Embedded System to Detect, Track and Classify Plankton Using a Lensless Video Microscope

Thomas G. Zimmerman[*], Vito P. Pastore, Sujoy K. Biswas, Simone Bianco
IBM Research-Almaden, San Jose, CA, USA, Center for Cellular Research, *tzim@us.ibm.com

Plankton provide a foundation for life on Earth. To advance our understanding of the marine ecosystem for scientific, commercial and survival purposes, more *in situ* continuous monitoring and analysis of plankton is required. Cost, complexity, power and data communication demands are barriers to widespread deployment of *in situ* plankton microscopes. We address these barriers by developing and evaluating a low-cost lensless microscope with a data pipeline optimized for the Raspberry Pi 3. The objective is to design a real-time *in situ* image capture and classification at a cost low enough for wide deployment, to monitor the health of waterways in the environment. The pipeline records 1080p video of multiple plankton swimming in a sample well while simultaneously detecting, tracking and selecting salient cropped images for classification @ 5.1 frames per second. Thirteen machine learning classifiers and combinations of nine features sets were evaluated on nine plankton classes, optimized for speed (F1=0.74 @ 1 msec. per image prediction) and accuracy (F1=0.81 @ 0.80 sec.). System performance results confirm that performing the entire data pipeline from image capture to classification is possible on a low-cost open-source embedded computer.

Index Terms—**Plankton, supervised machine learning, embedded system, detection, tracking, image processing, Raspberry Pi**

Plankton are vital for life on Earth, providing most of the breathable oxygen, carbon sequestration and larvae nutrition for the planet. Plankton are any organism that live in water and are unable to swim against current. This broad definition includes bacteria, phytoplankton and zooplankton and organisms like starfish that are only plankton during their larval (juvenile) stage of life. In this paper we are primarily interested in plankton within our imaging range (~50 to 1000 um).

To understand the environmental and biological processes that regulate plankton populations, a fundamental need is to count and classify species over time and space[1]. This is a daunting endeavor considering 71% of the Earth's surface is water-covered. To advance our understanding of the marine ecosystem more *in situ* continuous monitoring and analysis is required[2,3]. Our objective is to dramatically reduce the cost of *in situ* microscopes several orders of magnitude (e.g., <$1000) to substantially increase deployment opportunities[4]. Detection and classification must be done locally in the microscope to reduce the communication and processing burden created by thousands of *in situ* microscopes.

We used a lensless microscope composed of a point-light source and image sensor to view plankton swimming in a chamber. An image capture and classification pipeline was optimized to run on a Raspberry Pi 3 open-source platform, using the Pi's video co-processor (GPU) and ARM processors (CPU) to simultaneous capture and processing video, respectively. Image resolution was reduced to increase image processing throughput by removing objects too small to classify. Objects with low velocity were rejected to ignore sand, detritus and other inanimate objects. Merging and splitting events during tracking were detected as large changes in inter-frame object area. When tracking a moving object, classification was performed on the image presenting a profile view, indicated by a large aspect ratio. We evaluated combinations of features and classifiers to determine the optimal tradeoff of classification accuracy vs. computation time**.** Classifier training time was also considered for future development of *in situ* learning.

Results

Processing Time. The processing times for each module in the image processing pipeline running on a Pi 3 (Broadcom BCM2837 64 bit Quad Core Processor @ 1.2 GHz, 1GB DDR2 internal RAM), averaged over all the frames in the seven training videos are listed in Table 1. For each video the Median module is run once (run time: min=1.3 sec, avg=2.2 sec, max=2.8 sec), but listed as the time averaged over all the video frames to be consistent with the other metrics. The relative time of module procedures are listed in Table 2, indicating that over half of the Detector and Segment module's time was spent loading image frames from memory. Classification time was insignificant because it was performed once per tracklet (average length 146 frames), and used the optimal classifier (DCT, Decision Tree Classifier) with features already calculated for the Detect and Track modules.

Classification scores. Thirteen classifiers (Table 3) were run against all combinations of nine feature sets (Table 4). The feature set combination that produced the highest F1 score for each classifier is presented in Table 5. The highest F1 score (0.81) was achieved with the LDA classifier using a combination of Zernike moments, local binary patters, Haralick and intensity features, and required an average of 988 msec. for feature calculations and 10 msec. for inference. The best tradeoff of computation time vs. F1 score (1 msec. and 0.74, respectively), determined by a scatter plot (Figure 1), was a DCT classifier using features calculated for the Detect and Track modules (area, aspect ratio, normalized edge sum and contour smoothness). A confusion matrix for these two configurations is presented in Table 6.

Discussion

Our objective was to develop a low-cost microscope and data pipeline that could classify a few species of plankton at video rates. We developed an image processing pipeline that performs detection, tracking and supervised classification of nine plankton classes running on a Pi 3 with an average system throughput of 5.1 frames per second and an F1 classification accuracy of 0.74 using an inexpensive lensless microscope under a wide range of sample population densities. The high throughput was achieved by using a reduced image size for detection and tracking, performing classification with features calculated for detection and tracking, and simultaneously capturing and processing video with the Pi's GPU and CPU. Retrieving images from the video was the dominant computational load. By sharing images between the Detect and Segment modules, an average of 57 msec. per frame could be saved, increasing the throughput from 5.1 to 7.2 frames per second.

Algae presented a significant classification challenge as they can take on many shapes. Limiting detection to moving objects with low to moderate texture eliminated most algae from the classifier. Two plankton classes were added to accommodate algae that passed these requirements. The largest source of classification error was the confusion between Blepharisma and Paramecium, due to the shape of Blepharisma changes based on nutrition[5] and appearing like a Paramecium. We have since replacing the white LED with an inexpensive red laser (<$1), to create higher resolution holographic images to improve class discrimination, although image reconstructing adds considerable computation time.

There are over 4,000 species of plankton in the world[6], making it impractical to create a comprehensive dataset to train *in situ* microscopes to classify every possible species. Fortunately, the variety of species in a specific area (species richness) is low, typically in the range of 10-50[7]. This simplifies the classification task as each microscope must only classify the small collection of location-specific species it will encounter. But how to train thousands of microscopes for the small collection site-specific species each would encounter *a priori*? We postulate the solution to this conundrum is for each microscope to learn *in situ*[8]. Hence our interest in selecting a classifier with low training time. Finally, our microscope and pipeline tracks plankton trajectory, however this data was only used to select salient images for classification. Future work will examine the use of trajectory profiles as classification features, and to support environmental and behavior research.

Methods

Lensless microscope. A lensless microscope[9] (Figure 2) was used to created videos of plankton samples swimming in a glass sample well (12 mm square by 0.8 mm deep), illuminated by a white LED located 100 mm above the image sensor, casting shadows on the image sensor directly below (OV5647, 1.4 um square pixels, 3.76 x 2.74 mm sensing area). The resolution is limited by diffraction, making the objects look fuzzy and tiny objects appear as halos. The image sensor operated as a Pi camera, enabling full use of Pi camera libraries and utilities.

Plankton samples. Seven live plankton species (Figure 3) (Protozoa Set Item #131020, Carolina Biological Supply Company, Burlington, NC) were used to create a data set of plankton images. For each plankton species, at least two short videos (71 or 144 seconds) were recorded, one for training and one for testing. A 50 uL solution of plankton was hand pipetted onto the image sensor well and allowed to stabilize for 10 seconds. The videos were recorded using the Pi OS *rapsivid* utility with the default settings (automatic gain, 1920 x 1080 resolution, 30 frames per second, H.264 encoding, ~ 1.6 MB/sec file size). The videos were converted to MP4 format with the open-source FFmpeg utility to enable random access to frames. A library of 24 videos were created. Each video featured one species, although contamination with algae, tiny ciliates and detritus was common. The videos were reviewed by a human observer for quality. Some videos were rejected for excessive concentration of plankton, a rare condition *in situ*, or for lacking the target species. Of the remaining videos, seven videos were selected for training and seven for testing.

Training and test set. A collection of plankton images was automatically generated from the seven training videos using the data pipeline described in this paper (Figure 4). From this collection, 33 images were manually selected for each of the seven plankton classes to train the classifiers. There were a considerable number of images of algae with morphology distinct enough to create two additional groups, labeled *Ag1* and *Ag2*. The supervised selection process produced a total balanced training set of 9 classes with 33 members per class (Table 7).

The seven test videos were passed through the data pipeline to create a collection of 1237 salient test images. Each image was assigned one of ten categories by a human observer: one for each of the nine classes and one for a rejection group. The rejection group consisted of 54 images (4.4%) of objects that the human observer could not identify, typically unspecified algae or algae that obscured a species beyond recognition. The final unbalanced test image set consisted of 1183 labeled images.

Reducing computation time and image fragmentation. Video was recorded at full resolution (1920x1080@ 30 fps) to provide high resolution images for feature calculations (Fig. 5a). However, the object bounding box coordinates used for image cropping did not require high position resolution (Fig. 5b). Reducing image resolution reduced image fragmentation, a condition where one object was detected as multiple objects, and eliminated small objects (e.g.<50 um) that do not present enough spatial features for classification. At full resolution, the average number of objects per frame in the training videos ranged from 503 (*Stentor*) to 7687 (*Euplotes*). The *Euplotes* videos had a high concentration of tiny algae, detritus and fragmentation of large objects. When frame resolution was resized by a factor of 15, the range reduced to 45 (*Stentor*) and 354 (*Euplotes*).

Correcting for uneven illumination. The microscope's point light source created uneven lighting across the image sensor, forming a bright center that decreases radially. For each video, a reference intensity image was created by calculating the median intensity from 20 randomly selected frames (resized by a factor of 15), ignoring the first 100 frames to allow the image system's automatic gain control (AGC) time to settle. During detection, the reference intensity image is subtracted from each video frame, to enable the use a global threshold (a fast and simple method to binary quantize a high contrast image).

Plankton detection. The Detector module sequentially processed each rescaled frame (128x72), detecting and saving the coordinates of objects within an acceptable area range for the nine classes (1k to 40k pixels), using the OpenCV *findContours* method. Accepted objects were dilated with a kernel size of seven, then the *findContours* method was applied again. If more than one object was detected, the object with the largest area was selected. The contour of the final selected object was filled with the *drawContours* method, creating a solid blob object without holes suitable for feature extraction.

Plankton Tracking. The Track module received a list of objects from the Detect module and created tracklets; a sequence of object positions (Figure 6). Each tracklet was assigned a unique label (ID) that persisted across frame boundaries, beginning when an object was first detected and terminated when it left the viewing area or encountered another object (merging). A tracket also began when a merged object separated (splitting). Merging and splitting were detected when the relative area of an object's bounding box across a frame boundary dramatically increased or decreased, respectively. A tracklet was rejected when merging occurred. A new tracket was created and assigned a new ID when splitting occurred. Each current object was matched to the closest unassigned object in the previous frame, and assumed it's ID. If the distance was too great, the object in the current frame was assigned a new ID. Once this matching process was completed, any unassigned objects were considered "ghost" objects and retained their ID for up to ten frames.

Trackets less than ten frames long or without substantial movement were rejected, as they were often algae or detritus. For the seven training videos, 68% of the trackets were sufficiently long, while only 24% were sufficiently long and moving. For each acceptable tracklet, the Segment module selected a salient image for classification, defined as the image of the object with largest aspect ratio that does not touch the image boarder. Each salient image was cropped from the high-resolution color frame using the object's low-resolution rectangular bounding box coordinates provided by the Detect module.

Feature Extraction. Nine combinations of feature sets were evaluated for their contribution to the accuracy, training, and testing times of all 13 classifiers (Table 5). The first feature set were those used by the Detect and Track modules, preferred since they added no additional computational load. They consist of object area, aspect ratio, texture (sum of Canny edge pixels divided by area) and contour smoothness (contour pixels divided by square root of area). Area was also evaluated as an independent feature for it is the most common observable trait for binning plankton[10]. The remaining seven feature sets examined shape (Fourier descriptors, geometry, Hu and Zernike movements) and texture (local binary patterns, Haralick, intensity histogram)[8].

Classification. Thirteen classifiers were selected from the scikit-learn.org library, to represent a diverse group of multi-class supervised learning methods, and were used without any custom tuning. Neural networks were not considered due to their relatively long inference time and extremely long training time.

References

[1] Roelke, D.L., Spatharis, S., "Phytoplankton succession in recurrently fluctuating environments," *PLOSONE,* DOI:10.1371/journal.pone.0121392 (2015)

[2] Quéré, C.L., *et al.*, "Ecosystem dynamics based on plankton functional types for global ocean biogeochemistry models," *Global Change Biology,* DOI:10.1111/j.1365-2486.2005.1004.x (2005)

[3] Erickson, J., *et al*., "In Situ phytoplankton analysis: there's plenty of room at the bottom," *Analyt Chem*, pp 839-850, 2012, DOI:10.1021/ac201623k (2012)

[4] Hunter, K., "Detecting deadly algae", *Science Notes*, UC Santa Cruz, retrieved from https://ucscsciencenotes.com/feature/detecting-deadly-algae/ (2021)

[5] Giese, A.C., Smith, A.M., "Blepharisma in introductory biology", *Am. Bio. Teacher,* Vol. 35 No. 7, pp. 407-419, DOI:10.2307/4444465 (1973)

[6] Sournia, M.J., Chrdtiennot-Dinet, M.R., Ricard, M., "Marine phytoplankton: how many species in the world ocean?" *J. of Plankton Res.*, pp 1093–1099, DOI: 10.1093/plankt/13.5.1093 (1991)

[7] Stomp, M., Huisman, J., Mittelbach, G., Litchman, E., Klausmeier, C., "Large-scale biodiversity patterns in freshwater phytoplankton," *Ecology*, pp 2096-2107, DOI:10.2307/23034942 (2011)

[8] Pastore, V.P., Zimmerman, T.G., Biswas, S., Bianco, S., "Annotation-free learning of plankton for classification and anomaly detection", *Sci Rep* **10,** 12142, DOI:10.1038/s41598-020-68662-3 (2020)

[9] Zimmerman, T., Smith, B., "Lensless stereo microscopic imaging", *SIGGRAPH*, page 15, DOI: 10.1145/1278280.1278296 (2007)

[10] Omori, M., Ikeda, T., "Methods in marine zooplankton ecology," *Krieger Publishing Company*, ISBN 978-0-89464-653-9 (1992)

Acknowledgements. This work is funded by the National Science Foundation (NSF) grant No. DBI-1548297.

Disclaimer: Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Author contributions. T.Z., S.B., V.P. conceived the experiments. T.Z. built the microscope, coded the pipeline and analysis, created all the figures and tables, and wrote the paper. V.P. selected and coded most of the feature methods and collected all the plankton videos. S.K.B. provided tracking advice, S.B. is the Principle Investigator for the project and reviewed the original manuscript.

Competing interests. The authors declare they have no competing interests.

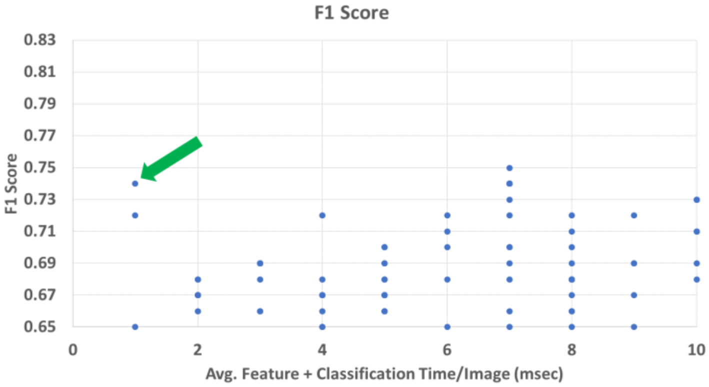


Figure 1. Each point is the F1 score of a combination of nine feature sets and 13 classifiers. The x axis is the average time to calculate the features and classify one image. The plot is used to determine the optimal tradeoff of computation time and classifier accuracy. The green arrow points to the Decision Tree Classifier (DCT) using geometric and area features, considered to be the optimal tradeoff of computation time and accuracy. Classification requires an average of 1 msec. per image. The feature computation time is considered zero since the features are calculated and used by the Detector and Tracker modules.

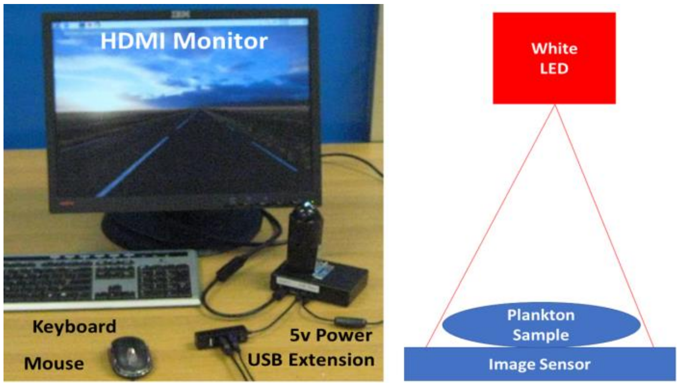


Figure 2. Lensless microscope used to capture video of plankton samples. (Left) Lensless microscope system build with a Raspberry Pi 3. (Right) A white LED 100 mm above the image sensor casts shadows of plankton swimming on top of the image sensor's protective glass cover.

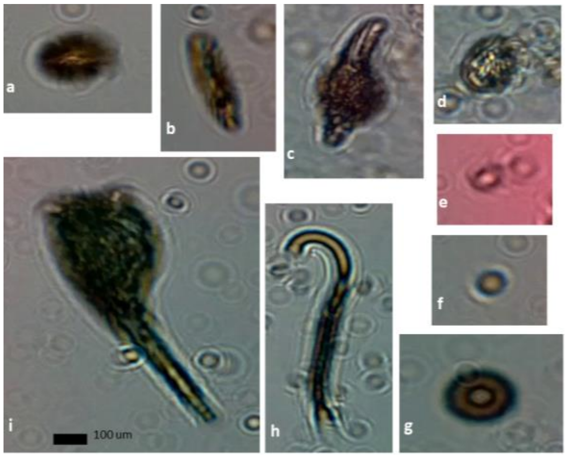


Figure 3. Sample training images for nine classes. All images are displayed at the same scale. (a) Didinum (b) Paramecium (c) Blepharisma (d) Euplotes (e) Algae1 (f) Algae2 (g) Volvox (h) Dileptus (i) Stentor

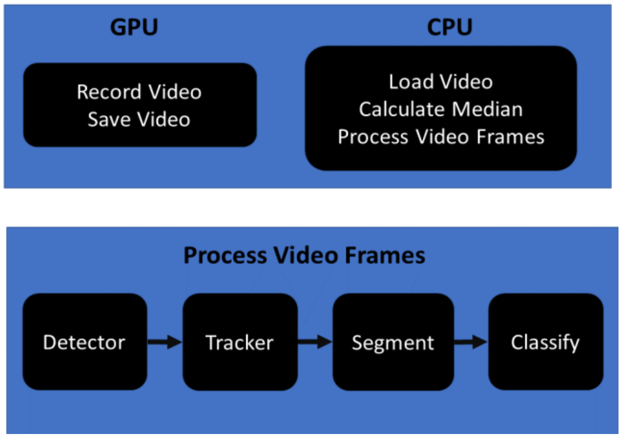


Figure 4. Image processing pipeline used to detect, track and classify plankton. (Top) The GPU records and saves videos while simultaneously the CPU loads the last saved video, calculates a median image from 20 randomly selected frames, then processes the video frames. (Bottom) Each module processes the entire video and passes results to the next module. The Detector reduces image resolution to speed up detection and outputs a list of object bounding boxes. The Tracker assigns object ID's by inter-frame proximity and outputs a list of tracklets. For each tracket, the Segment module selects the object with the largest aspect ratio (indicating a profile view), extract features and performs a classification.

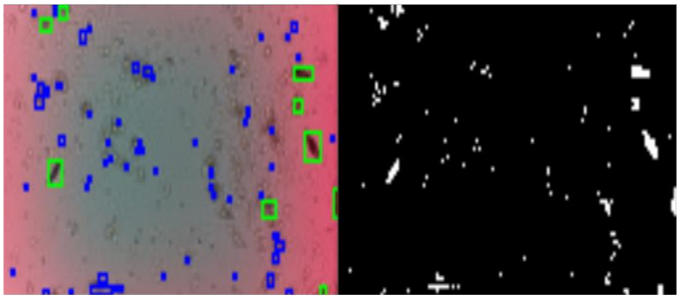


Figure 5. Reducing frame resolution to remove small objects. (Left) Frame from captured video at full resolution (1080p). Green box indicates object with acceptable area. Blue box indicates small rejected object. (Right) Frame resized by factor of 15 (128x72) to reduce detection time and remove tiny objects.

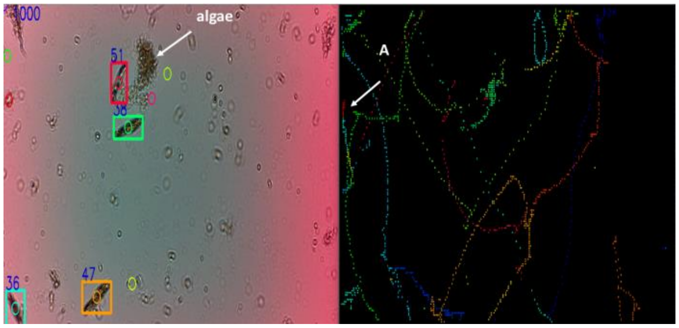


Figure 6. Creation of tracklets from detected objects in a sequence of frames. (Left) Eight objects are detected (circle) in this video frame. Four of them are currently being actively tracked (rectangular box with ID's). The clump of algae to the right of ID 51 is rejected by the Detector module. (Right) Location trails of nine objects integrated over 1000 frames. Each trail color represents a tracklet of an object with a unique identification (ID). Each dot is a position, determined at 30 times a second (video capture frame rate). The trail ending at label "A" is created by an object that goes in and out of the frame, creating short tracklets (red, cyan, yellow, blue, green) that are rejected by the Segment module for being too short.

MODULE	AVG	MIN	MAX
MEDIAN	0.9	0.5	1.5
DETECT	93.3	87.5	100.1
TRACK	2.9	0.9	7.4
SEGMENT	98.9	88.5	126.5
CLASSIFY	7.4	3.6	22.5
TOTAL	203.4	181.0	258.0
FPS	4.9	5.5	3.9

Table 1. Average frame processing time (milliseconds) for each module. The Median is performed once per video, but its time is presented averaged over all the frames to enable easy comparison to other modules. FPS is frames per second (1/total time).

MODULE	PROCEDURE	AVG %	AVG %
MEDIAN		<1%	
DETECT		46%	
	Load Frame		53%
	Image Processing		41%
	Detect Objects		5%
TRACK		1%	
SEGMENT		49%	
	Load Frame		58%
	Process Segments		37%
	Save Images		5%
CLASSIFY		4%	

Table 2. The relative processing time of selected modules and procedures contained in the modules.

Key	Classifier
KNN	**Nearest Neighbor**
NCT	**Nearest Centroid**
DTC	**Decision Tree**
RFC	**Random Forest**
GNB	**Gaussian Naïve Bayes**
LDA	**Linear Discriminant Analysis**
QDA	**Quadratic Discriminant Analysis**
LSV	**Linear Support Vector Machine**
NSV	**Nu-Support Vector**
BNB	**Bernoulli Naïve Bayes**
ABC	**Ada Boost**
GBC	**Gradient Boosting**
ETC	**Extra Trees**

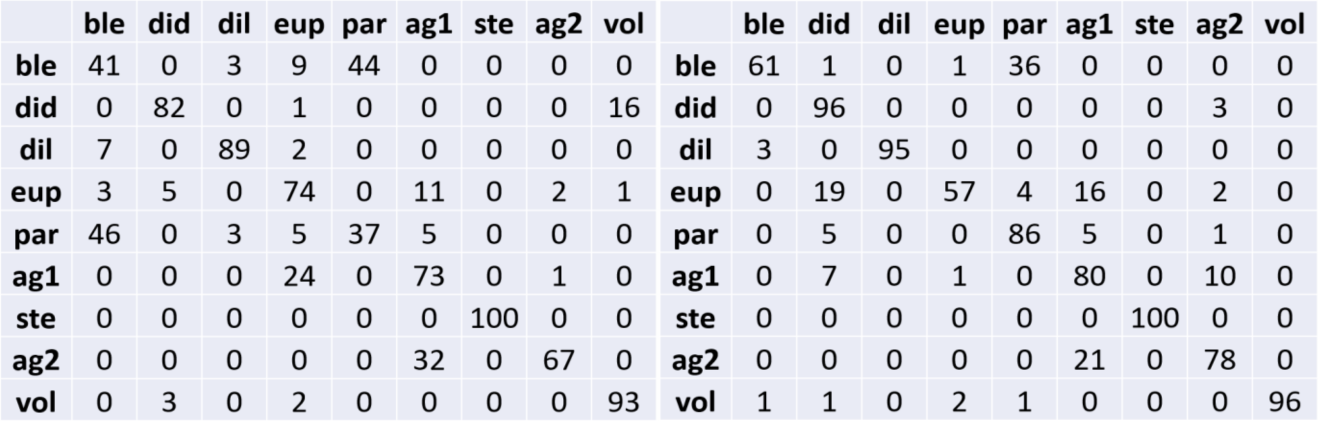
Table 3. The thirteen classifiers evaluated in the paper.

Key	Feature Set	Features	Calculation Time (msec)
g	Detector geometry	4	0
a	Area	1	0
G	Geometric	14	2.2
h	Hu moments	7	0.3
z	Zernike moments	25	218.6
L	local binary patterns	54	572.8
H	Haralick intensity	13	193.3
i	Intensity	8	3.2
f	Fourier descriptors	10	72.4

Table 4. The nine feature sets used in the paper, with the number of features in each set and average calculation time per object (milliseconds) running on a Pi 3.

Classifier	Features	Acc	P	R	F1	featureTime	TrainTime	PredictTime	TotalTrain	TotalPredict
LDA	zLHi	0.83	0.82	0.84	0.81	988	169	10	1157	998
LSV	gzLi	0.83	0.8	0.85	0.81	795	754	9	1549	804
NSV	gGLi	0.8	0.78	0.77	0.77	578	95	211	673	789
RFC	gaGLHi	0.79	0.77	0.8	0.77	772	106	21	878	793
ETC	ghzHi	0.78	0.75	0.77	0.76	415	68	19	483	434
KNN	gahzi	0.78	0.75	0.77	0.75	222	5	402	227	624
NCT	gahzi	0.77	0.74	0.77	0.75	222	4	7	226	229
DTC_OP	ga	0.74	0.73	0.75	0.74	0	6	1	6	1
GNB	gai	0.76	0.74	0.76	0.74	3	8	12	11	15
BNB	gaGzi	0.75	0.72	0.78	0.73	224	7	8	231	232
GBC	gazi	0.72	0.74	0.73	0.73	222	6340	66	6562	288
QDA	g	0.71	0.74	0.71	0.7	0	8	8	8	8
ABC	zLHif	0.45	0.45	0.53	0.46	1060	1460	338	2520	1398

Table 5. The combination of feature sets for each classifier that results in the highest F1 score. Included in the table is the classifier and feature set combination that produces the best tradeoff of accuracy and computation speed (labeled DCT_OP). The key for the feature combinations are listed in Table 4.



	ble	did	dil	eup	par	ag1	ste	ag2	vol
ble	41	0	3	9	44	0	0	0	0
did	0	82	0	1	0	0	0	0	16
dil	7	0	89	2	0	0	0	0	0
eup	3	5	0	74	0	11	0	2	1
par	46	0	3	5	37	5	0	0	0
ag1	0	0	0	24	0	73	0	1	0
ste	0	0	0	0	0	0	100	0	0
ag2	0	0	0	0	0	32	0	67	0
vol	0	3	0	2	0	0	0	0	93

	ble	did	dil	eup	par	ag1	ste	ag2	vol
ble	61	1	0	1	36	0	0	0	0
did	0	96	0	0	0	0	0	3	0
dil	3	0	95	0	0	0	0	0	0
eup	0	19	0	57	4	16	0	2	0
par	0	5	0	0	86	5	0	1	0
ag1	0	7	0	1	0	80	0	10	0
ste	0	0	0	0	0	0	100	0	0
ag2	0	0	0	0	0	21	0	78	0
vol	1	1	0	2	1	0	0	0	96

Table 6. Confusion matrix for optimal combination of classifier and features sets. (Left) Optimal combination for best tradeoff of accuracy and time. (Right) Optimal combination for fest F1 accuracy. The key to the plankton class is listed in Table 7.

Class	**Species**	**Train Images**	**Test Images**
blep	***Blepharisma***	**33**	**328**
eup	***Euplotes***	**33**	**169**
did	***Didinium***	**33**	**177**
par	***Paramecium***	**33**	**158**
dil	***Dileptus***	**33**	**132**
vol	***Volvox***	**33**	**82**
ag1	***Algae1***	**33**	**58**
ste	***Stentor***	**33**	**51**
ag2	***Algae2***	**33**	**28**
	Total Images	**297**	**1183**

Table 7. Nine classes of plankton evaluated in the paper. Algae contaminated many of the plankton samples and can take on many shapes which tends to confuse the classifiers. Most of the images of algae were detected by their low velocity and high texture and removed by the pipeline before classification. However, some algae passed through the algae detector and were added as two additional classes (*ag1* and *ag2*).