Title

Quantifying California current plankton samples with efficient machine learning techniques

ISBN

9780933957435

Authors

Ellen, Jeffrey

Li, Hongyu

Ohman, Mark D

Publication Date

2015

DOI

10.23919/oceans.2015.7404607

Peer reviewed

Quantifying California Current Plankton Samples with Efficient Machine Learning Techniques

Jeffrey Ellen, Hongyu Li
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404
Email: jellen@ucsd.edu, holi@ucsd.edu

Mark D. Ohman
Scripps Institution of Oceanography
University of California, San Diego
La Jolla, CA 92093-0218
Email: mohman@ucsd.edu

***Abstract*— This paper improves on the accuracy of other published machine learning results for quantifying plankton samples. The contributions of this work are: (1) Clarifying the number of expertly labeled images required for machine learning results. (2) Providing guidance as to what algorithms provide the best performance, and how to tune them. (3) Leveraging an ensemble of models to achieve recall rates beyond any single algorithm. (4) Investigating the applicability of abstaining. (5) Using size fractionation to learn more efficiently. (6) Analysis of efficacy of simple geometric features for plankton identification.**

Keywords—machine learning; image analysis; zooplankton; Zooscan

I. Introduction

Quantifying plankton is important, requires a high level of taxonomic skill, and is expensive. Automation of plankton sample enumeration can enable higher throughput, more efficient processing, and improved scientific understanding., Specific applications of interest include understanding plankton spatial distributions, parameterizing oceanographic models, and investigations of population ecology. In this paper, we address methods to improve automatic classification of images from preserved plankton samples.

II. Machine Learning Experimentation

A. Data Set Description

The California Cooperative Oceanic Fisheries Investigations (CalCOFI) is a field program that has been sampling the ocean, including plankton, since 1949 [1]. The CalCOFI plankton samples are collected at sea according to a standardized bongo net protocol [2] and immediately preserved. Substantial portions of the preserved CalCOFI samples recently collected in conjunction with the California Current Ecosystem Long Term Ecological Research site have been scanned with ZooScan [3]. The resulting grayscale images are very accurately controlled in terms of contrast, noise, and other variations (Fig. 1).

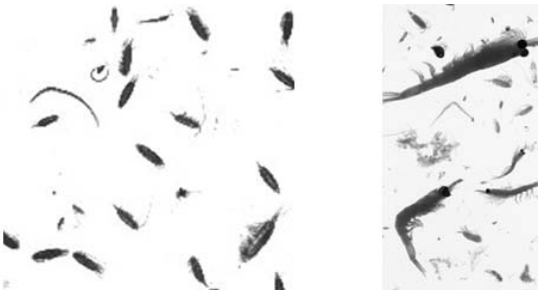

Fig. 1. Parts of two different scanned images of plankton samples, illustrating variety of ROI shapes and sizes, as well as the relative ease of ROI segmentation.

Because of the controlled conditions, the Regions of Interest (ROIs) are readily segmented from the larger image, which contains 1,000-2,000 ROIs. Figure 2 (upper row) shows two examples of animals scanned by the ZooScan, with photographs of similar animals for comparison (lower row).

Fig. 2. Left: *Nyctiphanes simplex*, a euphausiid common off the California coast. Right: a chaetognath. Both are relatively large for CalCOFI plankton: the scale bar in all 4 images is 1mm. Both have substructures and opacity differences that are preserved in ZooScan images. Photos from SIO Pelagic Invertebrates Collection [4].

Not all of the classes are so easily recognized. Figure 3 shows three more categories of varying size, shape, and

Contribution from the National Science Foundation–supported California Current Ecosystem Long Term Ecological Research site. Plankton sample analysis supported by NSF grants to M.D. Ohman, and by the SIO Pelagic Invertebrates Collection.

978-0-933957-43-5 ©2015 MTS

contrast. Some plankters are inherently more fragile, with gelatinous parts or thin appendages that are frequently damaged by net collection. Transparency is more variable in preserved samples than in live ones. Less rigid animals also have a less consistent posture and orientation. Some classes have a wide variety of sizes, and the smallest plankters have a lack of detail due to the limits of the scanning resolution. These identifications can be challenging for a human.

These challenges are not unique to plankton imaging, but are different from mainstream image processing/classification tasks, such as the ImageNet competition.

Fig. 3. Left: bryozoan larvae, of relatively uniform size and shape. Middle: a small chain of the salp *Pegea socia*, a gelatenous pelagic tunicate whose ZooScan samples exhibit some variation of scale and irregular shapes. Right: copepods, which have an even larger size range and variation within the image, despite being relatively rigid compared to the gelatinous tunicates. All ZooScan images have a scale bar of 1mm, the bryozoan larvae photo has a scale bar of 0.2mm, and the tunicate photo has a scale bar of 5mm. Photos from SIO Pelagic Invertebrates Collection [4].

The data used in this paper consists of 725,516 individual ROIs taken from samples collected during 46 different ocean transects from July 2005 to July 2012. The transects are line 80 and line 90 in the CalCOFI grid (Fig. 4), and samples are taken quarterly. Most ROIs contain a single entity, and are labeled with one of 24 categories of organisms such as 'siphonophore' or 'calanoid copepod'. There are also categories for 'detritus', 'multiples', and 'others'. The splits are functional rather than biological or genetic. A complete list is provided in Appendix A.

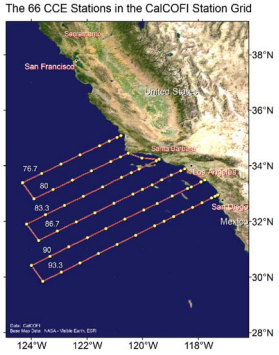

Fig. 4. The CalCOFI grid has been sampled for 67 years. Samples from line 80 and line 90, from July 2005 to July 2012 were used in this data set.

The data described in this paper will shortly be made available through the CalCOFI DataZoo website (http://oceaninformatics.ucsd.edu/datazoo/)

B. *Machine Learning Features*

The ZooScan software can generate low level geometric features and grayscale features for the purposes of biological object identification [5]. ZooScan has been used to count the abundance of both zooplankton [3] as well as fish eggs [6].

We used a subset of 51 of these features in the experiments described in this paper. Each feature is computed on the pixels within the ROI only, not the bounding rectangle. Features used include 19 size/shape measurements, such as area, circularity, major/minor axis length, feret diameter, and some ratios of these values. Also included are 17 grayscale distribution measurements, such as the min, max, mean, standard deviation, quartiles, skew, and cumulative histogram slope. The remaining features are positional, such as the centroid location, or more derived, such as the fractal dimension or the symmetry. Complete descriptions of the referenced features are on the ZooScan website at http://www.zooscan.obs-vlfr.fr/, and the complete list of features and some illustrations are provided in Appendix B.

C. *Experimental Procedure*

We conducted a series of machine learning experiments. We carried out two different 8-way classification experiments in order to compare our efforts with contemporary results. We also conducted two different 16-way classification experiments in order to examine the tradeoff between complexity and performance.

For each of the 8-way experiments, we varied the data set size from 500 to 76,800 ROIs. For the 16-way experiments, we used data set sizes from 6,000 to 725,516 ROIs. We formed balanced data sets (equal types of each image class) to facilitate experimental design in addition to interpretation of results. For example, when evaluating the impact of adding classes, or which class is the most difficult, it is important that those be held constant. Also, balanced classes allow for more simple summary statistics, such as recall, to be used to measure performance.

We built our classifiers using Python's Scikit-learn [7]. We evaluated 2 types of support vector machines (SVM), 3 types of random forests (RF) including an extra trees ensemble (XTR) and a gradient boosted random forest classifier (GBC), stochastic gradient descent with two different types of loss functions (SGD), 2 types of k-nearest-neighbor algorithms (standard (kNN) and nearest neighbor Ball Tree (nnBT)), and neural nets (implemented as a multi-layer perceptrons with a single hidden layer - MLP). For each algorithm mentioned, we experimented extensively with hyperparameters, including hundreds of combinations for SVMs to thousands of combinations for RFs as described in Section III.B and

Appendix C. We consistently used an 80/20 split for training data vs testing data, with the exact same ROIs made available to each algorithm for training.

III. Experimental Results

Our experiments were designed to provide insight into six important facets of this machine learning problem: how many data to use, which algorithm, whether ensembling helps, whether abstaining helps, whether size fractioning the data helps, and the effectiveness/efficiency of using geometric features. The results presented are a representative sample, not an average. Each experiment was repeated multiple times.

A. Determining Data Set Size Requirements

Since machine learning algorithms can be computationally expensive, and obtaining training data can be expensive, we want to quantify the 'rate of return' on hand-labeled training data. To investigate, we trained suites of classifiers with different numbers of examples per class. Each point in Fig 5 represents an independently trained 8-way classifier. 500 examples per class seemingly provides asymptotic performance. However, we continued to conduct experiments and found continued improvement as training set size increased to ~4,000 (Fig. 6).

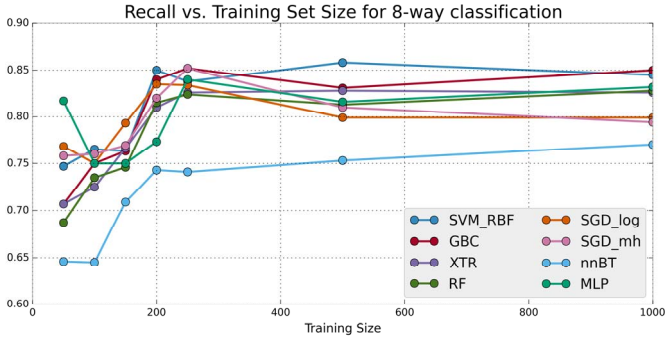

Fig. 5. Performance grouped by algorithm, shown with respect to training set size. Small data set sizes are noisy. The increase in performance apparently levels off after 500 examples per class. SVM_RBF is an SVM with a radial basis function for a kernel. SGD_log is stochastic gradient descent with log loss (logistic regression), and SGD_mh is SGD with 'modified huber' as a loss function. GBC,XTR,RF,nnBT, and MLP correspond with the descriptions in Section II.B.

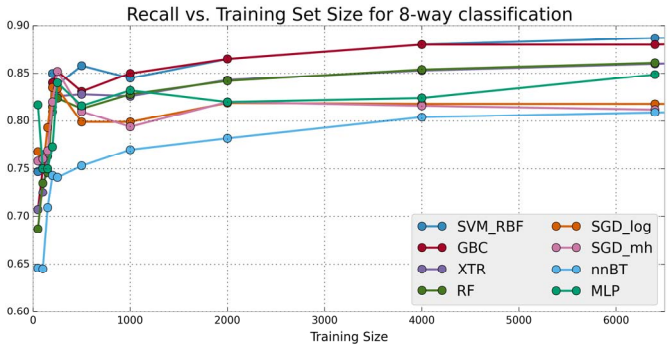

Fig. 6. Performance grouped by algorithm including larger training set sizes. Performance levels off after 4,000 examples per category for most algorithms.

For the numeric results summarized in Table I, the following holds regardless of algorithm: initially, doubling the training set size provides a 3-5% increase in performance; this rate decreases to a 1-2% improvement at larger training set sizes. The number of available expert-annotated ROIs in the 8 classes limited us to 7,680 training examples.

TABLE I. Recall Results for 8-way Classification Task

Training Size	SVM RBF	GBC	RF	SGD	nnBT	MLP
50	0.747	0.707	0.687	0.768	0.646	0.817
100	0.765	0.750	0.735	0.750	0.645	0.750
150	0.763	0.763	0.746	0.793	0.709	0.750
200	0.850	0.840	0.815	0.835	0.743	0.773
250	0.838	0.852	0.824	0.834	0.741	0.840
500	0.858	0.831	0.813	0.799	0.753	0.816
1000	0.845	0.850	0.828	0.799	0.770	0.832
2000	0.865	0.865	0.842	0.819	0.782	0.820
4000	0.880	0.880	0.854	0.818	0.804	0.824
6400	0.887	0.880	0.861	0.818	0.809	0.849
7680	0.888	0.883	0.864	0.818	0.811	

Figure 7 illustrates performance with respect to individual classes for the 8-way classification problem. Only four algorithms are shown: the results were consistent across all classifiers. In small sample sizes, the data are noisy, but as the training set size grows sufficiently large, calanoid copepods were consistently the most difficult category to classify, and eggs were the easiest. Not surprisingly, the more difficult classes also had the largest performance gain from additional training examples.

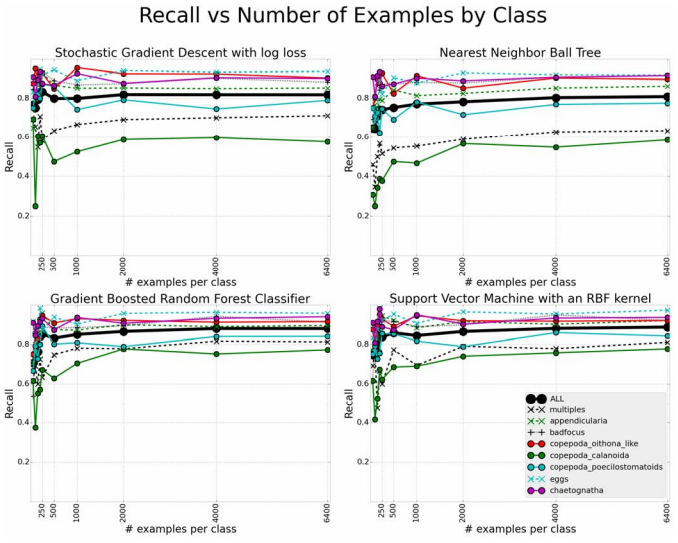

Fig. 7. Difficult classes tended to remain difficult, regardless of algorithm or data set size.

We consistently found SVM with an RBF kernel (SVM-RBF) and GBC to provide ~1-2% better accuracy than other methods.

Our overall results compare favorably with previous results. Our 8-way classification algorithm consisting of an ensemble of the GBC and SVM has an overall recall rate of 88.6%, which is 10 percentage points better than the best recall rate of 78% presented in Gorsky et al. [3]. In addition, the two best performing classes in Gorsky et al. are 'Bad Focus' and 'Fibers,' two inorganic classes.

Our results are also efficient for specific individual classes. For appendicularians we have a recall rate of 96.5% and a precision of 92.9%, as illustrated in Fig. 9, which is approximately 20-30 percentage points better than the performance in Forest et al. [8]. Our task is more difficult because we are attempting 8-way classification, where Forest et al. attempted 4-way classification. Their data set was much smaller, consisting of only 2,100 ROIs. For our training set of that size, 200 each for training and 50 for testing on each of 8 classes (2,000 total vignettes) we have an overall recall of 85%. For appendicularians specifically from that smaller data set we achieve a recall of 91.5% and a precision of 88.5%.

We achieve similar results to the phytoplankton classification task in [9]; Sosik and Olson had some simple classes for which 100% accuracy was achieved, and some ''difficult classes, such as detritus'' where only 68% accuracy was achieved. These percentages are on the order of our results. For example we also found detritus to be the most difficult category in our 16-way classifier, as shown in Figure 8. Sosik and Olson also found SVM with an RBF kernel to achieve the best results.

While minimal increases are obtained with larger sets, significant gains in recall are observed up through 4,000 examples. This value is a significantly different finding from Gorsky et al., who found that their performance plateaued at ~300 examples per class [3].

A more complicated, but closer to real world example is shown in Fig 8, which illustrates a confusion matrix for our best result for the 16-way classification problem. This classifier trained on 3,600 examples per class, which was the highest number available for all 16 classes, and achieved an overall recall rate of 0.813. When we trained a classifier on only 300 examples per class, as in [3], our best effort resulted in an SVM with an overall recall of 0.741, with similar types of errors as the confusion matrix shown.

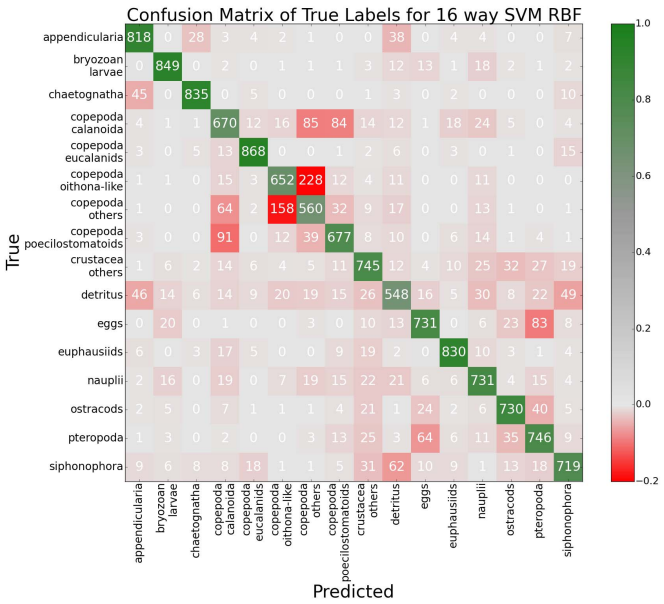

Fig. 8. Confusion matrix for our best results for the 16-way classification task. Depicted are the results for an SVM trained on 3,600 samples per class, and tested on 900 samples per class.

Ultimately, relative results are more important than absolute results, particularly because our ROIs are not a benchmark data set. For example, our categories of interest are often arbitrarily defined by the taxonomic resolution desired by a given lab. Note that in Fig 8, 862 of the errors are misclassifications of one type of copepod for another. Grouping all types of copepod into a single 'copepod' category would then raise performance by 0.075 to 0.888 on the 16-way task. Similarly, 'multiples' and 'bad focus' are two distinct classes in our data, but since they both refer to malformed ROIs, errors confusing these classes for each other should be considered less severe than all other errors.

Our relative results are competitive with other work. Our California Current sample data were previously processed with a RF classifier developed according to the results of [1]. The algorithm's recall is poor on all rare classes (often single digit accuracy), and in the teens for some common classes. Recall only exceeded 0.60 for a single class, detritus at 0.886. For a fair comparison of the SVM algorithm used in the present paper with a RF algorithm developed according to [1], classes were removed to create one 8-way classification task. Accordingly, the previously used RF implementation performed with 0.618 recall, compared to our recall of 0.887 in the present paper. For a 16-way classification task, the previous RF implementation had a performance of 0.580, compared to 0.813 for our implementation using 3,600 training items for each of the 16 classes. This 0.23-0.27 gain gives an idea of the improved performance, but slightly overestimates it because the real-world problem is harder than the treatment presented in this paper.

The 8 most prominent classes cover 85% and the 16 most prominent classes cover 92.9% of the data. So while the models presented in this paper were not trained with balanced data and not for our full 24-way classification problem, they

are useful as is, and could not perform worse than getting every single image from the rare classes incorrect. In the case of the 8-way classifier, this would yield an effective recall rate of 0.762 (an improvement of 0.221) and the 16-way classifier would yield an effective rate of 0.756 (an improvement of 0.215). Since the effective rate of the 8-way algorithm was better, it ultimately may be more effective to use a classifier with fewer classes and higher performance, which results in the need to completely sort difficult classes by hand.

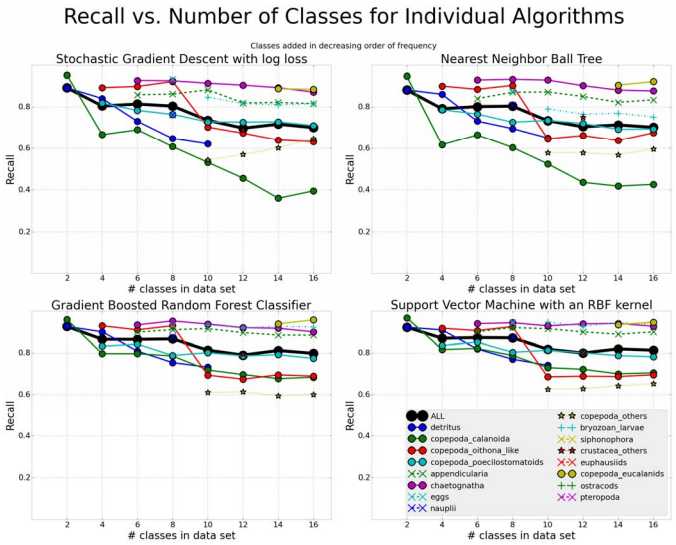

Fig. 9. An illustration of how increasing the number of classes affects recall rates. Four different algorithms are shown, and each algorithm has results for 8 different classification tasks presented. (2-way, 4-way, 6-way, 8-way, 10-way, 12-way, 14-way, and 16-way.)

Figure 9 shows how overall recall decreases when more classes are added. We trained a series of models where each algorithm had 3,600 training items, and additional classes were added to the training set in reverse order of overall abundance. Unsurprisingly, with each pair of additional classes performance decreased, although apparently more related to the difficulty of the added class than the overall number of classes. Note that improved performance on a more complex problem may be obtained by increasing the amount of training data.

B. Hyperparameter Tuning

The three algorithms with the best performance were the multi-layer perceptron (MLP), gradient boosted random forest (GBC), and support vector machine (SVM). In general, for the MLP, we found learning rate to be the most important single parameter as suggested by Bengio [11], and found that a large number of nodes in the hidden layer was not required. For the GBC, we ended up with shallow trees, a large number of samples per leaf, and moderate regularization. For the SVM, we found the regularization parameter needed to be increased on larger data sets, while the free parameter (gamma) was relatively constant.

For all experiments, cross validation consisted of a minimum of 5 folds during the hyperparameter search, but the final model was refit with the entire dataset. More information about hyperparameter tuning is provided in Appendix C.

C. Using an Ensemble to improve accuracy

Combining the results of two best performing classifiers consistently resulted in up to a 0.6% gain in recall, at no additional expense. Example results for one of our 8-way classification problems are shown in Table II. Our ensemble was done by averaging; each algorithm with the ability to returned a probability, rather than a classification. The estimated probabilities from all algorithms were then averaged pairwise, and evaluated as though they were the results of a single classifier.

Not surprisingly, the combination of the two best single-performing algorithms resulted in the strongest performance. Larger combinations of three or more algorithms sometimes achieved better results, but not as consistently as combining the SVM and the GBC. Overall, averaging provided improved results than either individual algorithms 33% of the time. Also, while not a strictly an improvement, note that SVMs helped every single other classifier exceed the recall that the other algorithm achieved independently.

TABLE II. RESULTS FOR AVERAGING 8-WAY PREDICTIONS (4000 TRAINING ELEMENTS/CLASS)

Algorithm(s) – Trained on 4,000 each	**Recall**	**Avg. Yields Improvement**
GBC and SVM_RBF	0.8866	Y
SVM_RBF and XTR	0.8855	Y
RF and SVM_RBF	0.8835	Y
SVM_RBF	0.8805	N/A
GBC	0.8799	N/A
GBC and RF	0.8783	N
GBC and XTR	0.8780	N
SGD_log and SVM_RBF	0.8758	N
GBC and SGD_mh	0.8755	N
GBC and SGD_log	0.8751	N
GBC and nnBT	0.8744	N
nnBT and SVM_RBF	0.8733	N
SGD_mh and SVM_RBF	0.8733	N
RF and XTR	0.8546	Y
RF	0.8536	N/A
XTR	0.8528	N/A
SGD_mh and XTR	0.8450	N
RF and SGD_mh	0.8441	N
RF and SGD_log	0.8439	N
SGD_log and XTR	0.8429	N
nnBT and RF	0.8401	N
nnBT and XTR	0.8375	N
nnBT and SGD_log	0.8278	Y
nnBT and SGD_mh	0.8276	Y
SGD_log and SGD_mh	0.8185	Y
SGD_log	0.8184	N/A
SGD_mh	0.8155	N/A
nnBT	0.8036	N/A

In addition, the improvement is primarily in the most difficult class, Chaetognatha, and improvement is consistent across numerous examples. The full impact of the best ensemble is shown in Fig 10.

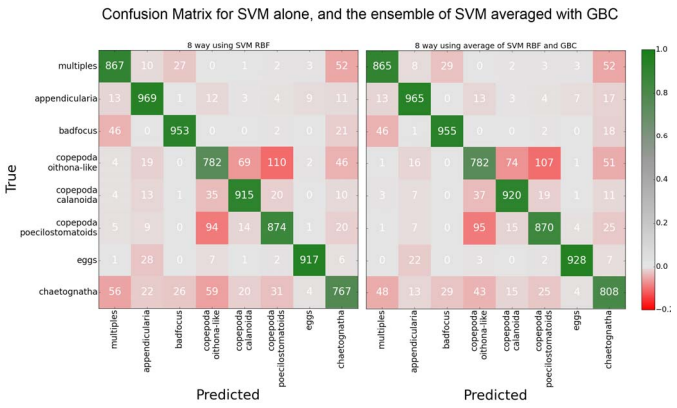

Fig. 10. The two confusion matricies shown illustrate the modest gains to be had by averaging the two best performing algorithms. The improvement is slight, 49 net additional correct classifications out of 8000, for an improvement of 0.61%

D. *Improving Abundance Estimation Through Abstentions*

By changing our classifier to output probabilities rather than labels we can allow abstentions. We allow for abstentions for ROIs with low confidence by ignoring guesses below a particular probability threshold. This technique eliminates false positives at the expense of some images remaining unlabeled. Therefore, this approach may be useful in circumstances where there is a high penalty for a false positive, but little penalty for a false negative. Table III provides an example.

TABLE III. ALLOWING ABSTENTIONS IN THE 8-WAY CLASSIFICATION MODEL (AVERAGE OF SVM AND GBC - 4000 TRAINING ELEMENTS/CLASS)

Confidence Threshold	% Labeled	Recall	% Labeled Correctly
0.3	0.9995	**0.8868**	0.8864
0.4	0.9941	**0.8904**	0.8851
0.5	0.9746	**0.8987**	0.8759
0.6	0.9279	**0.9196**	0.8533
0.7	0.8674	**0.9412**	0.8164
0.8	0.7943	**0.9600**	0.7625
0.9	0.6755	**0.9782**	0.6607
0.95	0.5445	**0.9867**	0.5372
0.99	0.2375	**0.9953**	0.2364

For example, setting the confidence threshold at 0.95 results in 0.9867 recall. This threshold results in the correct labeling of 4,299 of the original 8,000 ROIs and only 57 incorrectly labeled ROIs, as shown in Fig 11.

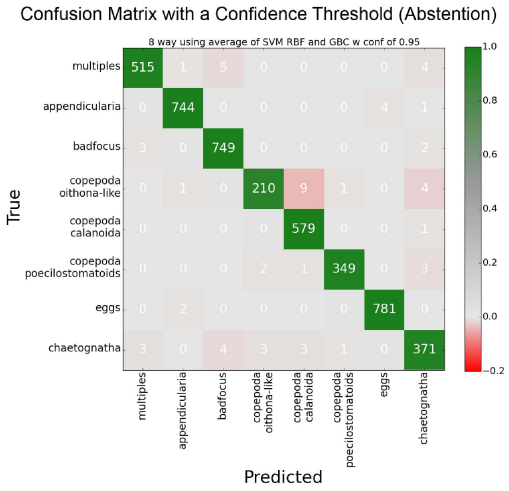

Fig. 11. Confusion matrix when the classifier is allowed to abstain from labeling images andonly classifying when probability exceeds 0.95. This approach greatly reduces the number of false positives compared to other classifiers.

If only some labeled images are required, and not a complete census, this technique may be useful for quickly labeling some images without incurring the expense of extensive manual resorting. We did not investigate whether the ratio of true positives to false negatives was more stable when allowing abstentions, but if so, an estimate of total abundance could be achieved through simply scaling these results.

E. *Efficiency Through Size Fractionation*

More examples in a training set improve recall but algorithm training times grow non-linearly with respect to the number of examples SVMs, for example, usually have a runtime of $O(n^2)$. According to Bottou [10], "(Runtime) grows at least like n^2 when C is small and n^3 when C gets large."

We show that size fractioning the data set can combat this penalty, and potentially allows accuracy beyond what the hardware could not otherwise achieve.

We performed a series of experiments creating specialist classifiers on different sizes of ROI. For example, we split the ROIs into quartiles by pixel area, and trained four independent classifiers. One model was trained on the smallest quartile of the ROIs, a second, independent model was trained on the next quartile larger ROIs, etc. The effect on recall was negligible. But most importantly, training four smaller classifiers is markedly faster than training one larger classifier. Completing the initial, coarse-grid search with cross validation on the single large model took 48 hours on our hardware. Training 16 specialist classifiers on size fractions of the ROIs, each over the same grid search, took 2 minutes per classifier, for a 100x speedup over the single classifier.

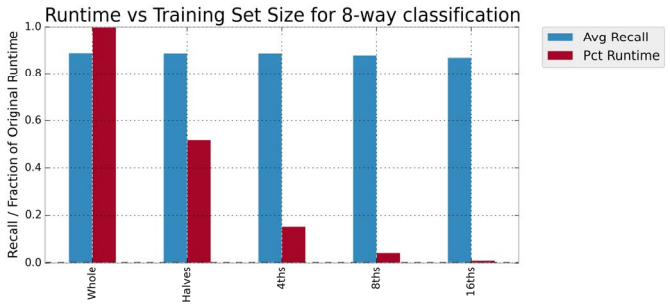

Fig. 12. Size fractioning the data results in significant time savings. Halving the data by size had minimal or no impact on recall, but drastically reduced execution time. Fraction of original runtime includes all classifiers from the group.

In one of our 8-way classification problems, size fractionating did not result in an overall gain, as none of the ensemble specialists is better than the baseline classifier on the whole data set. However, training multiple size-fractionated classifiers provides slightly better results than training a single, smaller classifier on all of the data, as shown in Fig.13.

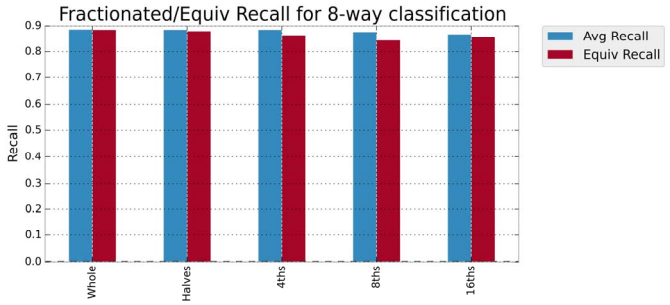

Fig. 13. The bar on the left is the baseline, a single classifier. The average of each ensemble (blue bars) is slightly higher than the recall of a single classifier of equivalent size (red bars).

To achieve maximum accuracy for our ROIs, creating a single large classifier slightly outperforms ensembles of size-fractionated models. However, size fractionating the data greatly reduces the training time, and in cases where the machine resources are limiting, creating multiple size-fractionated models will improve accuracy beyond creating a single classifier containing a selection of all of the data, as shown in the table below. In Table IV, if 8,000 examples per class are available then training a single 8-way SVM with 8,000 training examples per class yielded the best results. However, if the largest model able to be trained given hardware constraints is 1,000 examples per class, the results would be 0.02-0.03 better by training 8 models on various size fractions than trying to train a single model on data encompassing all size ranges. Size is a deterministic, objective criterion that does not require any human prescreening, and has a basis in the problem space (animals generally fall within certain size clusters per species) and therefore makes a reasonable separation criterion.

TABLE IV. SIZE FRACTIONATED RECALL VS. EQUIVALENT RECALL

Split	Recall			
	8-way classification Task		16-way classification Task	
	Split Avg	*Equiv Recall*	*Split Avg*	*Equiv Recall*
Whole	0.8869	N/A	0.8131	N/A
Halves	0.8856	0.8805	0.7948	0.775
4ths	0.8857	0.8650	0.7902	-
8ths	0.8770	0.8445	0.7843	-
16ths	0.8684	0.8580	0.7654	0.74

F. *Efficiency and Feature Set Size*

We use a set of only 51 features, and our algorithms learn on the order of thousands of parameters (weights) depending on the algorithm. 'Deep Learning', which usually means convolutional neural networks, has performed well in many image competitions and publications. In deep learning architectures, hundreds of millions of weights are learned, for example 133M to 144M weights were learned for 224x224 pixel images in [12].

Even with our much simpler features, our grid search and cross validation for some individual models took multiple days to complete on a system with 50 available CPU cores. While a few days may be an acceptable wait, the first model will not be the one ultimately used, and many will need to be trained before results are reliable. Deep Learning algorithms can take advantage of the high level of parallelism to utilize GPUs, but the computational cost of deep learning algorithms is still orders of magnitude above our method. While the cost of computing is cheap and classification accuracy frequently is maximized above processing costs, the number of parameters required for such networks is substantial.

To alleviate some of the computational expense of deep learning approaches, many researchers are using networks where the first set of filters has been copied from, or 'pre-trained' on a different data set, such as ImageNet data [13] (UCSD students, SciPy attendees, US Navy Scientists, personal communications, 2014-2015). Even copying filters still requires training millions of parameters in the later stages of the network.

However, in cases such as embedded systems, extremely large data sets, or initial investigations, a smaller set of features, such as the ones presented here is sufficient.

IV. CONCLUSION

In order to more effectively quantify our plankton samples, we executed a series of experiments to determine how to improve classification accuracy.

Carefully tuned support vector machines slightly outperformed gradient boosted random forest and multi-layer perceptron neural networks. Regardless of algorithm, performance increased until at least 4,000 training examples per class, although performance continued to increase with more data. Data set size impacted performance and had a bigger effect than choice of algorithm. In Table I, the first few rows of data with smaller training sets have a 10 percentage

point range between the lowest and highest performing algorithm. However, the columns show that training set size has an even bigger impact; the gain between an algorithm with less training data and the same algorithm with more data boosts recall by 15 percentage points or more. Correct hyperparameter tuning is also an important consideration. We share our methodology in Appendix A.

Our results are consistent across classes and algorithms. SVMs almost always performed best. Most hyperparameter searches ended up in the same narrow ranges across experiments. We found that creating an ensemble of our two best performing classifiers also increases performance at no additional computation or training cost.

Geometric features are inherently efficient compared to other approaches, and size fractioning the ROIs increases run time efficiency further. Our best results improve upon our previous random forest implementation by 22 percentage points. We found that our simple geometric features can achieve a recall of 0.887 for our best ensemble.

V. Appendix

A. Classification Labels

The 24 classification labels present in our data are: ['detritus', 'copepoda_calanoida', 'copepoda_oithona_like', 'copepoda_poecilostomatoids', 'multiples', 'badfocus', 'appendicularia', 'chaetognatha', 'eggs', 'nauplii', 'copepoda_others', 'bryozoan_larvae', 'siphonophora', 'euphausiids', 'crustacea_others', 'copepoda_eucalanids', 'ostracods', 'pteropoda', 'doliolids', 'others', 'radiolarians', 'polychaete', 'bubbles', 'copepoda_harpacticoida']. This list is sorted in order of frequency of occurrence.

B. Machine Learning Features

The 51 features we used for learning, spelled as provided by ZooScan are ['Angle', 'Area', 'Area_exc', 'CDexc', 'CV', 'CentroidsD', 'Circ.', 'Circexc', 'Convarea', 'Convperim', 'Elongation', 'Feret', 'FeretAreaexc', 'Fractal', 'Height', 'Histcum1', 'Histcum2', 'Histcum3', 'IntDen', 'Kurt', 'Major', 'Max', 'Mean', 'MeanPos', 'Median', 'Min', 'Minor', 'Mode', 'Nb1', 'Nb2', 'Nb3', 'Perim.', 'PerimAreaexc', 'PerimFeret', 'PerimMaj', 'Range', 'SR', 'Skelarea', 'Skew', 'Slope', 'StdDev', 'Symetrieh', 'Symetriehc', 'Symetriev', 'Symetrievc', 'ThickR', 'Width', 'X', 'XM', 'Y', 'YM'] [ref. 1]

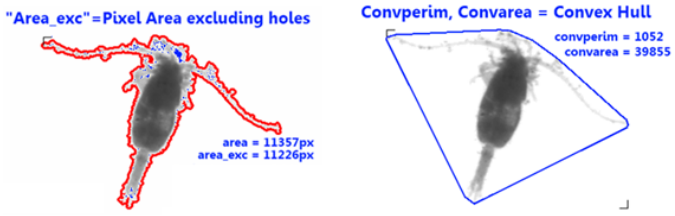

Fig. 14. Illustrations of how some of the feature values are calculated for actual ZooScan images.

C. Hyperparameter Optimization

For the multi-layer perceptron, a single hidden layer of 50 nodes often provided the best results. More nodes, even thousands, did not provide improved results. Learning rate was the most extensively searched hyperparameter, as it is reportedly the most important [11]. Initial searches covered many orders of magnitude. The final searches were fine-grained, with spacing of 2x (e.g. 0.05, 0.025, 0.0125). The optimal learning rate varied with training set size, and was most frequently 0.025 for smaller data sets with less than 500 examples per class, and decreased to 0.0025 for our largest data sets. The optimal L1 and L2 regularization hyperparameters were searched independently and consistently found to be 10^{-5} or 10^{-6}, with larger values found to be detrimental to performance.

For the Gradient Boosted Random Forest Classifier we evaluated four hyperparameters. For the maximum tree depth we tried values up to 25, but frequently a low value, such as 6, was optimal. We tried the odd-number values 3, 5, 7, 9 for the minimum samples per leaf, and the larger values, such as 7 or 9 provided the best performance. Maximum features were evaluated on deciles from 0-1, and intermediate values such as 0.3 performed best. Values of the number of estimators up to 2,500 were tried, and there was little discernable pattern.

For the support vector machine, the radial basis function with degree=3 was used for all reported results. For the regularization parameter, *C*, we experimented with various orders of magnitude from 1 to 100 million, but all results were obtained with values in the narrow range of 10,000, 100,000, or 1,000,000 with stronger regularization consistently providing better results on the larger datasets. For the free parameter, *gamma*, we again experimented with various orders of magnitude from 0.1 to very small, and found that except for very small sized data sets, gamma of 0.001 or 0.0001 was optimal.

References

[1] S. J. Bograd, D.A. Checkley, and Warren S. Wooster, "CalCOFI: A half century of physical, chemical, and biological research in the California Current System," Deep Sea Research Part II: Topical Studies in Oceanography, vol. 50, no. 14, pp. 2349-2353, 2003.

[2] M. D. Ohman, and P. E. Smith. "A comparison of zooplankton sampling methods in the CalCOFI time series." California Cooperative Oceanic Fisheries Investigations Report, pp. 153-158, 1995.

[3] G. Gorsky, M. D. Ohman, M. Picheral, S. Gasparini, L. Stemmann, J.-B. Romagnan, A. Cawood, S. Pesant, C. Garc ıa-Comas, and F. Prejger, "Digital zooplankton image analysis using the zooscan integrated system," Journal of Plankton Research, vol. 32, no. 3, pp. 285–303, 2010.

[4] L. Sala and M. D. Ohman, "Zooplankton of the San Diego Region," https://scripps.ucsd.edu/zooplanktonguide/, accessed 20 August 2015.

[5] P. Grosjean, M. Picheral, C. Warembourg, and G. Gorsky, "Enumeration, measurement, and identification of net zooplankton samples using the zooscan digital imaging system," ICES Journal of Marine Science: Journal du Conseil, vol. 61, no. 4, pp. 518–525, 2004

[6] S. Leli'evre, E. Antajan, and S. Vaz, "Comparison of traditional microscopy and digitized image analysis to identify and delineate pelagicfish egg spatial distribution," Journal of Plankton Research, vol. 34, no. 6, pp. 470-483, 2012.

[7] F. Pedregosa, et al. “Scikit-learn: Machine learning in Python.” The Journal of Machine Learning Research vol. 12, pp. 2825-2830, 2011

[8] A. Forest, L. Stemmann, M. Picheral, L. Burdorf, D. Robert, L. Fortier, and M. Babin, “Size distribution of particles and zooplankton across the shelf-basin system in southeast beaufort sea: combined results from an underwater vision profiler and vertical net tows,” Biogeosciences, vol. 9, no. 4, pp. 1301–1320, 2012.

[9] H. M. Sosik and R. J. Olson, “Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry,” Limnology and Oceanography: Methods, vol. 5, no. 6, pp. 204–216, 2007.

[10] L. Bottou and C. Lin, “Support vector machine solvers,” Large scale kernel machines, MIT Press, 2007, pp. 301-320, 2007.

[11] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” Neural Networks: Tricks of the Trade, Springer Berlin Heidelberg, pp. 437-478, 2012.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,”CoRR, vol. abs/1409.1556, 2014.

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton. “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, pp. 1097-1105. 2012.