sensors

MDPI

Article

MorphoCluster: Efficient Annotation of Plankton Images by Clustering

Simon-Martin Schröder [1,*], Rainer Kiko [2,3] and Reinhard Koch [1]

[1] Department of Computer Science, Kiel University, 24118 Kiel, Germany; rk@informatik.uni-kiel.de
[2] Laboratoire d'Océanographie de Villefranche-sur-mer, 06230 Villefranche-sur-Mer, France; rainer.kiko@obs-vlfr.fr
[3] GEOMAR Helmholtz Center for Ocean Research Kiel, 24148 Kiel, Germany
* Correspondence: sms@informatik.uni-kiel.de

Received: 30 April 2020; Accepted: 25 May 2020; Published: 28 May 2020

Abstract: In this work, we present MorphoCluster, a software tool for data-driven, fast, and accurate annotation of large image data sets. While already having surpassed the annotation rate of human experts, volume and complexity of marine data will continue to increase in the coming years. Still, this data requires interpretation. MorphoCluster augments the human ability to discover patterns and perform object classification in large amounts of data by embedding unsupervised clustering in an interactive process. By aggregating similar images into clusters, our novel approach to image annotation increases consistency, multiplies the throughput of an annotator, and allows experts to adapt the granularity of their sorting scheme to the structure in the data. By sorting a set of 1.2 M objects into 280 data-driven classes in 71 h (16 k objects per hour), with 90% of these classes having a precision of 0.889 or higher. This shows that MorphoCluster is at the same time fast, accurate, and consistent; provides a fine-grained and data-driven classification; and enables novelty detection.

Keywords: machine learning; deep learning; clustering; plankton image classification; marine image recognition; marine image annotation

1. Introduction

Current plankton imaging tools (e.g., ZooScan [1], UVP5 [2], ISIIS [3], FlowCytoBot [4], IFCB [5], or ZooImage [6]) deliver highly diverse and constantly growing plankton image data sets [7,8] that contain thousands, and sometimes millions, of images sorted into a varying number of classes [9]. It is expected that the volume and complexity of marine data will increase by orders of magnitude in the coming years [10]. Ecological analyses of these samples require accurate object counts to enable abundance estimates. Object counts can be acquired by different means [11] but, most often, each object is classified individually and the objects of each class are counted (*classify-and-count*). This confronts the field of marine ecology with the challenge of providing taxonomic identifications for enormous volumes of imaging data efficiently. The annotation rate of human experts is long surpassed by the amount of data that is generated [12]. Therefore, advanced automatic image recognition techniques are indicated. These should liberate taxonomy experts from the tedious task of routine identifications [13]. However, to extract valuable insights from the data, moderation of automatic techniques is imperative [10].

Published *marine image annotation software* [14] tools include photoQuad [15], VARS [16], Seascape [17], and BIIGLE [18]. Beyond that, there are several tools not formally published, like SQUIDLE+ [19], ZooImage [6] and EcoTaxa [20], or the older Plankton Identifier [21] and ZooImage [22]. Some tools address

the annotation of whole frames where objects of interest have to be localized first to be classified afterward. However, plankton image data usually has a uniform background, so no semantic segmentation is needed. Other tools are therefore specifically targeted towards the annotation of individual plankton images.

EcoTaxa [20] is a web-application for the semi-automatic annotation of large image data sets of individual plankton images. We and other colleagues have been using it to sort UVP5 data for more than five years. During this time, we noticed that we—due to time constraints—often accept the automatic predictions for less interesting categories (*default effect*), we aggregate differently-looking objects due to taxonomic knowledge, and we focus only on the categories that are presumably relevant for the particular study. For example, great effort went into the sorting of different Rhizaria [23] or finding instances of *Poeobius* sp. [24].

Generally, researchers aim to annotate the objects according to a certain scientific goal, e.g., they sort all images of animals into accepted taxonomic units. This means that, e.g., different views (dorsal, lateral) of the same animal are grouped, although they might look very different. Furthermore, taxonomic knowledge is applied when combining different taxonomic units into higher-order groupings (e.g., copepods, euphausiids, and their larval stages into the subphylum crustacea). On the other hand, a very detailed sorting of other parts of the data set is not done, although very different image classes do exist in this part. Fecal pellets, aggregates, and fibers might all be summarized under the term detritus. Typically, only a few tens of classes are used in plankton studies based on imaging data [25–29], and the number of classes depends on the imaging instrument, sample location, and research interest. This *interest-driven* data annotation approach—that is also encouraged in EcoTaxa and other tools—might be most feasible for exclusively manual annotation, as it saves time, but it could be relatively problematic to automatically classify images into a set of so-defined classes.

Previously, shallow models, like Support Vector Machines [30] or Random Forest [31], with handcrafted local features measured on the image (e.g., size, gray level distribution, etc.) were used to classify plankton [1,5,29,32,33]. In recent years, however, there has been a transition towards deep plankton image recognition models based on convolutional neural networks (CNNs) [24,27,28,34,35].

Automatic classifiers require enough training data for each class. Especially, all classes need to be known and well-represented in the training data. Plankton image data contains a variety of dead matter, plankton of different size, morphology and orientation, and aggregations of multiple objects [12]; therefore, it presents a considerable challenge for image recognition. This problem is further complicated because we observe a long-tailed abundance distribution of plankton in the wild [25,28]. While some of the ocean's inhabitants can be witnessed nearly everywhere, others are seldom seen at all. Even if we knew which classes to expect in the sample, many could not possibly be represented in the training data because they were never annotated beforehand [36]. A classifier with a fixed set of classes prevents us from ever detecting anything new and unexpected. Such objects will be forced into the known classes and "disappear". Therefore, the analysis can only provide insights that are compatible with the initial question and classification granularity and does not necessarily extend to the full information which the current sample actually provides.

Apart from them not being complete, reliance on training sets has further weaknesses: First, they might deviate from the distribution of the collected sample. In the case of *classify-and-count*, this could in some cases distort the abundance estimates severely [11]. Second, a consensus on the identification of objects is hard to obtain in practice [37], so training sets—like every collection of annotated real-world data—exhibit some inconsistencies.

Consequently, the incoming data has to be constantly monitored, meaning that the automatic classifications are often manually validated by experts [20]. Given the growing amount of data, this will prove less and less feasible. In Reference [24], the polychaete *Poeobius* sp. was only found in an Underwater Vision Profile 5 data set, after it was seen in underwater videos taken in parallel with the PELAGIOS [38].

A mostly manual examination of 1.8M UVP5 images from the Eastern Tropical Atlantic then yielded 450 images of *Poeobius* sp.

When objects are sorted manually, several human factors, like cognitive biases, fatigue, and boredom [37], influence the classification.

These factors altogether—dependence on training data, a fixed set of classes, changing long-tailed distributions, growing amounts of data, and adversarial human factors—limit the accuracy and utility of interest-driven data annotation. Instead, we argue for *data-driven* image sorting using *unsupervised machine learning techniques* in order to be able to define all classes in the data set, as well as to spot novelties and unexpected patterns and derive reliable abundance estimates.

MorphoCluster

In this work, we present MorphoCluster, a tool for data-driven, fast, and accurate annotation of large data sets of single object images. Although we present and discuss the tool in the context of marine image annotation, it should be applicable in many areas with similar data sets (images of individual objects).

Considering the strength of deep neural networks to learn distinctive features [39], we hypothesize that it is feasible to cluster these features to partition a plankton image data set in a meaningful way.

We therefore combine unsupervised clustering with an interactive tool to revise the initial clusters, arrange them hierarchically, manually correct the hierarchy, and annotate the clusters. The annotator can therefore explore the groupings inherent in the data and spot novelties and unexpected patterns. By annotating groups of similar images as a whole, we intend to enable the consistent manual review of large amounts of data in a rather short time.

In the following, we will show that, by paying attention to the cluster structure of a data set, MorphoCluster is at the same time fast, accurate, and consistent; provides a fine-grained and data-driven classification; and enables novelty detection.

2. Methods

In this section, we present the overall structure of the MorphoCluster approach and the details of our implementation.

2.1. General Overview of the MorphoCluster Process

The MorphoCluster process is outlined in Figure 1. First, a deep feature extractor is trained to obtain features that encode relevant object properties for the task of plankton recognition and therefore enable efficient clustering. Then, the entire data set is clustered using HDBSCAN* with settings that allow for the creation of large and homogeneous clusters. In the *cluster approval* phase, visually pure clusters are validated and mixed clusters are manually rejected. During *cluster growing*, the remaining pure clusters are used as seeds to find additional visually similar objects. The samples that are not assigned to a cluster after the growing step are re-clustered with a less restrictive setting that yields smaller clusters than in the previous round. Cluster approval and growth steps are thereafter repeated. The described process is conducted iteratively with less and less restrictive settings until no further meaningful clusters are found. Thereafter, the identified clusters are hierarchically arranged using agglomerative clustering to group similar clusters. The clusters and branches of the resulting tree can then be inspected manually, very similar clusters can be merged, and clusters and branches can be named in a user-defined manner. Validation, growing, and naming are conducted in a specially developed web application available at https://github.com/morphocluster.

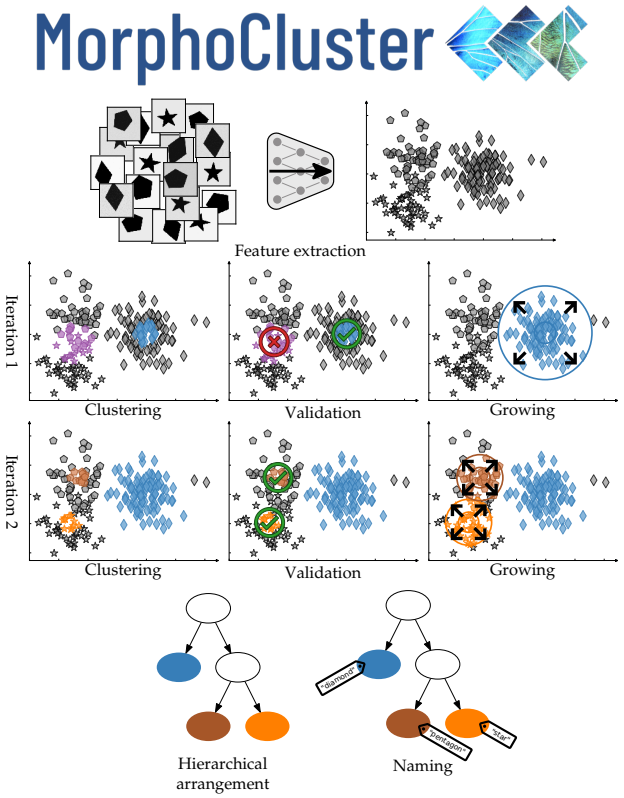

Figure 1. Overview of the MorphoCluster method. Images are projected to the feature space. Iteration 1: The blue cluster is validated and grown, while the purple one is rejected. Iteration 2: The orange and brown clusters are validated and grown. Finally, the clusters are arranged in a hierarchy and named.

2.2. Data Set Used

We evaluate our approach on a data set [40] of readily segmented grayscale images of individual particles in the water column which were acquired using the Underwater Vision Profiler 5 (UVP5) [2] in various regions of the world's oceans between 2012 and 2017. The depicted objects are very small (100 µm to several centimeters) and their orientation is unrestricted.The data set contains 1 M unlabeled images and 584 k labeled images that were sorted by experts into a selection of 65 classes from a taxonomy based on the widely recognized WoRMS [41] taxonomy using EcoTaxa. In that, the data set is similar to the ZooScanNet data set [8].

We call the initially ***u****nlabeled* set of images $\mathcal{U}$ and the initially ***l****abeled* set $\mathcal{L}_0$. The labeled data shows a severe class imbalance; the 10% most populated classes contain more than 80% of the objects and the class sizes span four orders of magnitude.

Like Orenstein and Beijbom [28] and Malde and Kim [36], we assume that the training set will be sufficient to learn features suitable for the distinction of all known and novel categories alike and that the distance in the feature space between two objects serves as a proxy for their similarity. To evaluate the ability of MorphoCluster to detect novel classes, we select four ***i****ndicator classes* $\mathcal{C}_\mathrm{i}$ (Veliger, Poeobius, T001, Flota) that are not used in the supervised training step.

The labeled set $\mathcal{L}_0$ is split into a ***t****raining set* $\mathcal{L}_t$ of 392 k objects and a ***v****alidation set* $\mathcal{L}_v$ of 192 k objects, stratified by class. $\mathcal{L}_t$, without the indicator classes $\mathcal{C}_\mathrm{i}$, is used to train the feature extractor. $\mathcal{L}_v$ is first used to monitor the feature extractor training (ignoring $\mathcal{C}_\mathrm{i}$) and later to evaluate the main MorphoCluster sorting process (including $\mathcal{C}_\mathrm{i}$).

To validate the outcome of the MorphoCluster progress, we combine $\mathcal{L}_v$ and $\mathcal{U}$ and sort them jointly. $\mathcal{L}_v$ enables us to map the categories annotated with MorphoCluster to the annotations made with EcoTaxa. The included indicator classes $\mathcal{C}_\mathrm{i}$ enable us to check if the MorphoCluster process allows detecting novel classes that the feature extractor was not trained on.

2.3. Supervised Training and Feature Extraction

The supervised training of the feature extractor is a preliminary step to acquire knowledge about the discriminative features of the data at hand. Transfer learning [39] allows the reuse of information provided by labeled samples to obtain features that are actually relevant to taxon identification.

The images of the training and validation sets $\mathcal{L}_t$ and $\mathcal{L}_v$ (excluding the indicator classes $\mathcal{C}_\mathrm{i}$) are used to train the network and monitor the classification loss, respectively. We select a ResNet18 [42] as the backbone of the feature extractor as it shows a favorable accuracy-speed trade-off compared to other network architectures [43]. The network is initialized with weights pre-trained on the ImageNet data set [44] and fine-tuned to the classification task at hand following the common practice [45]. To counter the class imbalance in the training set, we randomly sample up to 250 images from each class for each training epoch independently. Early stopping is used to avoid overfitting. The initial learning rate is set to 1×10^{-4} and decreased whenever the validation loss (measured on $\mathcal{L}_v$) plateaus until it reaches 1×10^{-8}. To consider all classes equally, we weight the validation loss by the inverse class size. The batch size is set to 128 images. The images are cropped to their tight bounding box and padded to a square with a minimum edge length of 128 px. Images larger than 128 px are shrunken to this size. The gray values are scaled to the $[0, 1]$ range. We perform training-time augmentation using random rotations in 90° steps, random horizontal and vertical flips, and additive Gaussian noise with $\sigma = 0.001$. The models are trained using the PyTorch deep learning library [46] on a NVIDIA GeForce GTX 1070 GPU.

Originally, the ResNet18 network produces a 512d feature vector for each image. In a fine-tuning step, an additional layer is trained to reduce the number of features to 32 to reduce computation time and storage requirements in the subsequent steps.

After removing the classifier layer, the decapitated network serves as a feature extractor. It is used to calculate feature vectors for all images in the data set (including labeled and unlabeled images).

2.4. Clustering

In this completely unsupervised stage, the images of the unlabeled set $\mathcal{U}$ and the validation set $\mathcal{L}_v$ (including the "novel" indicator categories $\mathcal{C}_\mathrm{i}$) are clustered jointly according to their feature vectors generated in the previous step.

To accumulate similar objects, we use the hierarchical density-based HDBSCAN* algorithm [47,48] which has some favorable properties: It handles clusters of variable density, makes few assumptions about the data distribution, has a small number of parameters, and is robust to outliers. Another remarkable property is that HDBSCAN* clusters only the densest regions of the feature space and rejects most of the objects as noise. This is favorable in our setting, since this way, the resulting clusters are very pure.

HDBSCAN* is parameterized by the neighborhood size k and the minimum cluster size m. We set neighborhood size $k = 1$ and vary minimum cluster size m throughout the iterations of validation and growing. Initially, a large value (e.g., 128) is chosen for m to extract the largest coherent groups first. It is decreased after each iteration of the process so that increasingly smaller clusters are found. This strategy is used to remove large groups of similar objects early in the process and to keep the number of clusters to be validated and grown in each step small. Too small values for m would lead to excessive fragmentation of the data resulting in many small clusters leading to a drastically increased effort in the following steps.

The detected dense regions of the feature space serve as *cluster seeds* for the further treatment in the following steps.

2.5. Cluster Validation

Figure 2 shows the user interface for manual cluster seed validation and review. One after the other, each cluster seed is displayed to the user. Its images are arranged in an alternating fashion so that two neighboring images are maximally dissimilar with respect to their deep learning features. The resulting contrast facilitates the annotator's judgment. The user then flags homogeneous cluster seeds as "validated". Impure cluster seeds are deleted and the corresponding objects are returned to the pool of unclustered objects.

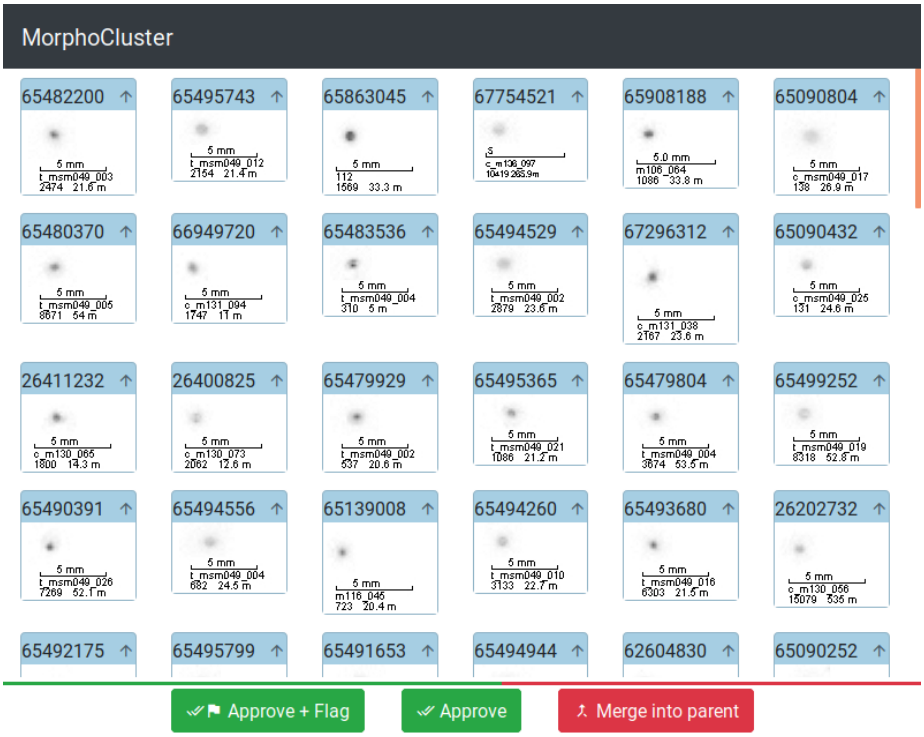

Figure 2. User interface for cluster validation. The images of a cluster are presented to the user. "Approve" marks a cluster as validated (=being pure). "Approve + Flag" additionally flags the cluster for preferred treatment during the growth step. "Merge into parent" deletes a cluster and moves its objects back to the pool of unclustered objects. Above the buttons, a progress indicator is visible.

2.6. Cluster Growing

After validation, only pure cluster seeds are left. Due to their construction (see Section 2.4), a seed is only the very core of a dense region. The purpose of *cluster growing* is therefore the accretion of further images from the neighborhood of this dense region until the boundaries of a cluster are reached.

For each cluster, the objects that make up the cluster seed are presented to the user (Figure 3). The objects that are so far no member of any cluster are displayed as *recommended members* ordered by decreasing similarity to the cluster seed (measured by their distance to the seed's centroid). The user then needs to find the first object in the list of recommended members that is not similar to the seed images. Finally, the objects earlier in the list (being more similar) are added to the cluster. This setup is similar to

the visual search engine in Reference [49]. The list of recommended members is partitioned into pages of 50 objects that are reviewed jointly.

The application assists in finding the similarity threshold by employing binary search to minimize the number of objects that a user has to review. In the first stage of the task, the right limit of the search interval (a point where all objects are strictly dissimilar) is determined: Beginning with the first page, the images of selected pages are reviewed if they match the seed images. The number of pages that are skipped between successive page reviews is doubled in each step. If the images start to differ from the seed images, the right limit of the search interval for the cluster radius is found.

Subsequently, the actual binary search step narrows down the search interval to find the last page with matching candidate objects. Because many objects are never seen by the user, the process is much faster than adding each object to the cluster individually.

This approach is permitted under the assumption that if all objects on a certain page are sufficiently similar to the seed, all objects of the previous pages are also similar to the seed.

A so-called "turtle mode" allows for a very detailed examination and definition of the cluster border by allowing single objects to be removed from the set of recommended members. Once an individual object is removed from the current page, turtle mode is activated and binary search is disabled. Now, in turtle mode, all remaining objects have to be validated individually, and the speed-up provided by binary search is traded for accuracy.

2.7. Cluster Naming

After the objects are treated and moved to clusters, these clusters are named with computer-assistance using the respective function of the MorphoCluster application. To this end, the list of clusters is transformed into a hierarchy by agglomerative clustering of the cluster centroids using average linkage (UPGMA) clustering [50] (p. 76). The resulting automatic hierarchy serves as a starting point for a user-defined taxonomy. Arranging clusters in a hierarchy makes them easier to annotate because many of the clusters found in the previous steps are very similar and can be given the same name or fall into the same superclass. Their similarity in the feature space makes them close neighbors in the thus defined tree. The tree is presented to the annotator, who can merge clusters if they are perceived as being identical. The annotator can also rearrange individual nodes and give them names. To this end, we started at the leaves of the tree and worked our way up to the root. Whenever a node looked different than its siblings, it was given a distinct name and moved up in the hierarchy. In the end, the name of each node was transferred to its corresponding objects. The resulting set of now labeled images is called $\mathcal{L}_{MC}$.

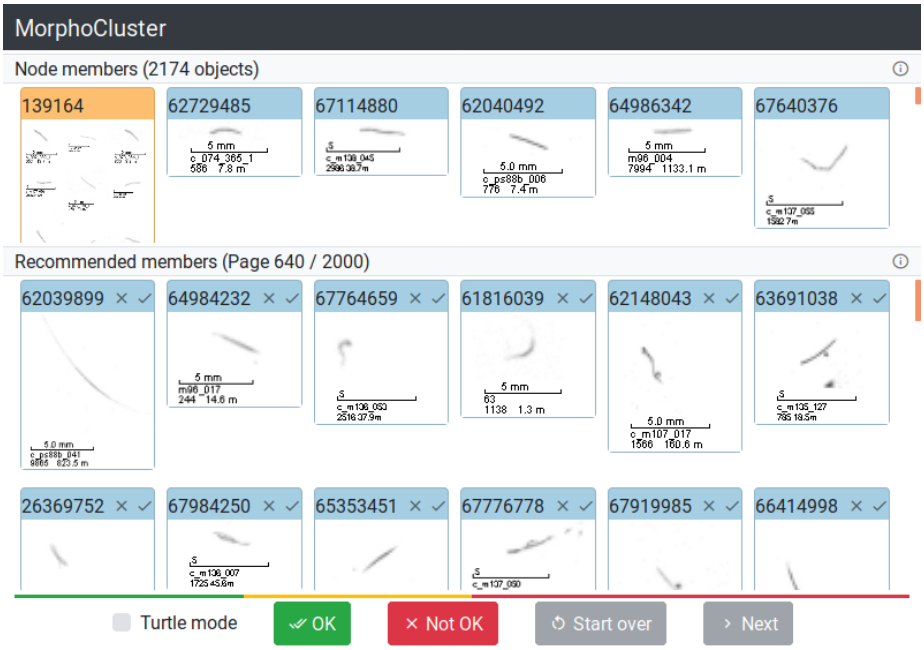

Figure 3. User interface for growing clusters. The top half of the screen displays the current member objects of a cluster. The bottom half always shows a page of 50 candidate members. The colored bar above the buttons visualizes the search interval for the pages of candidates that should be added to the cluster. Pages in green were judged to match the cluster, and pages in red were judged not to match. Pages in yellow were not reviewed yet.

2.8. Experimental Approach

We applied the entire process of clustering, cluster approval, cluster growth, and naming to the combination of images from the unlabeled set $\mathcal{U}$ and the validation set $\mathcal{L}_v$ (including the indicator classes $\mathcal{C}_i$). Annotator actions were tracked during the approval, growth and naming steps to monitor the time spent during each step. To account for longer breaks, the log was split into *sessions* that contained no breaks longer than ten minutes. The duration of a session is the time span between its first and last entry.

For the evaluation of their precision, up to 500 objects per class (some classes are smaller) were randomly sampled from $\mathcal{L}_v$, for $\mathcal{L}_{MC}$ only 400, due to the larger number of classes. The samples of each class were manually reviewed and outliers (false positives) were removed. The precision of a category is then the fraction of inliers.

The precision of $\mathcal{L}_{MC}$ and $\mathcal{L}_0$ in this analysis is a measure of self-consistency because the same person (R. Kiko) that did the sorting in MorphoCluster and in large parts that of the initial data set also evaluated the sub-samples.

2.9. Evaluation Metrics

The *precision* of a class c is the number of objects *correctly* classified as c (true positives) divided by the total number of objects classified as c (true positives and false positives):

$$Pr_c = \frac{TP_c}{TP_c + FP_c}. \tag{1}$$

Macro precision is the arithmetic mean of all individual precisions:

$$\overline{Pr} = \underset{c}{\text{mean}}\, Pr_c\,. \tag{2}$$

Given two different labelings $\mathcal{L}_a$ and $\mathcal{L}_b$ of the same objects, we define the *relative overlap* of two classes c_a from $\mathcal{L}_a$ and c_b from $\mathcal{L}_b$ as the number of objects that are assigned to *both* c_a and c_b divided by the number of objects assigned to *either* of them:

$$RelOverlap = \frac{|c_a \cap c_b|}{|c_a \cup c_b|} = \frac{|c_a \cap c_b|}{|c_a| + |c_b| - |c_a \cap c_b|}\,. \tag{3}$$

3. Results

3.1. Supervised Training

The trained classifier achieved comparatively low scores even when using the full set of 512 feature dimensions (Table 1). This could be expected as the overall macro precision of the training set $\mathcal{L}_0$ was also only 0.738, with some classes showing very low precision (Figure 4; left). The feature reduction to 32 dimensions did not compromise classification performance substantially and even increased macro precision by a small amount (Table 1). We did not optimize the hyper-parameters of the network for high classification scores to maintain its generalization capabilities as a feature extractor.

Table 1. Accuracy and macro precision of the classifier trained for feature extraction before (512d) and after dimensionality reduction (32d). Dimensionality reduction did not substantially change the capacity of the classifier.

	512d	**32d**
Accuracy	0.561	0.557
Macro Precision	0.294	0.302

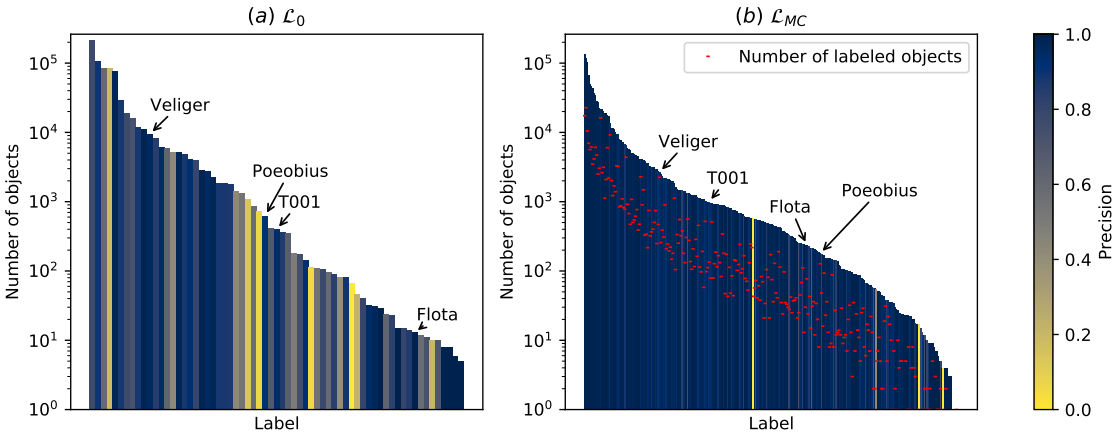

Figure 4. Classes in the initial labeling $\mathcal{L}_0$ (**a**) and the MorphoCluster labeling $\mathcal{L}_{MC}$ (**b**) ordered by their size. Four indicator classes $\mathcal{C}_i$ (Veliger, Poeobius, T001, Flota), indicated by arrows, are used to evaluate the ability of MorphoCluster to detect novel classes. The class sizes of $\mathcal{L}_0$ and $\mathcal{L}_{MC}$ are in the same range, but the latter contains many more classes. The precision of each class is color-coded (see Section 3.5). The number of objects from $\mathcal{L}_0$ in each class of $\mathcal{L}_{MC}$ is denoted in red. It is roughly one order of magnitude lower than the MorphoCluster class size.

3.2. MorphoCluster Efficiency

The metrics collected during the iterative cluster validation and growing steps of the MorphoCluster process are depicted in Table 2 and Figure 5. The number of clusters found in each iteration increased as a function of the minimum cluster size m. Most of the proposed cluster seeds were validated which indicates that the calculated clusters are in fact very pure. Only a few objects were assigned to clusters during the validation phases because the cluster seeds consist only of the densest regions. Growing a cluster added a large number of objects from the neighborhood of a cluster and the majority of objects were assigned to clusters during growing. During the first rounds of validation and growing, very large clusters were identified that mainly contained detritus-like objects. During later rounds, smaller clusters containing more rare objects (e.g., copepods, veliger larvae, etc.) were validated and grown. Figure 5 shows the number of objects sorted per hour during the entire MorphoCluster process. Most time was spent in the validation and growing steps to group similar parts of the data set and assignment of names to the identified clusters only accounts for a fraction of the total time. Validation and growing alone took 58.7 h. When considering these steps in isolation, 20,085 objects were sorted per hour. Naming took 12.2 h. The first three rounds of validation and growing yielded remarkably high sorting speeds (Figure 5). After that, sorting got drastically slower in each iteration.

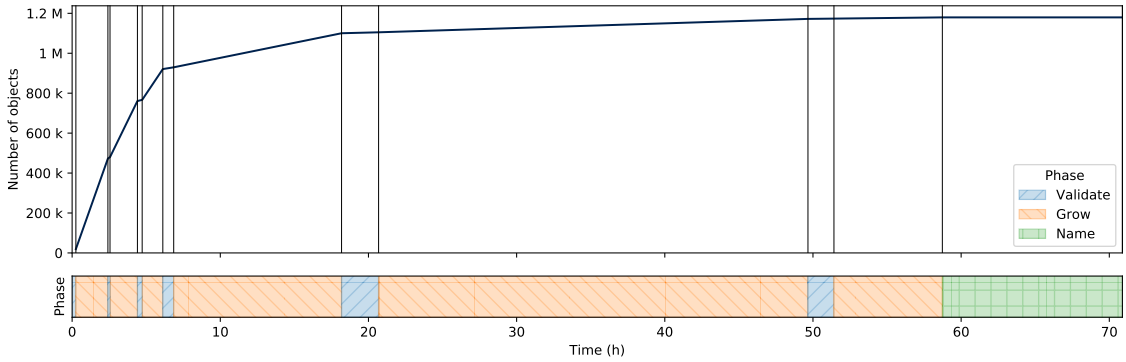

Figure 5. Number of validated objects during data annotation. The time periods are colored according to their respective phase.

Table 2. Iterations in the MorphoCluster process with metrics in each step. Minimum cluster size m; number of proposed new clusters; number of validated clusters; number of objects sorted per hour. Note that, at this point, the raw clusters have not been grouped and named yet.

Iteration	m	New Clusters	Validated Clusters	Objects Sorted per Hour
1	128	37	26	195779
2	64	51	49	144559
3	32	110	100	93333
4	16	299	288	14875
5	8	447	438	2282
6	4	612	291	834
Total		1556	1192	20085

3.3. Hierarchical Ordering and Naming

Figure 6 displays the 1192 unordered clusters as the result of the iterated clustering, approval and growing (left), their automatic hierarchical organization (middle) with 2382 nodes, and the revised hierarchy after reordering and naming (right) with 280 named branches (bold) in 26 broad supercategories

(colored). It is apparent that the initial hierarchy already introduced a high level of order and contained large branches that were pure with respect to the considered supercategories. However, branches that belong to the same supercategory according to expert knowledge were still scattered throughout the tree. To obtain the final result (right), these branches were manually mounted to a common supercategory, and relevant branching points were named using free-form input. This also reduced the depth of the tree from 23 to 12. The final result illustrates yet again that these supercategories are finely branched. Re-arranging the initial hierarchy and naming the branches took 12.2 h, only 17.1 % of the total time.

Considering this step in isolation, 97,068 objects, or 23 complete classes, were labeled per hour. Including validation, growing and naming, we spent a total 70.9 h on sorting 1,179,619 objects into a set of 280 new categories (16,641 objects per hour), while most objects were already sorted in the early steps.

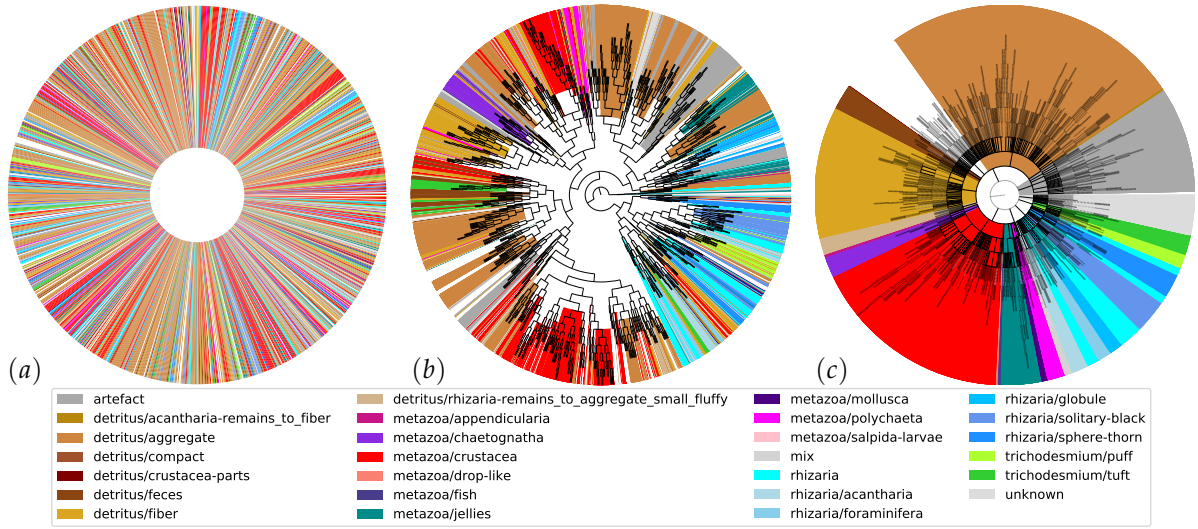

Figure 6. Unordered nodes (**a**), automatic hierarchy (**b**), and revised hierarchy with named branches denoted by bold lines (**c**). Corresponding sections of the three charts are colored alike according to broad supercategories. See the Supplementary Materials for the labels contained in these supercategories.

3.4. Completeness

Sixteen thousand four hundred *residual objects* (1.37 % of all objects) were not assigned using the MorphoCluster approach because they were neither clustered and validated nor moved to an existing cluster in the growing step. They were ultimately left untreated.

Fifty-eight of the 65 classes in the initial labeling $\mathcal{L}_0$ were reproduced in the new labeling $\mathcal{L}_{MC}$, while objects from some initial classes (*Annelida_Polychaeta*, *Crustacea_leg*, *Diplostraca_Cladocera*, *Euopisthobranchia_Thecosomata*, *Mollusca_Cephalopoda*, *Pyrosomatida_Pyrosoma*, *Solmundella_Solmundella bitentaculata*, *detritus_light*, *othertocheck_darksphere*, *temporary_t009*) could not be reproduced. In part, their objects were not put into any class at all, and in part their objects were included in other classes. All of these categories contain less than 40 objects and/or show high intra-class variability. Moreover, images of *Pyrosomatida_Pyrosoma* (large colonies of individual animals) are very large, and down-scaling them to the fixed input size of the feature extractor network removes nearly all of their distinctive features.

3.5. Accuracy

Using MorphoCluster, a very large fraction of classes was sorted with high precision. Figure 4 shows the class size and individual precision per class, which is consistently higher for $\mathcal{L}_{MC}$ compared to $\mathcal{L}_0$. Roughly a tenth of the objects in each class in $\mathcal{L}_{MC}$ was already labeled in $\mathcal{L}_0$ (red), which allows

calculating the agreement between both labelings. Table 3 shows this agreement ($\mathcal{L}_{MC}$ vs. $\mathcal{L}_v$) and also the macro precision of $\mathcal{L}_{MC}$ and $\mathcal{L}_0$ individually.

Table 3. Comparison of precision. The columns show the macro precision of MorphoCluster according to the original labels ($\mathcal{L}_{MC}$ vs $\mathcal{L}_v$) and the macro precision of $\mathcal{L}_{MC}$ and $\mathcal{L}_0$ according to manual examination. $\overline{Pr}$ is the macro precision, Pr_{10} the 10 % quantile of individual precisions, and N the number of classes. The results are further broken down by living (animals, plants) and non-living (fibers, aggregates, feces, etc.) categories.

	$\mathcal{L}_{MC}$ vs. $\mathcal{L}_v$	$\mathcal{L}_{MC}$			$\mathcal{L}_0$		
	$\overline{Pr}$	$\overline{Pr}$	Pr_{10}	N	$\overline{Pr}$	Pr_{10}	N
Total	0.650	0.949	0.889	280	0.738	0.288	65
Living	0.719	0.947	0.880	126	0.644	0.187	42
Non-living	0.592	0.952	0.935	146	0.862	0.730	11

For the calculation of the agreement between the MorphoCluster labeling $\mathcal{L}_{MC}$ and the initial labeling $\mathcal{L}_0$, only $\mathcal{L}_v$ was used to avoid overly optimistic results coming from data which the feature extractor was trained on. We computed the proportions of objects from all initial classes in $\mathcal{L}_v$ for every MorphoCluster category in $\mathcal{L}_{MC}$. Each category in $\mathcal{L}_{MC}$ was then assigned its predominant $\mathcal{L}_v$-class-label. The agreement was measured as the precision of a $\mathcal{L}_{MC}$ class according to the respective predominant $\mathcal{L}_v$ class.

To some degree, the labeling of MorphoCluster is consistent with the initial one (Table 3, $\mathcal{L}_{MC}$ vs. $\mathcal{L}_v$ in the first row). The agreement is, however, consistently lower than the precision of $\mathcal{L}_{MC}$ according to manual examination. This suggests that MorphoCluster categories often contain objects from multiple initial categories. The reason becomes apparent when looking at the precision of the initial labeling $\mathcal{L}_0$ (Table 3, $\mathcal{L}_0$): Macro precision over all categories is only 0.738, with 90% of the classes having a precision of only 0.288 or higher. In contrast, the precision of the MorphoCluster labeling $\mathcal{L}_{MC}$ is excellent (Table 3, $\mathcal{L}_{MC}$): Macro precision over all categories is 0.949, with 90 % of the classes having a precision of 0.889 or higher.

The categories were also divided into living and non-living categories and macro precision was calculated for each group individually. Some categories ("unknown_*", "mix_*") could not be assigned to either living or non-living and are therefore not included in these results. According to Table 3, non-living categories are sorted with higher precision than living categories in both $\mathcal{L}_{MC}$ and $\mathcal{L}_0$, so it might be easier to be self-consistent on the classification of non-living categories.

3.6. Fine-Grained Data Set Exploration

Figure 4 compares the initial labeling $\mathcal{L}_0$ to the resulting labeling $\mathcal{L}_{MC}$. Using MorphoCluster, the data set could be sorted into 280 categories in contrast to the initial 65 categories. In addition, the relative class abundances of the indicator classes $\mathcal{C}_\mathrm{i}$ were misestimated in the initial sorting. The high ranking of Poeobius in $\mathcal{L}_0$ likely originates from the high effort that was put into finding examples for this class after it had been discovered [24].

Although the largest part of the data set was sorted in the early steps (see Section 3.2), Figure 5 shows that the later steps were nevertheless required to achieve this large number of categories.

Spiking the data with labeled objects from the validation set $\mathcal{L}_v$ allowed the calculation of relative overlap between initial and new classes $\mathcal{L}_0$ and $\mathcal{L}_{MC}$. This relative overlap is depicted in the correspondence matrix Figure 7. For each $\mathcal{L}_0$ class, the corresponding $\mathcal{L}_{MC}$ classes are aligned by descending overlap in a horizontal group. A single category in the initial labeling $\mathcal{L}_0$ sometimes has a direct correspondence (red) and often decomposes into multiple categories in the MorphoCluster labeling $\mathcal{L}_{MC}$,

partly into finer subcategories (entries in the same group), partly into similar-looking but unrelated categories (entries elsewhere in the row). Conversely, $\mathcal{L}_{MC}$ classes often recruit their members from multiple $\mathcal{L}_0$ classes, indicated by columns with multiple entries. For a complete list of correspondences, see the Supplementary Materials.

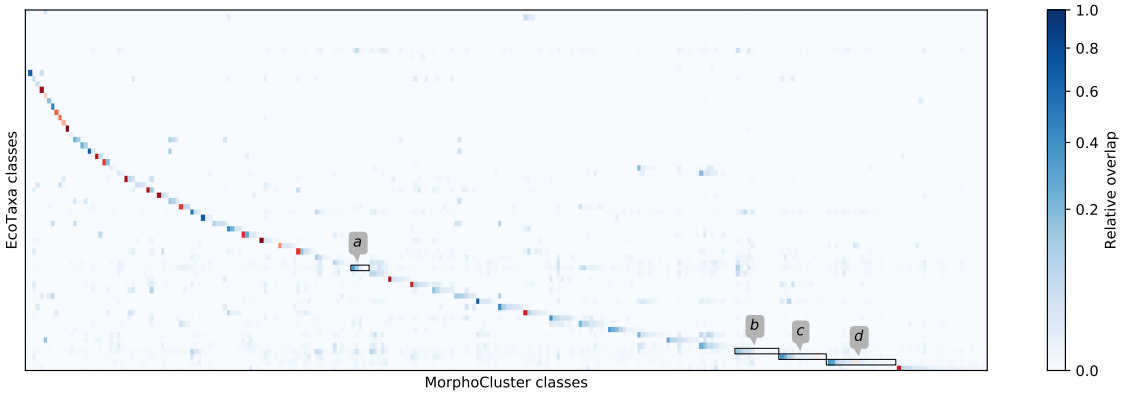

Figure 7. Correspondence of MorphoCluster and initial labels measured by relative overlap. $\mathcal{L}_0$ classes are ordered by their number of correspondences, $\mathcal{L}_{MC}$ classes are ordered by their corresponding $\mathcal{L}_0$ class. Therefore a diagonal structure emerges. Manually established direct correspondences are colored using shades of red. The first rows are $\mathcal{L}_0$ classes without a correspondence in $\mathcal{L}_{MC}$. Selected $\mathcal{L}_0$ classes are annotated for further analysis: *a* fluffy_light, *b* fluffy_dark, *c* Trichodesmium_puff, *d* Maxillopoda_Copepoda.

Subdivisions show that the images taken by the UVP5 could allow a more fine-grained sorting than previously attempted. To illustrate the high level of diversity within the classes in the initial labeling and the strong homogeneity within individual $\mathcal{L}_{MC}$ classes, the objects of four selected $\mathcal{L}_0$ classes (annotated in Figure 7) are depicted in detail in Figure 8.

Aggregations of objects from multiple original classes are signs that the initial labeling was inconsistent or that the previously applied classification scheme did not fit the cluster structure in the data. $\mathcal{L}_{MC}$ also contains many transitional classes that lie in between two clear-cut classes, as depicted in Figure 9. These contain objects that can not be assigned to either of both categories with certainty. In most cases, these seem to be decaying organisms that are losing their distinctive morphological features and seem to turn into dead matter (detritus). Some classes were annotated in $\mathcal{L}_{MC}$ that did not share any objects with an existing class in $\mathcal{L}_0$, most of them being detritus subcategories. These are not included in the correspondence matrix.

In summary, these results suggest that the subdivisions, aggregations, and transitional classes in $\mathcal{L}_{MC}$ go beyond the previous labeling $\mathcal{L}_0$ by refining it. Decision boundaries seem to align better with the data structure.

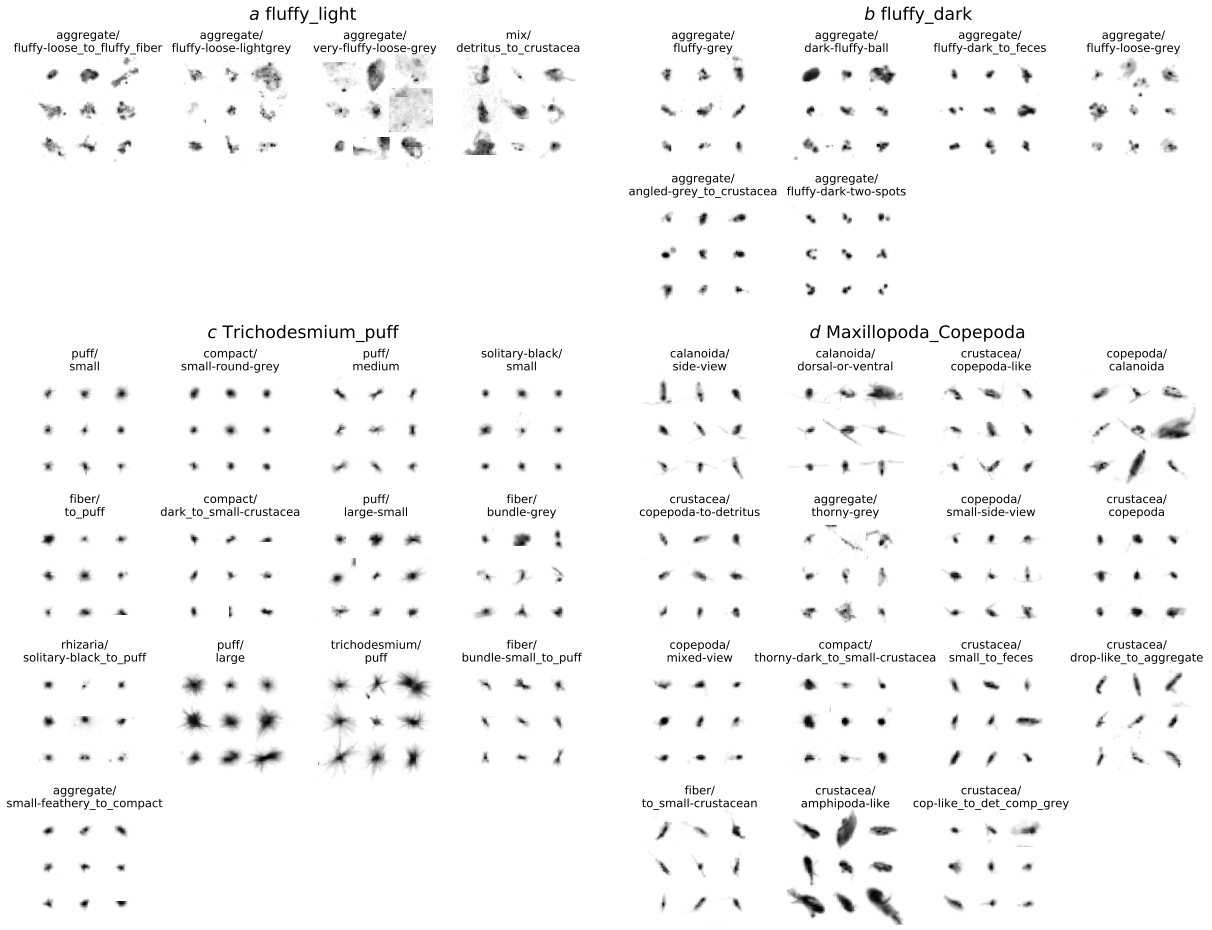

Figure 8. Four $\mathcal{L}_0$ classes (denoted in Figure 7) and their corresponding $\mathcal{L}_{MC}$ classes. These $\mathcal{L}_0$ classes are highly diverse and can be split up into finer, very homogeneous groups using MorphoCluster.

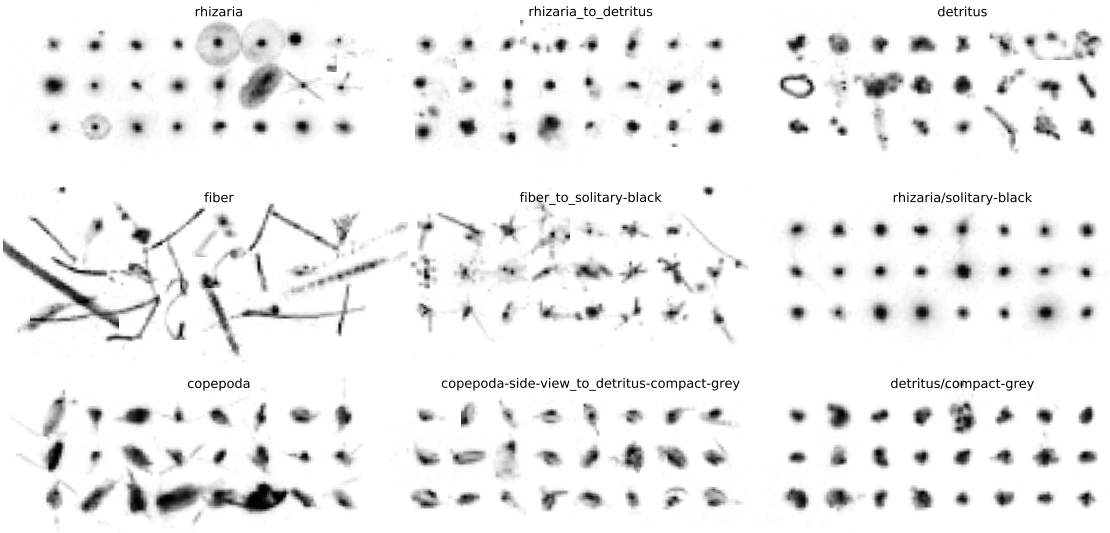

Figure 9. Some classes defined using MorphoCluster ("X_to_Y", middle column) form a transition between two clear-cut classes.

3.7. Novelty Detection

The four held-out indicator classes $\mathcal{C}_{\text{i}}$ were retrieved confidently, meaning that they were the predominant class of at least one cluster, respectively. Figure 10 shows how Veliger, T001, Flota, and Poeobius and the other classes started as very small cluster seeds and reached their final size throughout the processing of the data set.

Figure 11 illustrates the relationship between class size and time until retrieval: As intended, larger classes were found in earlier iterations and the smaller a class, the later it was found during the process. Veliger, the largest class with a very distinct shape, was retrieved early on. Poeobius, the smallest of these four, was not found until the last iteration. This trend is also reflected in the other classes.

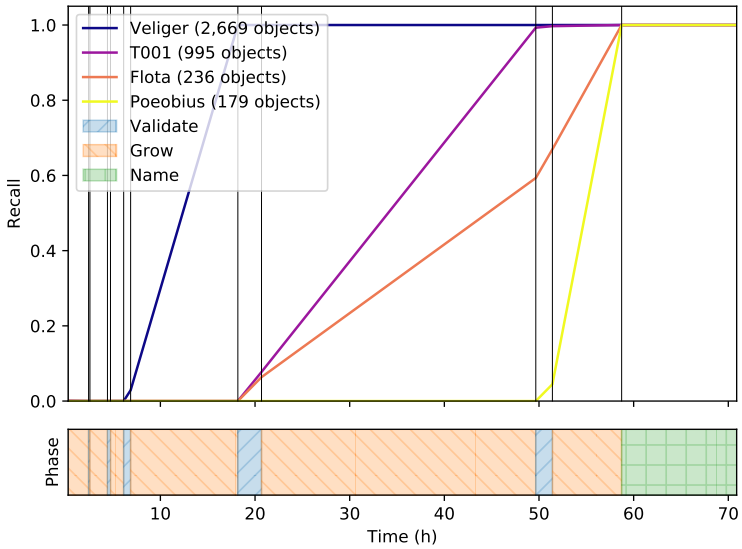

Figure 10. Recall of the indicator classes $\mathcal{C}_{\text{i}}$. The time periods are colored according to their respective phase. Veliger is found in the third iteration, T001 and Flota in the fourth, and Poeobius in the last iteration.

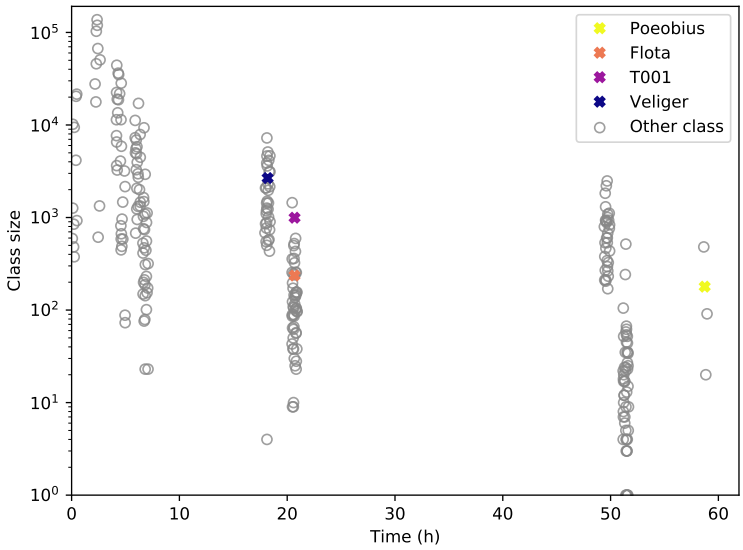

Figure 11. Discovery of classes during the process. The larger a class, the earlier its seed (with at least 5 % of the final number of objects) was found, as intended.

4. Discussion

Within plankton research—but also in many other domains—we face a flood of image data that requires interpretation [51]. While supervised machine learning approaches are generally very fast and can be very accurate, they are limited to a fixed classification scheme, so without further measures, they fail at novelty detection [52] and might perpetuate biases from the training set [53]. Humans, on the other hand, excel at fine-grained object classification and novelty detection but are limited in their annotation rate. Thus, we need to develop techniques that exploit and augment the human ability to perform object classification and novelty detection by accelerating annotation and increasing consistency [13]. Our approach augments the human abilities with machine vision performance.

MorphoCluster excels at cluster-based manual mass allocation of images into homogeneous groups, followed by hierarchical ordering in a semantic tree for easy naming of classes. By paying attention to the cluster structure of a data set, we achieve an outstanding combination of properties: MorphoCluster is at the same time fast; allows for a flexible, fine-grained, and data-driven classification; is accurate and consistent; and enables novelty detection. It is available as open-source software at https://github.com/morphocluster. We expect that the approach can be adapted to any kind of image collection where individual objects can be extracted and useful features that enable meaningful clustering can be calculated using a deep convolutional neural network (CNN).

4.1. Feature Extraction and Clustering Using Deep Learning Approaches

CNNs can generate features that are powerful and general enough to perform classification tasks using shallow classifiers like random forests, support vector machines, or logistic regression [20,28,54–56], consistently outperforming hand-crafted features [55]. Malde and Kim [36] show—by using some selected categories from a well-sorted data set—that features extracted with a siamese network can also be used to cluster images into relevant categories and allow for nearest neighbor and closest centroid classification. CNN image features also enable clustering into semantic categories on which the network was never explicitly trained [56,57]. Features learned on one task (e.g., natural objects, like birds, horses, and sheep) are also often transferable to a different task (e.g., the distinction of man-made objects, like bicycles, cars,

and trains) [28,58,59]. We therefore tested in some preliminary experiments if we could train a feature extractor with ImageNet [44] data. However, this did not produce well-defined clusters, and we fine-tuned the network with plankton images so that it could learn the characteristic appearances of different kinds of plankton. The CNN features extracted using this auxiliary training set then allowed efficient clustering and transformation into a hierarchy by agglomerative clustering.

We use the advantageous characteristics of the CNN features to provide a complete workflow to separate and classify plankton images in a real-world data set. By merging supervised and unsupervised tools with human intervention, MorphoCluster enables flexible, fine-grained mass annotation of images and detection of novel classes in a data-driven way.

4.2. MorphoCluster Is Data-Driven

Image classification is often *interest-driven*, i.e., driven by prior knowledge and expectations of the data, which is reflected in the routinely small number of classes used [26,29]. The applied classification scheme is then based on a certain research question and the annotation effort is largely influenced by this question, as well. Accordingly, some "interesting" object types are sorted with high effort, some "less interesting" types are subsumed in general classes. Furthermore, classification methods typically assume that training data and test data are independent and identically distributed [11,60]. However, this is often not the case as distribution patterns change with temporal (e.g., seasonal) and spatial dynamics [24,61] and can therefore be different for each sample [9,11]. Because classifiers are optimized for the distribution of the training sample and inherit their biases, their prediction might not represent the true data distribution of a test sample [9].

Computer-aided image classification tools (e.g., EcoTaxa [20], SQUIDLE+ [19], Pl@ntNet-Identify [49], and others [62,63]) assume that most images can be sorted into a set of classes that are defined beforehand or ad hoc. Furthermore, predictions might be skewed towards the class proportions of the training set and objects are predicted into a similar but incorrect category. Annotators might then tend to accept the prediction when they feel no strong preference (*default effect*). On the other hand, because of the *contrast effect*, an annotator might move objects, which are correctly predicted as one class (e.g., "detritus_dark") but are in some property different (e.g., lighter) than the other displayed objects surrounding them, to another (incorrect) class (e.g., "detritus_light"). Interest-driven sorting using conventional tools is therefore sometimes rather subjective and might cause a certain blindness towards the nuances in the data.

While an annotator working with MorphoCluster is still influenced by the same cognitive biases, these biases have different effects than during the usage of conventional tools. MorphoCluster allows sorting data without a preconception about the relative class abundance and takes a *data-driven*, explorative, yet manually controlled image annotation approach. Creating classes from homogeneous clusters in our view fits the granularity of the data set itself well. This approach minimizes negative subjective influences and makes structures in the data visible. The impact of the default effect is less pronounced: During cluster validation, an annotator might be tempted to just accept the proposed cluster which would impair sorting accuracy if the cluster is not clean. Due to the simplicity of the task (homogeneous/not homogeneous), however, the problem should not be as severe as with conventional sorting. The contrast effect is actually exploited to reject clusters with major impurities by showing dissimilar images side by side. If a meaningful cluster is rejected (e.g., in the second round of clustering and growing), this will slow down the process but will not affect the final result. This cluster should be proposed again in the subsequent round of clustering and growing and will still be detected. Therefore, the annotator is bothered by little remorse to reject a cluster during cluster approval. In addition, during growing, we use the contrast effect to our benefit as we oppose the cluster seeds and the images to be added to the cluster. Strong differences therefore can be easily spotted. We introduced the "turtle mode" to make the acceptance or rejection

of images at the cluster borders more flexible. Specifically, bulk acceptance might be a problem due to the default effect, whereas bulk rejection will only slow down the process. The contrast, default and recency effects should have little impact during the annotation of the cluster hierarchy in the last step of the process. The hierarchic arrangement is data-driven, and we observe that similar clusters are located in according branches. An annotator might keep branches of the automatic hierarchy (default and recency effect) until a strong contrast is found. Nuances in the data set therefore might be overlooked, but as only comparatively few clusters need to be named, the decisions are few and can be made with great care. In general, fatigue and boredom during cluster approval, growth, and naming is, in our view, much reduced in comparison to conventional sorting. The cognitive demanding classification task to allocate a name to a given object needs to be executed only in comparatively few cases, whereas the detection of new or exceptionally large clusters can be perceived as especially rewarding. As with any sorting tool, appropriateness of the sorting and annotation in MorphoCluster finally depends on the care the annotator assigns to the task. We nevertheless expect the results to be rather objective as the annotator is guided by the data structure and mostly needs to execute simple and effective tasks.

MorphoCluster is designed to work with large collections of images that are similar to the data set [40] used in the experiments. It should work particularly well for data sets in which classes contain 100 objects or more. Smaller data sets that contain fewer objects per class might not result in enough meaningful clusters. This would limit the speed and accuracy advantages of MorphoCluster over traditional methods.

4.3. MorphoCluster is Fast

Our strategy transforms time-consuming image annotation of single images into the much faster annotation of clusters.

For manual or prediction-based tools, sorting time depends on the number of objects and the number of classes [64], but details on effort and speed required to sort a data set are often not reported in the literature (e.g., References [1,7,26,44]). With overall nearly 17 k objects per hour, MorphoCluster reaches or even surpasses the sorting speed of the well-optimized supervised classification approach implemented in EcoTaxa [20] (personal communication).Depending on the size and complexity of a project, EcoTaxa allows sorting speeds between approximately 300 and 15 k objects per hour. Typically, objects are automatically classified in EcoTaxa, then the predicted images for each class are manually validated. The validation of predictions with high classification scores is commonly fast while low classification scores require extensive manual resorting. In the first iterations of the process, the sorting speed can reach 200 k objects per hour, whereas it also slows down when cluster sizes decline. Most projects in EcoTaxa use up to 90 annotation categories (personal communication), substantially less than those that emerged in MorphoCluster.It is known that it takes longer to pick a category from a larger menu [65], which indicates that the difference in sorting speed between EcoTaxa and MorphoCluster might be larger if the same granularity would be targeted.

Tian et al. [64] propose a face annotation framework that, like our approach, uses partial clustering and subsequent annotation of clusters and remaining data to quickly label large amounts of face images. In agreement with our results, they observe that clustering can substantially reduce the annotation workload because each user interaction affects a large number of individual objects and partial clustering groups images into meaningful and homogeneous clusters. They provide a rough estimate that their approach is 5 times as fast as conventional sorting.

To increase the overall speed of MorphoCluster, we optimized each individual step. During validation, clusters of similar objects are accepted as a whole, which drastically reduces the number of entities that require annotation in further steps. In the cluster growth step, binary search enables the user to quickly find the border of a cluster. Thus, adding any number of objects to a cluster requires only a small fraction of

the time required to annotate these objects individually. When the border of the cluster is reached, the user can also delete or accept single images, which activates a "turtle mode", disables binary search, and forces the user to conduct single image approval. The suitability of our cluster growth strategy is clearly confirmed by the high sorting accuracy. We investigated if the growth of the clusters could be optimized by accounting for non-spherical clusters but noticed no improvement. The hierarchical arrangement of similar clusters facilitates their naming. The same time to identify a single object in traditional approaches is spent to identify many objects, sometimes even thousands, which in turn leads to less time pressure in assigning proper names. MorphoCluster's high sorting rate is a result of the fact that simple user decisions in each step affect a large number of objects and as partitioning and naming are different steps, more effort can be put into a precise and fine-grained classification.

4.4. Flexible and Fine-Grained Classification

We developed a strategy for cluster retrieval that guarantees that large clusters are retrieved at the beginning of the process and small clusters only at the end. Preliminary experiments showed that settings that allow for small cluster sizes immediately lead to an over-separation of some classes and fragmented larger classes into many more or less indistinguishable clusters. These mostly consisted of some detritus categories. Merging and/or naming of these clusters would have become very time-consuming, and, in very many cases, we would have given identical names for these clusters. Our strategy to first retrieve large clusters improved the situation, but still, some clusters were retrieved that were subsequently merged during the naming step. Our hierarchical naming tool nevertheless makes these decisions less subjective, as it contrasts similar clusters. In the end, the decision of whether or not two groups of images show the same category is made by the user. Further research is necessary to optimize the strategy of cluster retrieval and growth as an optimal path through the data should exist that could reduce the need to merge clusters. In comparison to the original data set which was sorted into 65 classes, we retrieved 280 classes and, in general, a more fine-grained sorting, which might reveal new insights. Detritus, for example, was previously often sorted into less than ten classes, although there can be strong differences in shape and size which are likely related to its biogeochemical properties. A nuanced isolation of these shapes makes it easier to find such properties in data.

4.5. Detection of Novel Classes

As data sets increase in size, former outliers may grow into new categories: Consider a data set containing 1 k images. It might contain a single image of *Poeobius* sp., a species found in very low numbers throughout the whole Atlantic Ocean which under certain conditions proliferates strongly [24]. Sorting the whole data set by hand, an expert would create a class "Poeobius" because of their knowledge of its appearance. Another possibility is that these images are subsumed under a more general category during interest-driven sorting. Using our tool, we would not find this single image because MorphoCluster is geared towards finding groups of similar objects. If we now collect more images from the same source and grow this image data set, the number of *Poeobius* sp. images might grow proportionally, and we should find 1000 images in a 1 million image data set. Our experiment indicates that these images would then be found as a cluster that can be identified and named.

MorphoCluster's data-driven approach allowed the reliable detection of the held-out indicator classes (Veliger, T001, Flota, and Poeobius), and we predict that, by applying the natural decision boundaries dictated by the density structure of the data, it is equally likely to find other novel classes. Several of the transitional classes we identified (like depicted in Figure 9) could also be considered novel classes.

Therefore, we deem MorphoCluster well-suited to search the numerous sources of constantly growing marine imaging data for previously undocumented categories.

4.6. Accuracy and Consistency

The accuracy of human sorting mainly depends on the operator. Within plankton research, experts can reach a panel consistency of up to 95 % for small numbers of categories [66]. Using MorphoCluster, most of the resulting 280 classes were sorted with very high consistency in the same range (see Section 3.5), and similar-looking objects share the same annotation. This can be explained by the fact that the process starts with very homogeneous clusters of objects that stay homogeneous even after growing. As discussed previously, a user is less affected by cognitive biases when using MorphoCluster than when using conventional methods. This way, the homogeneity of clusters is carried through to the end of the whole process.

In manual or prediction-based sorting tools, objects are typically sorted individually, and the context of similar objects is not available. Conversely, clustering-based approaches provide this kind of context by constructing homogeneous groups of objects [64], a huge advantage that is also shared by MorphoCluster.

4.7. Possible Improvements

4.7.1. Feature Learning and Clustering

Feature learning and clustering are sequential steps in the current implementation, and we rely on an initial training set to train the feature extractor. Recent works on unsupervised learning of deep image descriptors combine feature learning and clustering and do not require any labels [67–71]. These unsupervised feature learning methods could be investigated to reduce the reliance on labeled data.

A small number of objects was ultimately left untreated (residual objects) and a handful of known small classes was not retrieved. An adjustment of the feature extractor or the use of a different clustering algorithm could maybe help to mitigate this problem. Still, it is obvious that classes with a very small number of objects (*low-shot* or *one-shot classes* [72,73]) cannot be retrieved by clustering, although human knowledge indicates their presence. To facilitate their retrieval, *spiking* the unlabeled data with labeled objects could increase their density in the feature space, and low-shot learning techniques [25] could be employed to identify them prior to clustering, but this does not work for unknown classes. Therefore, methods of novelty detection [74] (e.g., Reference [75]) should be investigated.

One of the classes not retrieved using MorphoCluster, *Pyrosoma* sp. (named *Pyrosomatida_Pyrosoma*), exhibits some very large images. Large variations in image size are a general problem for convolutional neural networks. To be able to process these images, we scale the images down to the input size of the network. Unfortunately, this can weaken and sometimes even remove their distinctive features. A possible future research direction is therefore the exploration of attention mechanisms [76–78] that allow the network to focus on specific image regions and view them in full resolution. Some distinguishing features of an object might not be represented in the features learned by the deep feature extractor, either because of insufficient sensor resolution or because they are of a different modality (e.g., genetic, environmental, etc.). The introduction of other morphometric [79] (e.g., size or texture features) and environmental [27] (e.g., depth or temperature) information into the deep learning image recognition could therefore be a viable option to improve clustering and reduce the number of residual objects.

The HDBSCAN* algorithm that was used in this work has a runtime super-linear in the number of objects and the number of dimensions at best [47]. Speeding up the clustering approach could enable the execution of the clustering, growing and approval procedure in single rounds so that only the largest and best-defined cluster is extracted in every iteration, thereby enabling a more interactive user experience. This would especially be useful at the beginning of the procedure as it would yield a more optimal path through the data. The main competitor is k-means with a best-case runtime linear in the number of objects and the number of dimensions [47], which becomes quite an advantage with large data volumes.

However, k-means is a *partitioning* clustering algorithm, while HDBSCAN* does not necessarily assign a cluster for all points, and the question remains on how it can be adapted to the requirements of the MorphoCluster framework.

4.7.2. Hierarchical Naming

Although the morphology of an organism is in part determined by its genes, this relationship is very complex. As an example, larvae and adults can look completely different, although they share the same set of genes [80]. The class hierarchy that we used as a starting point in the naming step was generated from the list of clusters using agglomerative clustering, which successively contracts similar clusters [50] (p. 73).

The calculated cluster hierarchy coincides only in few cases with the known phylogenetic tree of life because the phylogenetic tree is derived not only from images but also, for example, from genetic, ontogenetic, and microscopic analysis. We chose average linkage (UPGMA) clustering as a robust default method, and it should be investigated if alternatives (e.g., WPGMA [50] (p. 79)) lead to a closer match between precomputed hierarchy and manually tuned end result.

The final sorting emerges from the interaction of the taxonomic knowledge of the annotator and the data-driven arrangement of the data set. This interaction could be further facilitated by including an extensible reference taxonomy in the application, spiking the input data with existing labeled data to match the emerging clusters to known classes (like we did in the evaluation of our approach), or providing some sort of vocabulary to avoid the occasional naming inconsistencies introduced by the free-form input. It also seems useful to use the clusters from a first MorphoCluster run as seeds in future runs, which only need to be grown using the new data.

4.7.3. Division of Labor

MorphoCluster could enable a unique distribution of efforts between users with different expertise to accelerate sorting and make better use of available human resources. The separation of sorting and naming could allow entrusting the relatively simple task of validating and growing homogeneous clusters to less experienced staff, while professional taxonomists, whose time is a precious resource [13], could focus on the more complex but less time-consuming task of cluster identification.

Multi-user approaches during which several users work on different clusters of a given data set should also be possible. The high throughput of MorphoCluster could even enable the replication of the entire process by different experts or teams, which should increase the overall annotation quality even further.

5. Conclusions

With MorphoCluster, we present a novel approach to image annotation that does not require the user to take a look at *every single image*. Rather, similar images are automatically aggregated in clusters, which are checked for consistency. These clusters are thereafter grown and named de novo, avoiding biases of a given prediction or sorting scheme. We succeeded to shift the unit of labor during the sorting process from individual images to often very large clusters. The development of useful CNN features was, in our view, critical for this success. The result of our efforts is a simple and fast manual annotation tool, which yields a consistent and fine-grained sorting. The sorting effort with MorphoCluster scales primarily with the number of classes of a given data set, while with other tools, the effort scales with the number of images. We argue that our approach is less biased by contrast, default and recency effects and avoids pitfalls of interest-driven sorting. The primary use case for our tool is the *rapid annotation of images* to acquire huge volumes of labeled data for further data analysis or to initialize a training set. Importantly, it also enables novelty detection and facilitates the data-driven creation of possibly meaningful subcategories. By using

MorphoCluster, we can shift away from accidental discoveries and a lot of manual labor to a systematic and fast strategy for surveying the ocean. It will hopefully help to stem the flood of plankton image data that we expect and may be just as useful for annotating other image data sets.

Supplementary Materials: The following are available online at http://www.mdpi.com/1424-8220/20/11/3060/s1, Label supercategories (all labels contained in the colored supercategories in Figure 6) and Corresponding labels (corresponding names in $\mathcal{L}_0$ and $\mathcal{L}_{MC}$ as in Figure 7).

Author Contributions: S.-M.S.: Concept, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing—original draft; R.K. (Rainer Kiko): Concept, Testing, Funding acquisition, Resources, Supervision, Writing—original draft; R.K. (Reinhard Koch): Funding acquisition, Resources, Supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Cluster of Excellence 80 "Future Ocean" (CP1733). "Future Ocean" is funded within the framework of the Excellence Initiative by the Deutsche Forschungsgemeinschaft (DFG) on behalf of the German federal and state governments. R. Kiko was furthermore supported by the SFB 754 "Climate-Biogeochemistry Interactions in the Tropical Ocean" (www.sfb754.de, grant no. 27542298 of the German Science Foundation DFG) and via a "Make Our Planet Great Again" grant of the French National Research Agency within the "Programme d'Investissements d'Avenir"; reference ANR-19-MPGA-0012.

Acknowledgments: We thank Svenja Christiansen, Jean-Olivier Irisson and Marc Picheral for insightful discussions on plankton image sorting. Jean-Olivier Irisson and Marc Picheral furthermore provided background information about EcoTaxa.

Conflicts of Interest: The authors declare no conflict of interest.

Data and Code Availability: We released the source code for the MorphoCluster Application under the GNU General Public License (GPL) at https://github.com/morphocluster. The initial training set $\mathcal{L}_0$ and the new labeling $\mathcal{L}_{MC}$ are available at https://www.seanoe.org/data/00618/73002/.

References

1. Gorsky, G.; Ohman, M.D.; Picheral, M.; Gasparini, S.; Stemmann, L.; Romagnan, J.B.; Cawood, A.; Pesant, S.; Garcia-Comas, C.; Prejger, F. Digital zooplankton image analysis using the ZooScan integrated system. *J. Plankton Res.* **2010**, *32*, 285–303. [CrossRef]
2. Picheral, M.; Guidi, L.; Stemmann, L.; Karl, D.M.; Iddaoud, G.; Gorsky, G. The Underwater Vision Profiler 5: An advanced instrument for high spatial resolution studies of particle size spectra and zooplankton. *Limnol. Oceanogr. Methods* **2010**, *8*, 462–473. [CrossRef]
3. Cowen, R.K.; Guigand, C.M. In situ Ichthyoplankton Imaging System (ISIIS): System design and preliminary results. *Limnol. Oceanogr. Methods* **2008**, *6*, 126–132. [CrossRef]
4. Olson, R.J.; Sosik, H.M. A submersible imaging-in-flow instrument to analyze nano-and microplankton: Imaging FlowCytobot. *Limnol. Oceanogr. Methods* **2007**, *5*, 195–203. [CrossRef]
5. Sosik, H.M.; Olson, R.J. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. *Limnol. Oceanogr. Methods* **2007**, *5*, 204–216. [CrossRef]
6. Grosjean, P.; Denis, K.; Wacquet, G. Zoo/PhytoImage. Available online: https://www.sciviews.org/software/zooimage/ (accessed on 20 May 2020).
7. Orenstein, E.C.; Beijbom, O.; Peacock, E.E.; Sosik, H.M. WHOI-Plankton—A Large Scale Fine Grained Visual Recognition Benchmark Dataset for Plankton Classification. *arXiv* **2015**, arxiv:1510.00745.
8. Elineau, A.; Desnos, C.; Jalabert, L.; Olivier, M.; Romagnan, J.B.; Brandao, M.; Lombard, F.; Llopis, N.; Courboulès, J.; Caray-Counil, L.; et al. ZooScanNet: Plankton images captured with the ZooScan. *SEANOE* **2018**. [CrossRef]
9. González, P.; Álvarez, E.; Díez, J.; López-Urrutia, Á.; del Coz, J.J. Validation methods for plankton image classification systems. *Limnol. Oceanogr. Methods* **2017**, *15*, 221–237. [CrossRef]
10. Malde, K.; Handegard, N.O.; Salberg, A.B. Machine intelligence and the data-driven future of marine science. *ICES J. Mar. Sci.* **2019**. [CrossRef]

11. González, P.; Castaño, A.; Chawla, N.V.; Coz, J.J.D. A Review on Quantification Learning. *ACM Comput. Surv.* **2017**, *50*, 1–40. [CrossRef]
12. Benfield, M.; Grosjean, P.; Culverhouse, P.F.; Irigolen, X.; Sieracki, M.; Lopez-Urrutia, A.; Dam, H.; Hu, Q.; Davis, C.; Hanson, A.; et al. RAPID: Research on Automated Plankton Identification. *Oceanography* **2007**, *20*, 172–187. [CrossRef]
13. MacLeod, N.; Benfield, M.; Culverhouse, P. Time to automate identification. *Nature* **2010**, *467*, 154–155. [CrossRef] [PubMed]
14. Gomes-Pereira, J.N.; Auger, V.; Beisiegel, K.; Benjamin, R.; Bergmann, M.; Bowden, D.; Buhl-Mortensen, P.; De Leo, F.C.; Dionísio, G.; Durden, J.M.; et al. Current and future trends in marine image annotation software. *Prog. Oceanogr.* **2016**, *149*, 106–120. [CrossRef]
15. Trygonis, V.; Sini, M. PhotoQuad: A dedicated seabed image processing software, and a comparative error analysis of four photoquadrat methods. *J. Exp. Mar. Biol. Ecol.* **2012**, *424–425*, 99–108. [CrossRef]
16. Schlining, B.; Stout, N.J. *MBARI's Video Annotation and Reference System*. In Proceedings of the OCEANS 2006, Boston, MA, USA, 18–21 September 2006; pp. 1–5. [CrossRef]
17. Teixidó, N.; Albajes-Eizagirre, A.; Bolbo, D.; Le Hir, E.; Demestre, M.; Garrabou, J.; Guigues, L.; Gili, J.M.; Piera, J.; Prelot, T.; et al. Hierarchical segmentation-based software for cover classification analyses of seabed images (Seascape). *Mar. Ecol. Prog. Ser.* **2011**, *431*, 45–53. [CrossRef]
18. Langenkämper, D.; Zurowietz, M.; Schoening, T.; Nattkemper, T.W. BIIGLE 2.0—Browsing and Annotating Large Marine Image Collections. *Front. Mar. Sci. Spec.* **2017**. [CrossRef]
19. SQUIDLE+—A Tool for Managing, Exploring & Annotating Images, Video & Large-Scale Mosaics. Available online: https://squidle.org/ (accessed on 20 February 2020).
20. Picheral, M.; Colin, S.; Irisson, J.O. EcoTaxa—A Tool for the Taxonomic Classification of Images. 2017. Available online: http://ecotaxa.obs-vlfr.fr/ (accessed on 20 February 2020).
21. Gasparini, S.; Antajan, E. Plankton Identifier: A Software for Automatic Recognition of Planktonic Organisms. Available online: http://www.obs-vlfr.fr/~gaspari/Plankton_Identifier/index.php (accessed on 20 February 2020).
22. Bell, J.L.; Hopcroft, R.R. Assessment of ZooImage as a tool for the classification of zooplankton. *J. Plankton Res.* **2008**, *30*, 1351–1367. [CrossRef]
23. Biard, T.; Stemmann, L.; Picheral, M.; Mayot, N.; Vandromme, P.; Hauss, H.; Gorsky, G.; Guidi, L.; Kiko, R.; Not, F. In situ imaging reveals the biomass of giant protists in the global ocean. *Nature* **2016**. [CrossRef]
24. Christiansen, S.; Hoving, H.J.; Schütte, F.; Hauss, H.; Karstensen, J.; Körtzinger, A.; Schröder, S.M.; Stemmann, L.; Christiansen, B.; Picheral, M.; et al. Particulate matter flux interception in oceanic mesoscale eddies by the polychaete Poeobius sp. *Limnol. Oceanogr.* **2018**, *63*, 2093–2109. [CrossRef]
25. Schröder, S.-M.; Kiko, R.; Irisson, J.-O.; Koch, R. Low-Shot Learning of Plankton Categories. In Proceedings of the 40th German Conference on Pattern Recognition (GCPR), Stuttgart, Germany, 9–12 October 2018; pp. 391–404. [CrossRef]
26. Bochinski, E.; Bacha, G.; Eiselein, V.; Walles, T.J.W.; Nejstgaard, J.C.; Sikora, T. Deep Active Learning for In Situ Plankton Classification. In *Pattern Recognition and Information Forensics*; Zhang, Z., Suter, D., Tian, Y., Branzan Albu, A., Sidère, N., Jair Escalante, H., Eds.; Springer: Cham, Switzerland, 2019; pp. 5–15. [CrossRef]
27. Ellen, J.S.; Graff, C.A.; Ohman, M.D. Improving plankton image classification using context metadata. *Limnol. Oceanogr. Methods* **2019**, *17*, 439–461. [CrossRef]
28. Orenstein, E.C.; Beijbom, O. Transfer Learning and Deep Feature Extraction for Planktonic Image Data Sets. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 27–31 March 2017; pp. 1082–1088. [CrossRef]
29. Ellen, J.; Li, H.; Ohman, M.D. Quantifying California current plankton samples with efficient machine learning techniques. In Proceedings of the OCEANS 2015—MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–9. [CrossRef]
30. Vapnik, V.N. *Statistical Learning Theory*; Adaptive and Learning Systems for Signal Processing, Communications, and Control; Wiley: New York, NY, USA, 1998.

31. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
32. Culverhouse, P.F.; Simpson, R.; Ellis, R.; Lindley, J.; Williams, R.; Parisini, T.; Reguera, B.; Bravo, I.; Zoppoli, R.; Earnshaw, G.; et al. Automatic classification of field-collected dinoflagellates by artificial neural network. *Mar. Ecol. Prog. Ser.* **1996**, *139*, 281–287. [CrossRef]
33. Blaschko, M.B.; Holness, G.; Mattar, M.A.; Lisin, D.; Utgoff, P.E.; Hanson, A.R.; Schultz, H.; Riseman, E.M. Automatic In Situ Identification of Plankton. In Proceedings of the 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05), Breckenridge, CO, USA, 5–7 January 2005; pp. 79–86. [CrossRef]
34. Lee, H.; Park, M.; Kim, J. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3713–3717. [CrossRef]
35. Graham, B.; van der Maaten, L. Submanifold Sparse Convolutional Networks. *arXiv*, **2017**, arxiv:1706.01307.
36. Malde, K.; Kim, H. Beyond image classification: Zooplankton identification with deep vector space embeddings. *arXiv* **2019**, arxiv:1909.11380v1.
37. Culverhouse, P.F. Natural Object Categorization: Man versus Machine. In *Automated Taxon Identification in Systematics: Theory, Approaches and Applications*; MacLeod, N., Ed.; CRC Press: Boca Raton, FL, USA, 2007; pp. 25–46. [CrossRef]
38. Hoving, H.J.; Christiansen, S.; Fabrizius, E.; Hauss, H.; Kiko, R.; Linke, P.; Neitzel, P.; Piatkowski, U.; Körtzinger, A. The Pelagic In situ Observation System (PELAGIOS) to reveal biodiversity, behavior, and ecology of elusive oceanic fauna. *Ocean Sci.* **2019**, *15*, 1327–1340. [CrossRef]
39. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 1717–1724. [CrossRef]
40. Kiko, R.; Schröder, S.M. UVP5 data sorted with EcoTaxa and MorphoCluster. *SEANOE* **2020**. [CrossRef]
41. Costello, M.J.; Bouchet, P.; Boxshall, G.; Fauchald, K.; Gordon, D.; Hoeksema, B.W.; Poore, G.C.; van Soest, R.W.; Stöhr, S.; Walter, T.C.; et al. Global Coordination and Standardisation in Marine Biodiversity through the World Register of Marine Species (WoRMS) and Related Databases. *PLoS ONE* **2013**, *8*. [CrossRef]
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]
43. Canziani, A.; Paszke, A.; Culurciello, E. An Analysis of Deep Neural Network Models for Practical Applications. *arXiv* **2016**, arxiv:1605.07678.
44. Deng, J.; Dong, W.; Socher, R.; Li, L-J.; Li, K.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
45. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *arXiv* **2014**, arxiv:1405.3531.
46. Paszke, A.; Chanan, G.; Lin, Z.; Gross, S.; Yang, E.; Antiga, L.; Devito, Z. Automatic differentiation in PyTorch. *Adv. Neural Inf. Process. Syst. (NIPS)* **2017**, *30*, 1–4.
47. McInnes, L.; Healy, J. Accelerated Hierarchical Density Based Clustering. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; Volume 2005, pp. 33–42. [CrossRef]
48. Campello, R.J.G.B.; Moulavi, D.; Zimek, A.; Sander, J. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Trans. Knowl. Discov. Data* **2015**, *10*, 1–51. [CrossRef]
49. Joly, A.; Goëau, H.; Bonnet, P.; Bakić, V.; Barbe, J.; Selmi, S.; Yahiaoui, I.; Carré, J.; Mouysset, E.; Molino, J.F.; et al. Interactive plant identification based on social image data. *Ecol. Informatics* **2014**, *23*, 22–34. [CrossRef]
50. Everitt, B.S.; Landau, S.; Leese, M.; Stahl, D. *Cluster Analysis*; John Wiley & Sons: New York, NY, USA, 2011.
51. Lombard, F.; Boss, E.; Waite, A.M.; Uitz, J.; Stemmann, L.; Sosik, H.M.; Schulz, J.; Romagnan, J.B.; Picheral, M.; Pearlman, J.; et al. Globally consistent quantitative observations of planktonic ecosystems. *Front. Mar. Sci.* **2019**, *6*, 196. [CrossRef]

52. Shu, L.; Xu, H.; Liu, B. Unseen Class Discovery in Open-world Classification. *arXiv* **2018**, arxiv:1801.05609.
53. Kotsiantis, S.; Kanellopoulos, D.; Pintelas, P. Handling imbalanced datasets: A review. *Science* **2006**, *30*, 25–36. [CrossRef]
54. van Ginneken, B.; Setio, A.A.A.; Jacobs, C.; Ciompi, F. Off-the-shelf convolutional neural network features for pulmonary nodule detection in computed tomography scans. In Proceedings of the 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), New York, NY, USA, 16–19 April 2015; pp. 286–289. [CrossRef]
55. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Columbus, OH, USA, 24–27 June 2014; pp. 512–519. [CrossRef]
56. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In Proceedings of the ICML'14: Proceedings of the 31st International Conference on International Conference on Machine Learning, Bejing, China, 22–24 June 2014; pp. 647–655.
57. Guérin, J.; Gibaru, O.; Thiery, S.; Nyiri, E. CNN Features are also Great at Unsupervised Classification. *arXiv* **2017**, arxiv:1707.01700.
58. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arxiv:1409.1556.
59. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *4*, 3320–3328.
60. Forman, G. Quantifying counts and costs via classification. *Data Min. Knowl. Discov.* **2008**, *17*, 164–206. [CrossRef]
61. Mackas, D.; Greve, W.; Edwards, M.; Chiba, S.; Tadokoro, K.; Eloire, D.; Mazzocchi, M.; Batten, S.; Richardson, A.; Johnson, C.; et al. Changing zooplankton seasonality in a changing ocean: Comparing time series of zooplankton phenology. *Prog. Oceanogr.* **2012**, *97–100*, 31–62. [CrossRef]
62. Wäldchen, J.; Mäder, P. Machine learning for image based species identification. *Methods Ecol. Evol.* **2018**, *9*, 2216–2225. [CrossRef]
63. Wäldchen, J.; Mäder, P. *Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review*; Springer: Amsterdam, The Netherlands, 2018; Volume 25, pp. 507–543. [CrossRef]
64. Tian, Y.; Liu, W.; Xiao, R.; Wen, F.; Tang, X. A Face Annotation Framework with Partial Clustering and Interactive Labeling. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8. [CrossRef]
65. Fasolo, B.; Carmeci, F.A.; Misuraca, R. The effect of choice complexity on perception of time spent choosing: When choice takes longer but feels shorter. *Psychol. Mark.* **2009**, *26*, 213–228. [CrossRef]
66. Culverhouse, P.; Williams, R.; Reguera, B.; Herry, V.; González-Gil, S. Do experts make mistakes? A comparison of human and machine identification of dinoflagellates. *Mar. Ecol. Prog. Ser.* **2003**, *247*, 17–25. [CrossRef]
67. Aljalbout, E.; Golkov, V.; Siddiqui, Y.; Strobel, M.; Cremers, D. Clustering with Deep Learning: Taxonomy and New Methods. *arXiv* **2018**, arxiv:1801.07648.
68. Haeusser, P.; Plapp, J.; Golkov, V.; Aljalbout, E.; Cremers, D. Associative Deep Clustering: Training a Classification Network with No Labels. In Proceedings of the 40th German Conference on Pattern Recognition (GCPR), Stuttgart, Germany, 9–12 October 2018; pp. 18–32. [CrossRef]
69. Caron, M.; Bojanowski, P.; Joulin, A.; Douze, M. Deep Clustering for Unsupervised Learning of Visual Features. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 139–156. [CrossRef]
70. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; Volume 1, pp. 740–749.
71. Yang, J.; Parikh, D.; Batra, D. Joint Unsupervised Learning of Deep Representations and Image Clusters. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5147–5156. [CrossRef]
72. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching Networks for One Shot Learning. *arXiv* **2016**, arxiv:1606.04080.

73. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 3, pp. 1856–1868.
74. Pimentel, M.A.F.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. *Signal Process.* **2014**, *99*, 215–249. [CrossRef]
75. Bodesheim, P.; Freytag, A.; Rodner, E.; Denzler, J. Local Novelty Detection in Multi-class Recognition Problems. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2015; pp. 813–820. [CrossRef]
76. Sun, X.; Xv, H.; Dong, J.; Zhou, H.; Chen, C.; Li, Q. Few-shot Learning for Domain-specific Fine-grained Image Classification. *IEEE Trans. Ind. Electron.* **2020**, *46*. [CrossRef]
77. Sun, G.; Cholakkal, H.; Khan, S.; Khan, F.S.; Shao, L. Fine-grained Recognition: Accounting for Subtle Differences between Similar Classes. *arXiv* **2019**, arxiv:1912.06842.
78. Zheng, H.; Fu, J.; Zha, Z.J.; Luo, J. Looking for the Devil in the Details: Learning Trilinear Attention Sampling Network for Fine-Grained Image Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 5007–5016. [CrossRef]
79. Campbell, R.W.; Roberts, P.L.; Jaffe, J. The Prince William Sound Plankton Camera: A profiling in situ observatory of plankton and particulates. *ICES J. Mar. Sci.* **2020**. [CrossRef]
80. Kiko, R.; Kramer, M.; Spindler, M.; Wägele, H. Tergipes antarcticus (Gastropoda, Nudibranchia): Distribution, life cycle, morphology, anatomy and adaptation of the first mollusc known to live in Antarctic sea ice. *Polar Biol.* **2008**, *31*, 1383–1395. [CrossRef]