as rebalanced test data tends to yield high accuracy values (Li *et al* 2021) that are not representative of real-world performance.

We also examined all of our models performances using uncorrected output probabilities (figure D1), which confirms for every case that the correction increases accuracy (in some cases >10%) and only slight reductions in most recall rates.

5.2. Label-free feature learning improves accuracy

We next carried out a series of experiments to quantify the benefits of unsupervised pretraining for plankton classification. Despite the simplicity and modest parameter count of our ResNet50 architecture, accuracy exhibited a clear dependence on the training set size over the entire range we examined, with no sign of saturating at the upper end of this range. Fully supervised networks trained on all available labels (∼150k) correctly classified 95.0% of images (figure 4(a), dark blue plus-signs). While high in absolute terms, this accuracy is not particularly impressive given the dataset's imbalanced class frequencies: 90.4% of all images occurred in the 'blurry' and 'air bubbles' classes, so the utility of any classifier performing below that level is questionable.

In repeated fully supervised training runs with varying numbers of labels, we observed a steady decline in performance as the training dataset became smaller (figure 4(a), dark blue plus-signs). When 10% of training data were used, overall accuracy dropped to 93.0%, with a further drop to 89.2% with 1% of training data used. Similarly, the class-averaged recall rate, which averages the probability of correct detection across classes, decreased from 63.3% for all labels to 35.2% for 10% and 17.1% for 1% of labels.

To address the challenge of data scarcity, we tested whether performance on labeled test data could be improved by pretraining network parameters through unsupervised learning on unlabeled data. For pretraining, we used labeled and unlabeled images from our Cape Verde dataset as well as a further 123 947

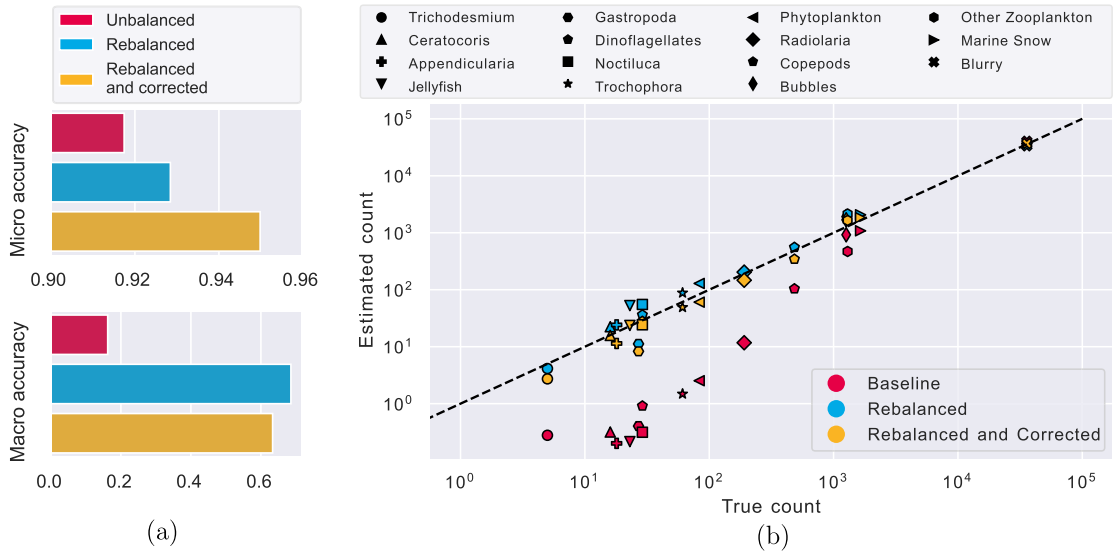

Figure 3. Class rebalancing and Bayesian correction improve performance of fully supervised classifiers. (a) Overall accuracy (top) and per-class detection rate (bottom) on raw data (red), rebalanced data (blue), and rebalanced data with subsequent Bayesian correction (yellow). (b) Estimated counts for each of the classes were calculated by summing predicted probabilities over all input images, resulting in an unbiased estimate when the network successfully retrieves the correct posterior distribution.

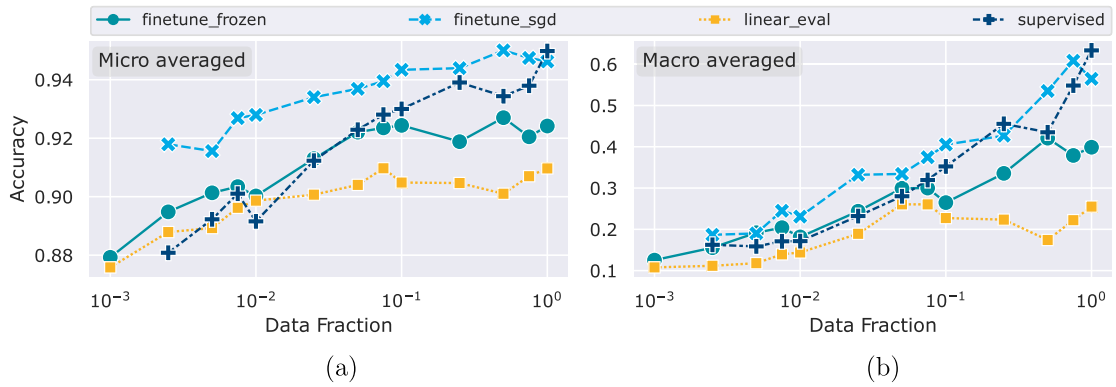

Figure 4. Overall top-1 accuracy on the test dataset at different fractions of labeled training data used. Averaging was once performed over the complete dataset (a) and once per class (b). Finetune SGD outperforms any other method for data sparse regimes.

images collected at Helgoland (374 722 images total, see methods). We applied the unsupervised SimCLR algorithm (Chen *et al* 2020a), which trains a neural network to extract image features that uniquely identify each image while remaining invariant to a selected set of image transformations (see methods). By 'learned image features' we refer to the activations of the final layer of the feature extractor (the ResNet50 backbone without a task-specific head), which can be used in subsequent supervised learning tasks.

For a ResNet50 pretrained on unlabeled data and fine-tuned on all available labeled Cape Verde images, overall accuracy did not increase significantly when compared to supervised learning. However, as the number of labeled images available for training decreased, fine-tuned semi-supervised learning exhibited 2%–5% increases in total accuracy compared to fully supervised learning (figure 4(a), light blue crosses). For example, fine-tuning on only 0.25% of labeled data (358 images) yielded a total accuracy of 92.4%, nearly reaching the 95% achieved by supervised learning on all 150k labeled images (figure 4). In contrast, the supervised learning based on 0.25% data reached an accuracy of only 78.8%. Hence, images were misclassified 2.2 times more often without pretraining. In this particular case, semi-supervised learning could potentially reduce human labor by a factor of 400, without significantly degrading performance. More modest benefits were observed for frozen mode, with performance comparable to supervised learning (albeit at a fraction of the cost, see below), while linear mode yielded poorer performance. Pretraining on unlabeled data also increased the rate at which each image class was correctly identified: the class-averaged recall rate increased by 5%–10% compared to supervised learning when fine-tuning on 0.25%–10% of images.

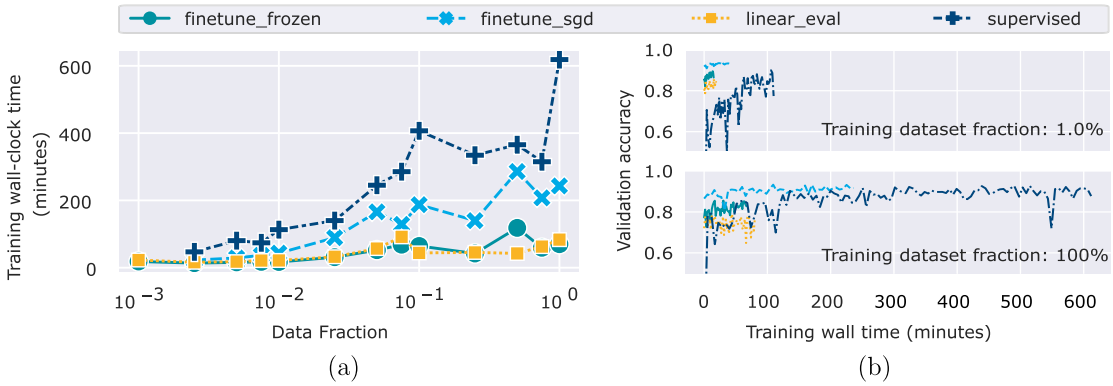

Figure 5. Wall-clock times for training the models. Every model was trained until convergence, making results strongly dependent on the amount of training data and the number of trainable parameters (a). For training on 1% and 100% of the data, the validation accuracy after each epoch is displayed over the training duration (b).

Somewhat paradoxically, we found that while pretraining improved the accuracy with which individual images were classified overall and within each class, statistical mean class densities were more accurate with supervised learning (figure B1). Thus, if the purpose of classification is simply to quantify class densities across the dataset, supervised learning may be preferable, whereas if inferred labels are to be compared to other environmental, temporal or geographic covariates then semi-supervised learning can provide a significant benefit.

5.3. Pretraining accelerates subsequent learning

In addition to improving the final accuracy attained in supervised classification tasks, initialization of neural network weights via SimCLR-based pretraining has been shown to reduce the training time required to reach a given level of accuracy. Together with reductions in the human labor required to generate training data, faster training could provide a significant advantage for many time-critical operations. For example, researchers working in the field would benefit from rapid determination of target species abundances in different locations, to be able to adapt their sampling plans and efforts on the fly.

We therefore examined how pretraining affected the total 'wall-clock time', which is the elapsed time since the start of the tuning, including data loading, augmentations and other overhead on a single A100 GPU required for supervised training of neural networks. Across a wide range of label counts, we observed a two to three-fold reduction of the time until convergence for fine-tuning compared to supervised learning (figure 5(a)). Linear and frozen modes were further accelerated, with training ∼7.4 times faster on the complete dataset, compared to fully supervised training.

We also examined how accuracy progressed during training, to examine what performance can be expected as a function of training duration. Strikingly, after a single 'epoch' with each image used to update parameters only once, fine-tuning with 1% of labels had already outperformed the final, converged performance of all other approaches, including supervised learning (figure 5(b) top, light and dark blue). As a single epoch required only 43 s compared to 113 min for convergence of supervised learning, fine-tuning on a small amount of new data effectively reduces training time by more than two orders of magnitude.

While fine-tuning did not improve converged accuracy on the test set beyond supervised learning when 100% of labels were used, it did perform better on the validation dataset (figure 5(b) bottom, light and dark blue). We found that fine-tuning all network parameters provided higher accuracy than linear or frozen modes for any training run (figure 5(b)). This result was somewhat surprising, as frozen and linear modes have been proposed as efficient alternatives for limited computational budgets, and may provide a useful guideline when applying networks pretrained on plankton imaging to supervised classification problems.

5.4. Learning with conflicting labels

We next turned our attention to a second, more challenging classification task on imaging data collected from a stationary CPICS lander at Helgoland. For the task of assigning each image to one of 8 classes, only 700 labeled images were available for training. However, in contrast to the Cape Verde classification task above, here each image was labeled by the same three experts independently. This allowed us to examine the level of agreement between experts on the same data, and to explore training strategies that could make use of multiple, possibly conflicting labels.

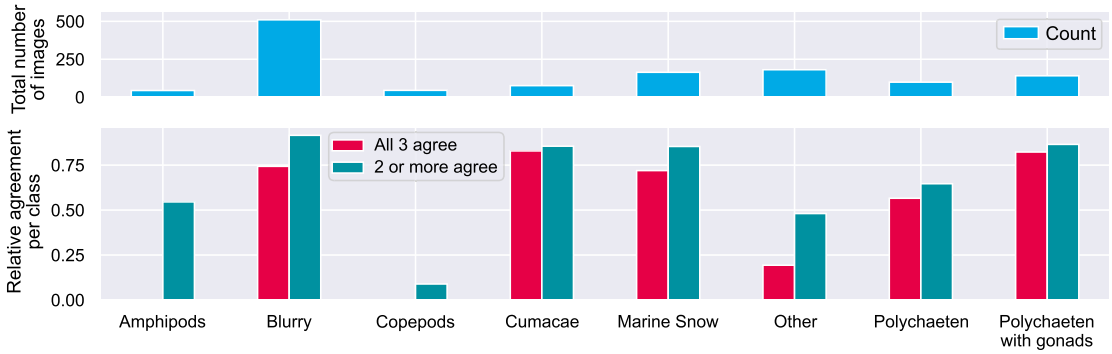

Figure 6. Agreement of the labels in the Helgoland dataset. An image belongs to a class, when at least one label is assigned to that class (top). Therefore, an image can belong to multiple classes and the sum of image counts across classes exceeds 1000.

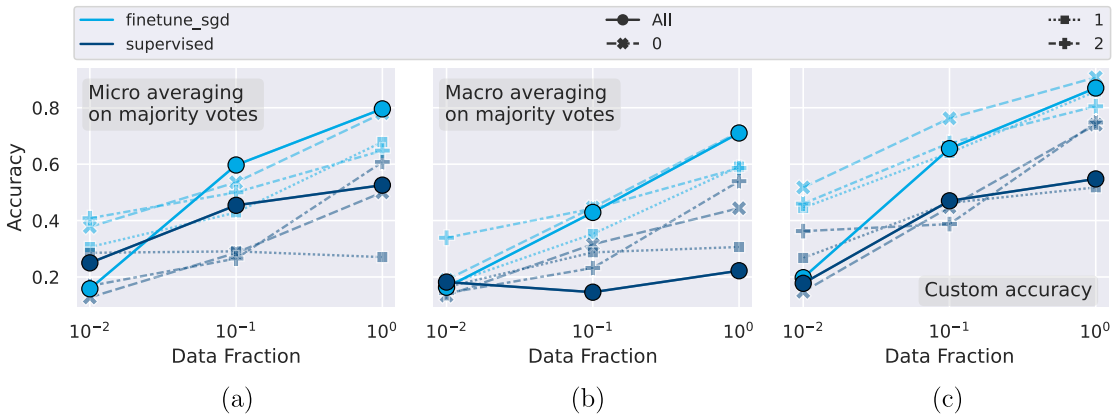

Figure 7. Accuracies averaged over the test dataset when training on multi labeled images. The *x*-axis is referring to the amount of labeled data from the training dataset that was used. Models were trained on each scientist separately using NLL ('0', '1', '2') and are compared to the models trained on the distribution of votes using KL divergence ('All'). Compared are the metrics micro (a) and macro (b) averaging on majority votes, and our custom defined accuracy (c).

We found that all three experts provided the same class label for 76.7% of images, and at least two experts agreed for 97.3% (figure 6). Thus, one fourth of the images exhibited some degree of label conflict, and the overall chance that two randomly selected labels for the same image agree is 80.2%. These results underscore the difficulty of the task and suggest that automatically assigning the 'correct' label with perfect accuracy may not be possible, or even well-defined in every case.

We first examined how fully supervised learning with a standard cross-entropy loss function performed on this task, when trained on a single expert's label. Since the correct label was not known with certainty for all images, we did not apply class rebalancing or Bayesian probability adjustment on this dataset. Using a novel accuracy measure designed for datasets with conflicting labels (see methods), we observed a modest overall accuracy of 51.8%–74.0% on this challenging task, which showed a further decrease when fewer than all 700 labels were used (figure 7(c), dark blue dashes). The class-averaged recall rate (evaluated on images with at least two agreeing experts) was only 22.2%–44.4% for supervised learning (figure 7(b)), only slightly higher than random guessing at 14.3%.

We next applied SimCLR-based semi-supervised learning to the Helgoland dataset. Since pretraining had already proceeded using unlabeled images from both datasets, no further training or computational costs were required for this, beyond the modest cost of supervised fine-tuning on 700 Helgoland labels. Semi-supervised learning improved performance by a much larger margin on this dataset, with fine-tuning on all of a single expert's labels improving overall accuracy to 80.5%–90.8% (figure 7(c), light blue dashes). Any expert's labels used for semi-supervised learning outperformed the same or any other expert's labels used for supervised learning. The class-averaged recall rate also increased to 58.6%–71.4% for all labels (figure 7(b)).

In order to make effective use of all, possibly conflicting expert labels for each image, we next carried out supervised learning and semi-supervised fine-tuning using a KL-divergence (D_{KL})-based loss function. In this approach, we treated the multiple expert labels for each image as a discrete probability distribution, and

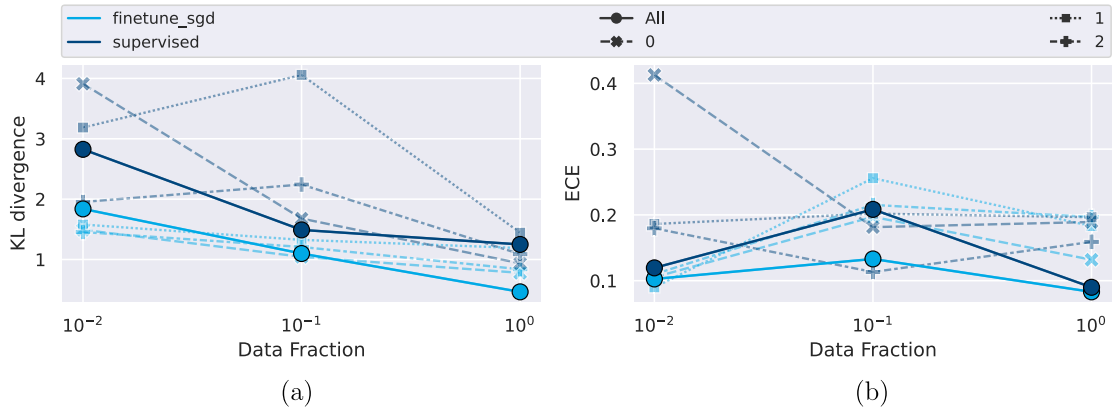

Figure 8. KL divergence (a) and expected calibration error (ECE) (b) averaged over the test dataset. Shaded lines refer to training using a single experts labels, full lines indicate training on the distribution of all expert labels using KL divergence as objective.

minimized the KL-divergence between this distribution and the network's inferred probabilities (details in 4.7). For semi-supervised fine-tuning, training on all labels with a D_{KL}-based loss yielded total accuracy and class-averaged recall as good as or better than networks trained on any individual expert's labels, when at least 10% of labels were used (figures 7(a)–(c), solid light blue). A clear benefit was not as apparent for supervised learning, for which some networks trained on a single expert's labels were better (figures 7(a)–(c), solid dark blue). Notably, for both total accuracy and class-averaged recall rate, semi-supervised learning with 10% of labels outperformed supervised learning with all labels. We also trained additional models and examined different fractions of training data (figure C3), and found 'linear evaluation' to perform surprisingly well compared to its results on the single-label dataset, where it was outperformed by every other approach. This could indicate that depending on the data, this cheaper training mode can provide a strong initial baseline.

The availability of multiple conflicting labels allows us to identify images for which the correct classification is ambiguous, so that a well-trained network should ideally infer nonzero probabilities across multiple classes. We therefore examined whether trained networks correctly exhibited uncertainty that matched the distribution of expert labels. This question cannot be answered by examining overall accuracy of class-specific recall, since these measures depend on only the most likely class inferred by the network, and not on the network's posterior distribution or the uncertainty it expresses. When measuring the D_{KL} between the inferred posterior and the expert, we found that semi-supervised training with a divergence-based loss and at least 10% of labels matched or outperformed all experts in minimizing divergence, while this was not the case for supervised training (figure 8(a)). We also measured the expected calibration error (ECE), which quantifies a network's tendency to be overconfident in its own predictions. Divergence loss-based training yielded ECE values similar to or better than the best expert at all data fractions for semi-supervised, but not supervised learning (figure 8). Together, these results show that D_{KL}-based training improves accuracy, better captures uncertainty and reduces overconfidence, but that semi-supervised learning is required to reap these benefits effectively in a label-poor scenario.

6. Discussion

We could not exhaustively and quantitatively compare our results to previously introduced approaches due to computational constraints and differences in datasets, which were acquired with a wide variety of imaging methods and systems (reviewed in (Pierella Karlusich *et al* 2022)). Nonetheless, overall our approach provided quantitative and useful improvements to the performance of deep image classifiers on *in situ* imaging of aquatic life and particles, in both label-abundant and label-poor scenarios. We demonstrated how to improve performance by applying class rebalancing, Bayesian probability correction, semi-supervised learning and a multi-label loss function. The utility of these techniques was shown using datasets that are image-rich but label-poor, as in most aquatic imaging studies. A key finding is that pretraining can reduce the amount of labeled data needed to train the final model by about 100-fold, while achieving similar accuracy to a fully supervised trained model on the complete data. Broadly speaking, our results are consistent with previous SimCLR and MOCO studies (Eldele *et al* 2022, Patel *et al* 2022, Yang *et al* 2023), which also reported improvements in accuracy and convergence speed of classifiers with unsupervised pretraining that become more apparent as the ratio of labeled to unlabeled data decreases.

Based on our results, one could conceivably argue that semi-supervised learning would no longer be required for plankton classification tasks concerning data with 150k or more available labels, as fine-tuning and supervised results mostly converge when the fraction of labels used approaches 100%. This is consistent with the original SimCLR paper (Chen *et al* 2020a) where when using 1% of the labels they achieved a ∼40% better performance and when using 10% of the labels the accuracy only improved ∼10% compared to supervised training. Our improvements were slightly less, but we account this to the differences in the total sizes of training data. However, the truly relevant factor for the utility of pretraining might actually be the ratio R of labeled to unlabeled data. In our case, when using 0.25% of labels $R = 643.7$, but for 100% of labels $R = 1.6$. We suspect that pretraining with a higher number of unlabeled images would therefore improve performance, even when using all available labels. Pretraining may also be especially useful for new, more powerful classifier architectures with higher parameter counts, which generally require more training data (Zagoruyko and Komodakis 2016, Dosovitskiy *et al* 2020, Dai *et al* 2021). We also note that, even for $R = 1.6$, pretraining gives a significant reduction in training time for the label-based task (figure 5). Surprisingly, we found that fine-tuning of all network weights is preferable to the frozen or linear training modes, for a computational budget as small as 1 min on a single GPU. For these reasons, we believe that unsupervised pretraining is likely to offer significant improvements to effective deep learning for marine science in the foreseeable future.

6.1. Future outlook

We anticipate that reducing the quantity of labels required for effective training will have widespread practical utility for marine science, especially in time- and computational resource-limited applications, such as on research voyages. While the pretraining itself remains computationally demanding, this step can be carried out in parallel to data-labeling, and in some cases before new image data are collected. One important open question is whether pretraining would still improve performance in the presence of domain shift, when unlabeled images come from a different imaging device, environment or species distribution. Nevertheless, our results suggest that the improvement of classification accuracy after pretraining will increase at higher ratios of unlabeled to labeled images.

While we have used datasets from different times and locations, covering multiple values for external factors such as temperature and salinity, more diversification in the data would be desirable. In future work, we aim to examine the effects of environmental factors and imaging systems on performance, and to measure long-term stability of performance. Another important research direction concerns optimally combining human and automated efforts. A semi-automated pipeline could return uncertain predictions to a human researcher for manual annotation. Those newly annotated images can afterward be used to update the network weights again in an iterative human-in-the-loop approach (Wu *et al* 2022). Our approach could also be applied to other classification tasks in marine science, such as automatic labeling of fish or cetaceans.

7. Conclusion

In this study, we presented an approach to classify under-water images of plankton using a multistep neural network training approach. In a first unsupervised training step, the images are used without any labels to pretrain the first part (feature extractor) of the final network, making use of the SimCLR framework. In a second step, expert-assigned labels are used for a small proportion of the images to fine-tune the neural network further from the first step in a supervised manner with artificially rebalanced data. After successful training, Bayes' rule is applied to readjust the learned label distribution back to the prior label distribution deduced from the unbalanced training data.

We showed that the gain from pretraining is especially high when the amount of labeled images is low, and that the approach works on single-labeled and multi-labeled data. We further demonstrated how the pipeline can be applied to multi-labeled images in general, and introduced a new metric to evaluate accuracy in the presence of multiple conflicting labels. Finally, the benefit of using conflicting labels assigned by multiple domain experts is elaborated, and we show its prospect to overcome overconfidence for the trained neural networks.

Our work enables domain scientists to analyze larger datasets using all collected image-data without the need to manually label or preprocess each image. We anticipate that the techniques we employed to address label scarcity, class imbalance and conflicting labels could reduce the time and cost of scientific studies, and allow more ambitious, difficult questions to be asked.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

We would like to thank Marie Flatow for helping us label, organize, and provide the data. Furthermore, the authors gratefully acknowledge the Gauss Center for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer JUWELS at Jülich Supercomputing Center (JSC). This work was funded by Helmholtz Association's Initiative and Networking Fund through Helmholtz AI. All visualizations of data in this manuscript were created using the seaborn python library (Waskom 2021). We further thank Marcel Nonnenmacher for useful discussions and Ankita Vaswani for helpful comments on the manuscript.

Appendix A. Example images

The images in figures A1 and A2 are examples from the training datasets from the Cape Verde and Helgoland datasets, respectively. The images were selected in no particular order. They are only included to give the reader a better impression of the collected data. Noticeably, the shown images from the Helgoland dataset have a higher proportion of red in them compared to the Cape Verde data. This is not only true for the shown images, but also existent for the whole dataset (labeled and unlabeled images)

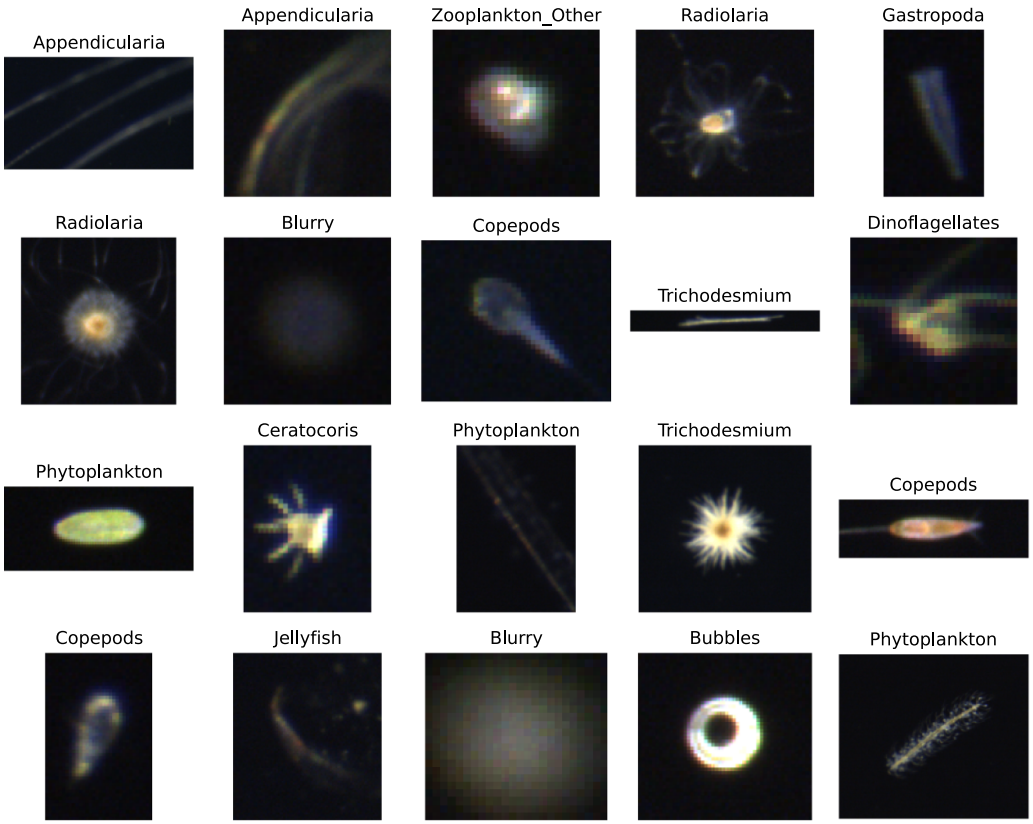

Figure A1. Randomly selected images from the Cape Verde training dataset. The images were resized by using bilinear interpolation, preserving the aspect ratio.

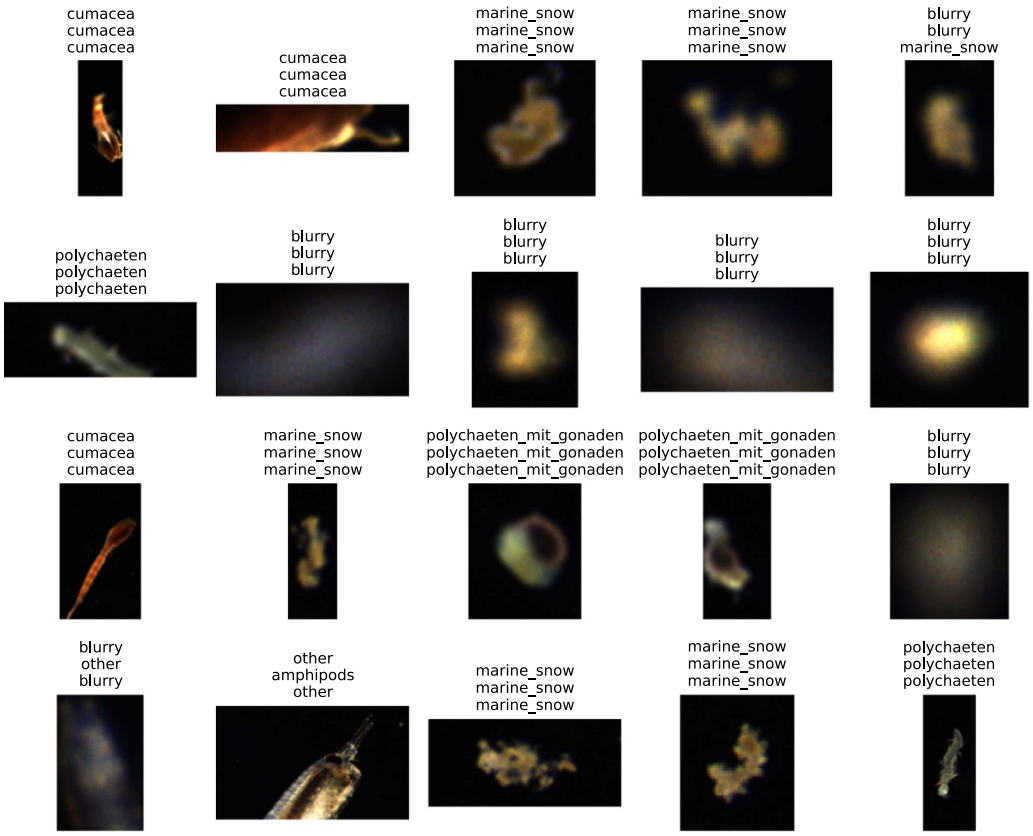

Figure A2. Randomly selected images from the Helgoland training dataset. The images were resized by using bilinear interpolation, preserving the aspect ratio. The label votes from the three human experts are printed above each image.

Appendix B. Predicted species frequencies

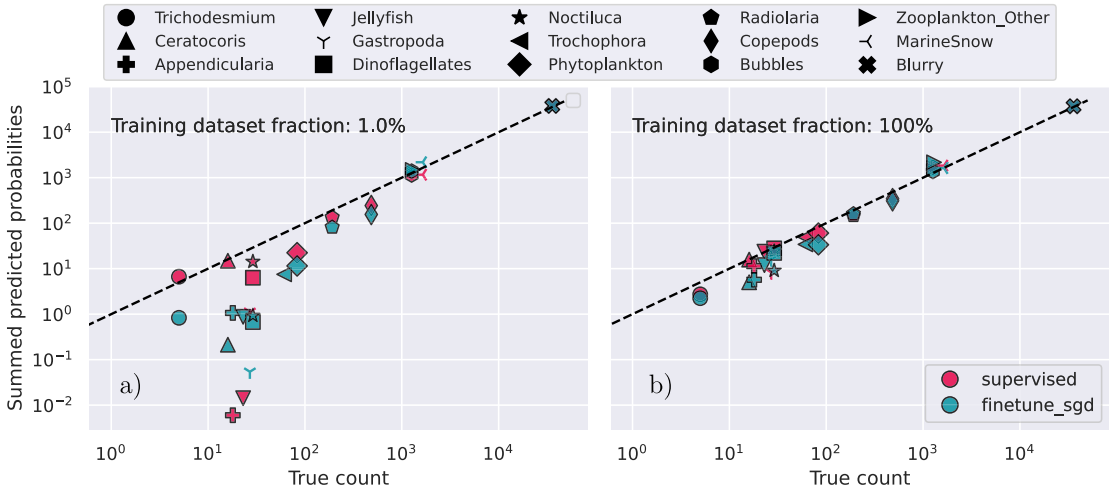

Figure B1. True species counts vs. summed predicted probabilities in the test dataset.

Appendix C. Further models and their results

We used one other configuration for the second training stage. This configuration adds a single linear classification layer, retrains all model weights and uses the ADAM optimizer. For naming conventions, we called this approach 'finetune'. It did not perform any better than the supervised model, but needs pretraining and adjusts all network weights during the second training stage. This makes it computationally as expensive as 'finetune SGD', but without its benefits. For completion, we report the accuracies of this model together with all other models (figure C1) and the training wall-clock times (figure C2).

Also for the multi-label training data, we examined the performance of multiple models, with every approach outperforming the supervised training (C3). This is a strong foundation for our claim that not only the amount of labeled training data is significant for the performance differences of pretrained vs. fully supervised trained models, but more the ratio R of labeled to unlabeled data might play an important role.

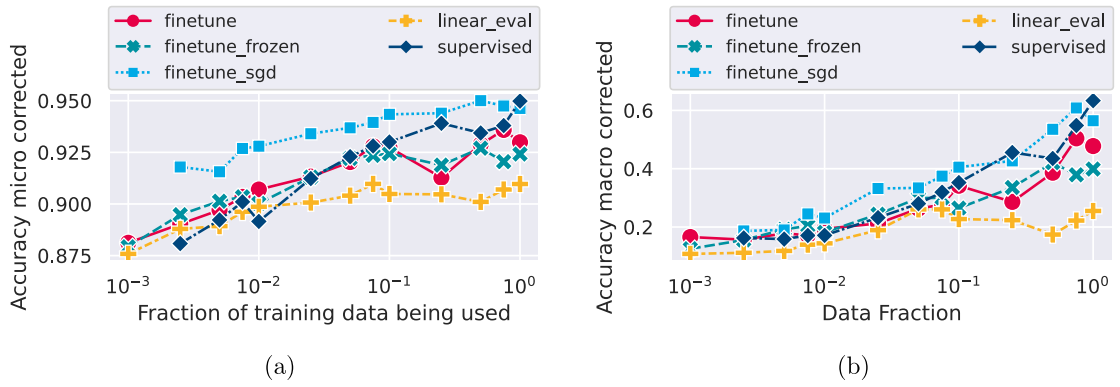

Figure C1. Micro (a) and macro (b) averaged accuracies for all tested models, including fine-tuning with the ADAM optimizer.

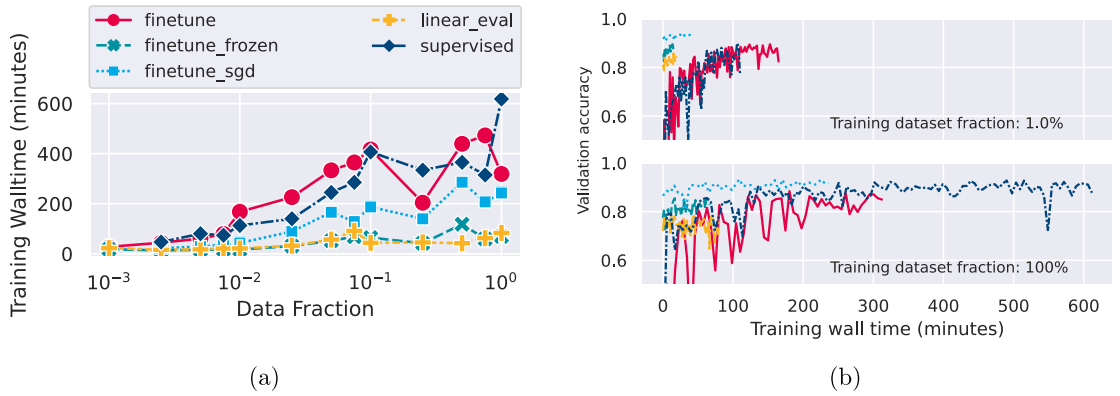

Figure C2. Wall-clock times for training the models. Since every model was trained until convergence with patience of 20 epochs, training times (a) strongly depended on the amount of training data and the trainable parameters. Change in validation accuracy over time (b) during training with 1% vs. 100% of labels used.

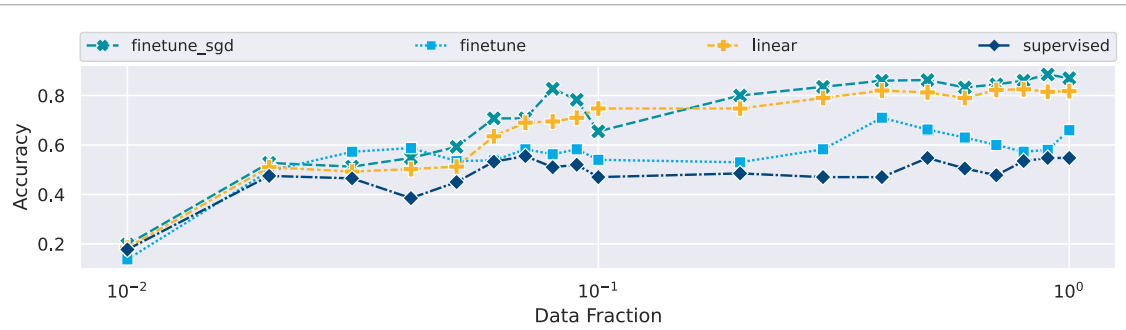

Figure C3. Custom accuracy on the held out test dataset at different fractions of data used during training on the multi-label (Helgoland) dataset.

Appendix D. Accuracies of corrected vs. uncorrected predicted probabilities

As described in the methods section 4.6.2 and shown in the results section 5.1, we corrected the networks' predicted probabilities using Bayes' formula. We also investigated the performance of every model without correction (figure D1, left column) but found an overall performance increase when using the corrected probabilities. Although the macro averaged accuracies are slightly lower than in the uncorrected case, the large increase in the micro averaged accuracies affirm and justify applying the correction. For the 'linear evaluation' and 'finetune frozen' models, the accuracies are even below the chance threshold of the two combined nuance classes, which together already make out ∼90% of the data. Therefore, any micro averaged accuracy below that threshold is considered worse than chance.

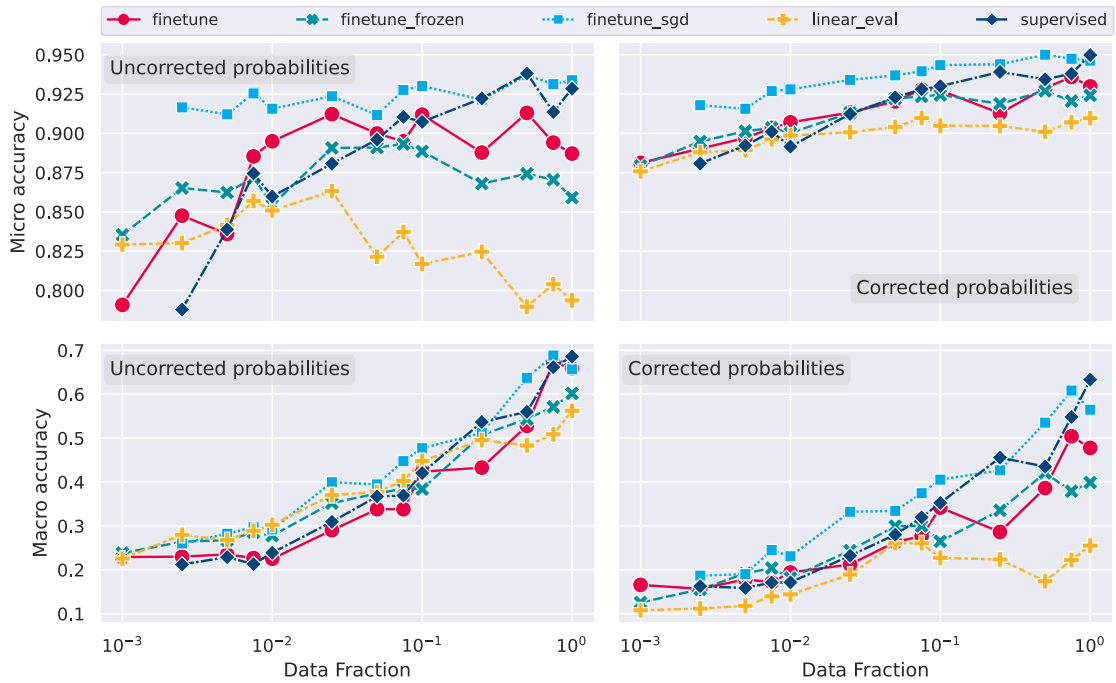

Figure D1. Accuracies of the predictions with raw model outputs (left) and posterior corrected probabilities (right).

ORCID iDs

Tobias Schanz https://orcid.org/0000-0002-8455-4179
Klas Ove Möller https://orcid.org/0000-0003-3118-2161
Saskia Rühl https://orcid.org/0000-0002-4650-6045
David S Greenberg https://orcid.org/0000-0002-8515-0459

References

TorchVision-maintainers and contributors 2016 Torchvision: Pytorch's computer vision library (available at: https://github.com/pytorch/vision)

Alzubaidi L, Zhang J, Humaidi A J, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel M A, Al-Amidie M and Farhan L 2021 Review of deep learning: concepts, CNN architectures, challenges, applications, future directions *J. Big Data* **8** 53

Bardes A, Ponce J and LeCun Y 2022 Vicreg: variance-invariance-covariance regularization for self-supervised learning (arXiv:2105.04906)

Bochinski E, Bacha G, Eiselein V, Walles T J W, Nejstgaard J C and Sikora T 2019 Deep active learning for *in situ* plankton classification *Pattern Recognition and Information Forensics (Lecture Notes in Computer Science)* ed Z Zhang, D Suter, Y Tian, A Branzan Albu, N Sidère and H Jair Escalante (Springer) pp 5–15

Branco P, Torgo L and Ribeiro R 2015 A survey of predictive modelling under imbalanced distributions (arXiv:1505.01658)

Caron M, Touvron H, Misra I, Jégou H, Mairal J, Bojanowski P and Joulin A 2021 Emerging properties in self-supervised vision transformers *Proc. IEEE/CVF Int. Conf. on Computer Vision* (*ICCV*) pp 9650–60

Chen T, Kornblith S, Norouzi M and Hinton G 2020a A simple framework for contrastive learning of visual representations (arXiv:2002.05709 [cs, stat])

Chen T, Kornblith S, Swersky K, Norouzi M and Hinton G 2020b Big self-supervised models are strong semi-supervised learners (arXiv:2006.10029 [cs, stat])

Chen X, Fan H, Girshick R and He K 2020c Improved baselines with momentum contrastive learning (arXiv:2003.04297)

Chollet F *et al* 2015 Keras (available at: https://keras.io)

Cover T M and Thomas J A 1991 Information theory and the stock market *Elements of Information Theory* (Wiley) pp 543–56

Dai J, Wang R, Zheng H, Ji G and Qiao X 2016 ZooplanktoNet: deep convolutional network for zooplankton classification *OCEANS 2016* (*Shanghai*) pp 1–6

Dai Z, Liu H, Le Q V and Tan M 2021 Coatnet: marrying convolution and attention for all data sizes *Advances in Neural Information Processing Systems*
vol 34, ed M Ranzato, A Beygelzimer, Y Dauphin, P Liang and J W Vaughan (Curran Associates, Inc.) pp 3965–77

Di Mauro R, Cepeda G, Capitanio F and Viñas M D 2011 Using ZooImage automated system for the estimation of biovolume of copepods from the northern Argentine Sea *J. Sea Res.* **66** 69–75

Dosovitskiy A *et al* 2020 An image is worth 16 × 16 words: transformers for image recognition at scale *ICLR*

Eldele E, Ragab M, Chen Z, Wu M, Kwoh C-K and Li X 2022 Self-supervised learning for label-efficient sleep stage classification: a comprehensive evaluation (arXiv:2210.06286)

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

Falcon W The PyTorch Lightning team 2019 PyTorch lightning (https://doi.org/10.5281/zenodo.3828935)
Garrido Q, Chen Y, Bardes A, Najman L and Lecun Y 2022 On the duality between contrastive and non-contrastive self-supervised learning (arXiv:2206.02574)
Gorsky G, Ohman M D, Picheral M, Gasparini S, Stemmann L, Romagnan J-B, Cawood A, Pesant S, García-Comas C and Prejger F 2010 Digital zooplankton image analysis using the ZooScan integrated system *J. Plankton Res.* **32** 285–303
Gu J *et al* 2018 Recent advances in convolutional neural networks *Pattern Recognit.* **77** 354–77
Guo B, Nyman L, Nayak A R, Milmore D, McFarland M, Twardowski M S, Sullivan J M, Yu J and Hong J 2021 Automated plankton classification from holographic imagery with deep convolutional neural networks *Limnol. Oceanogr.: Methods* **19** 21–36
He H and Garcia E A 2009 Learning from imbalanced data *IEEE Trans. Knowl. Data Eng.* **21** 1263–84
He K, Fan H, Wu Y, Xie S and Girshick R 2020 Momentum contrast for unsupervised visual representation learning *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (*CVPR*)
He K, Zhang X, Ren S and Sun J 2015 Deep residual learning for image recognition (arXiv:1512.03385 [cs])
Hewson I, Poretsky R S, Dyhrman S T, Zielinski B, White A E, Tripp H J, Montoya J P and Zehr J P 2009 Microbial community gene expression within colonies of the diazotroph, trichodesmium, from the southwest pacific ocean *ISME J.* **3** 1286–300
Jiao L, Jiao L, Zhao J and Zhao J 2019 A survey on the new generation of deep learning in image processing *IEEE Access*
Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)
Kolesnikov S 2018 Catalyst - accelerated deep learning r&d (available at: https://github.com/catalyst-team/catalyst)
Koller C, Kauermann G and Zhu X X 2022 Going beyond one-hot encoding in classification: can human uncertainty improve model performance? (arXiv:2205.15265)
Kraft K *et al* 2022 Towards operational phytoplankton recognition with automated high-throughput imaging, near-real-time data processing and convolutional neural networks *Front. Marine Sci.* **9** 867695
Kyathanahally S P, Hardeman T, Merz E, Bulas T, Reyes M, Isles P, Pomati F and Baity-Jesi M 2021 Deep learning classification of lake zooplankton *Front. Microbiol.* **12** 746297
Le K T *et al* 2022 Benchmarking and automating the image recognition capability of an *in situ* plankton imaging system *Front. Marine Sci.* **9** 869088
Li J, Chen T, Yang Z, Chen L, Liu P, Zhang Y, Yu G, Chen J, Li H and Sun X 2021 Development of a buoy-borne underwater imaging system for *in situ* mesoplankton monitoring of coastal waters *IEEE J. Ocean. Eng.* **47** 88–110
Li Z, Zhao F, Liu J and Qiao Y 2014 Pairwise nonparametric discriminant analysis for binary plankton image recognition *IEEE J. Ocean. Eng.* **39** 695–701
Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P and Zitnick C L 2014 Microsoft coco: common objects in context *Computer Vision – Eccv 2014* ed D Fleet, T Pajdla, B Schiele and T Tuytelaars (Springer) pp 740–55
Lumini A and Nanni L 2019 Deep learning and transfer learning features for plankton classification *Ecol. Inf.* **51** 33–43
Lumini A, Nanni L and Maguolo G 2020 Deep learning for plankton and coral classification *Appl. Comput. Inf.* **19** 265–83
Luo J Y, Irisson J-O, Graham B, Guigand C, Sarafraz A, Mader C and Cowen R K 2018 Automated plankton image analysis using convolutional neural networks *Limnol. Oceanogr.: Methods* **16** 814–27
Möller K O, John M S, Temming A, Floeter J, Sell A F, Herrmann J-P and Möllmann C 2012 Marine snow, zooplankton and thin layers: indications of a trophic link from small-scale sampling with the video plankton recorder *Marine Ecol. Prog. Ser.* **468** 57–69
Möller K O, Schmidt J O, St John M, Temming A, Diekmann R, Peters J, Floeter J, Sell A F, Herrmann J-P and Möllmann C 2015 Effects of climate-induced habitat changes on a key zooplankton species *J. Plankton Res.* **37** 530–41
Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* vol 32, ed H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox and R Garnett (Curran Associates, Inc.) pp 8024–35
Patel C, Sharma S, Pasquarella V J and Gulshan V 2022 Evaluating self and semi-supervised methods for remote sensing segmentation tasks (arXiv:2111.10079)
Pierella Karlusich J J, Lombard F, Irisson J-O, Bowler C and Foster R A 2022 Coupling imaging and omics in plankton surveys: State-of-the-art, challenges and future directions *Front. Mar. Sci.* **9** 878803
Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg A C and Fei-Fei L 2015 ImageNet large scale visual recognition challenge (arXiv:1409.0575 [cs])
Shafiq M and Gu Z 2022 Deep residual learning for image recognition: a survey *Appl. Sci.* **12** 8972
Sohn K 2016 Improved deep metric learning with multi-class n-pair loss objective *Advances in Neural Information Processing Systems* vol 29, ed D Lee, M Sugiyama, U Luxburg, I Guyon and R Garnett (Curran Associates, Inc.)
Sosik H M and Olson R J 2007 Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry *Limnol. Oceanogr.: Methods* **5** 204–16
Sun C, Shrivastava A, Singh S and Gupta A 2017 Revisiting unreasonable effectiveness of data in deep learning era *2017 IEEE Int. Conf. on Computer Vision* (*ICCV*) (*Venice, Italy*)
Vilgrain L, Maps F, Picheral M, Babin M, Aubry C, Irisson J-O and Ayata S-D 2021 Trait-based approach using *in situ* copepod images reveals contrasting ecological patterns across an arctic ice melt zone *Limnol. Oceanogr.* **66** 1155–67
Wang C, Yu Z, Zheng H, Wang N and Zheng B 2017 CGAN-plankton: towards large-scale imbalanced class generation and fine-grained classification *2017 IEEE Int. Conf. on Image Processing* (*ICIP*) pp 855–9
Waskom M L 2021 Seaborn: statistical data visualization *J. Open Source Softw.* **6** 3021
Wu X, Xiao L, Sun Y, Zhang J, Ma T and He L 2022 A survey of human-in-the-loop for machine learning *Future Gener. Comput. Syst.* **135** 364–81
Yadan O 2019 Hydra - a framework for elegantly configuring complex applications *Github*
Yang Y, Liu X, Wu J, Borac S, Katabi D, Poh M-Z and McDuff D 2023 Simper: simple self-supervised learning of periodic targets (arXiv:2210.03115)
Yeh C-H, Hong C-Y, Hsu Y-C, Liu T-L, Chen Y and LeCun Y 2022 Decoupled contrastive learning *Computer Vision – Eccv 2022* ed S Avidan, G Brostow, M Cissé, G M Farinella and T Hassner (Springer) pp 668–84
You Y, Gitman I and Ginsburg B 2017 Large batch training of convolutional networks (arXiv:1708.03888 [cs])
Zagoruyko S and Komodakis N 2016 Wide residual networks (arXiv:1605.07146)
Zheng H, Wang R, Yu Z, Wang N, Gu Z and Zheng B 2017 Automatic plankton image classification combining multiple view features via multiple kernel learning *BMC Bioinform.* **18** 570
Zhu X and Wu X 2004 Class noise vs. attribute noise: a quantitative study *Artif. Intell. Rev.* **22** 177–210