Automatic image-based phytoplankton classification tool

Matheus Campos Fernandes[a], Thiago Ferreira Covões[a], André Luiz Vizine Pereira[b], Jose Juan Barrera Alba[b], Adriana Flueti Ciofi[b],

[a]*Center of Mathematics, Computing and Cognition (CMCC),*
Federal University of ABC (UFABC)
Santo André, Brazil
[b]*Instituto Do Mar*
Federal University of São Paulo (UNIFESP)
Santos, Brazil

Abstract

Planktonic organisms have been identified as indicators of changes in coastal ecosystems due to their rapid growth, rapid response to environmental changes and their trophic and morphological plasticity. An automatic system based on image analysis and classification of phytoplankton cells would be useful to monitoring the microplankton community in coastal systems. The main purpose of this paper is to develop a plankton classification software. This consists in identifying different types of phytoplankton based solely on image data. To compare different classifiers in such classification problem, we consider a dataset collected in the coastal region of São Paulo. After acquiring an image set, we employ computer vision techniques to perform steps of object identification, filtering and feature extraction in order to obtain our final dataset. We obtain a mean accuracy of 83.88% considering 4 classes, and 76.67% considering 13 classes. We also test the same methods on a related dataset used in a recent online machine learning challenge, for which we obtain 45.91%. Our results on the smaller-sized dataset are competitive with the literature even with images with smaller resolution. The developed software is open-source and available under the MIT license.

Keywords: Plankton Classification, Image-based Classification, Supervised Learning

1. Introduction

Coastal regions historically have been under intensive anthropic pressure, which leads to ecological imbalances, and are influenced by natural events, that also cause disturbances in these environments [1]. These environments are influenced, in different ways, by tidal regimes, fresh water discharge, vertical mixing and turbidity [2], as well as by the discharge of nutrients [3], the latter, especially notable in areas with intense human occupation.

Due to their rapid growth, rapid response to environmental changes and their trophic and morphological plasticity, planktonic organisms have been identified as indicators of changes in coastal ecosystems, which in turn favor the growth of phytoplankton and the maintenance of high biomass of trophic levels [4]. Considering that man-induced eutrophication can increase the occurrence of proliferation of certain harmful algae, the development of rapid-response monitoring methods for phytoplankton classification is imperative. In this sense, the recent events of emergence of dinoflagellates species potentially producing toxins in the coast of São Paulo have generated concern in the environmental agencies, showing the need to establish monitoring programs.

Email addresses: `fernandes.matheus@ufabc.edu.br` (Matheus Campos Fernandes), `thiago.covoes@ufabc.edu.br` (Thiago Ferreira Covões), `andre.vizine@unifesp.br` (André Luiz Vizine Pereira), `barrera.alba@unifesp.br` (Jose Juan Barrera Alba), `adriana.ciofi@unifesp.br` (Adriana Flueti Ciofi)

Phytoplankton monitoring has been systematically based either individually or in combination on cell density, on chlorophyll biomass, occasionally including accessory photosynthetic pigments [1], and on analysis of variation in taxonomic composition [5]. The latter involves the use of microscopy methods, which require a high degree of technical training and high investment in analysis time. In this sense, the development of an automatic methodological system based on image analysis and classification of phytoplankton cells to improve a morphometric and functional database would be useful to assist in the establishment of a long-term monitoring program of the microplankton community in coastal systems.

Therefore, we approach this as a classification problem, which consists in identifying different types of phytoplankton based solely on image data. The main purpose of this work is to develop software that helps researchers to quickly classify phytoplankton species based on microscope image data. To compare the behavior of different classifiers in such problem, we consider two plankton datasets. The first, collected by one of the authors, consists of samples from Santos on the coast of São Paulo. The second dataset was collected by researchers of the Oregon State University in the Straits of Florida [6] and made public for use in the Kaggle National Data Science Bowl Challenge 2015 [7].

The remainder of this paper is organized as follows. The next section discusses related work. Section 3 presents the methodology adopted to prepare our dataset, obtained in the coast of São Paulo, for classification. The empirical results we obtained on this dataset and the Kaggle dataset are discussed in Section 4. Section 5 explains the plankton classification software we developed. Finally, Section 6 concludes the paper and points to future work.

2. Related Work

Microorganism classification based on microscope imagery has received considerable attention in recent years, e.g., [8, 9, 10, 11, 12, 13]. A review of this literature is presented in [11], which shows that accuracy with water-borne microorganisms is usually around 80-90%, with a few exceptions over 90%. Also, shape features are the most explored type of feature, and Support Vector Machines (SVM) [14] and Neural Networks [15] are the algorithms with best results.

A Kaggle competition [7] involving plankton image classification took place in 2015. Several papers originated from the challenge, many of which used deep learning methods [16], such as [17, 18, 19]. Nevertheless, more classic approaches were also used. [20] employed a combination of handcrafted features descriptors with SVM and obtained competitive results in terms of F-Measure with well known deep learning models, e.g., AlexNet [21]. [22] used an approach based on multiple kernel learning. They extracted well-known features (geometric, grayscale, texture, etc), and reported a F-measure of 0.83.

We bring attention to the fact that deep learning is widely used in practical applications involving image-based classification, such as the Kaggle competition, due to its high accuracy. However, training deep neural networks requires a large amount of human-labeled data and, while plankton images could be quickly collected with the right equipment, manually labeling such images is a labor-intensive process, which is why this paper evaluates more traditional approaches that require fewer labels. This problem is further investigated in [23], which avoids labeling all plankton images by applying Active Learning techniques to select the most informative images.

Similar methods to the ones presented in this paper were used in [12]. The main difference lies in the image acquisition method, for which they used more sophisticated techniques (a submersible image capturing equipment, consisting of a collimated infrared LED beam and a high-resolution camera executed by an automated control system). After isolating the 15 classes that contain more than 100 images, they achieved 79% accuracy with SVM. Our work diverges from this due to the different classes under consideration and the use of microscope images, which, although of lesser quality, is the traditional way of obtaining phytoplankton imagery, and economically more viable.

3. Dataset

Water samples containing different types of plankton were collected in Santos, on the coast of São Paulo, and images were captured by the Zeiss Axiovert A.1 microscope. The images were then cropped to keep only the region that contains organisms, as well as classified by a specialist. A single image may contain one or more organisms, but they always belong to the same class.

Our final image set consists of 790 plankton images that were classified in two different levels of specificity:

- 4 “generic” classes: Cyanobacteria, Chlorophyta, Diatoms, Dinoflagellates.
- 13 “specific” classes: *Anabaena*, *Chaetoceros*, *Coscinodiscus*, *Cylindrotheca*, *Dinophysis*, *Dunaliella*, *Odontella*, *Prorocentrum*, *Protoperidinium*, *Pseudo-nitzschia*, *Tetraselmis*, *Thalassionema*, *Thalassiosira.*

To perform machine learning based image classification, we extracted the organisms from each image and extract relevant features. Thus, each image goes through steps of object identification, filtering and feature extraction. Figure 1 illustrates the complete process to generate the dataset from images. All code to perform this tasks was written in the Python programming language [24], using OpenCV [25] and Scikit-learn [26].

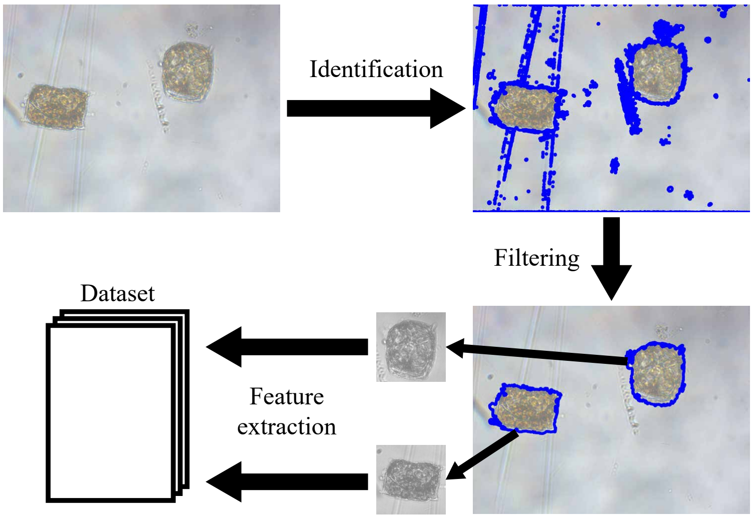


Figure 1: Steps to generate the dataset.

In order to find organisms, we first binarize each image and, then, apply object detection techniques to find Regions Of Interest (ROIs). Different methods were tested in sequence:

1. Otsu Thresholding [27];
2. Canny Edge Detection [28], Gaussian Blur;
3. Mean Adaptive Threshold, Gaussian Blur, Otsu Thresholding;
4. Median Blur, Gaussian Adaptive Threshold, Closing, Opening;
5. Median Blur, Mean Adaptive Threshold, Closing, Opening;
6. Histogram Equalization, Median Blur, Threshold, Closing, Opening;

Each method is described in more details in [29]. However, while all segmenters achieved satisfactory results in some images, they also failed in others. A segmenter ensemble was then used. A pixel is considered an object only if at least two segmenters consider it as one. A contour detection method [30] is performed on the final binary image.

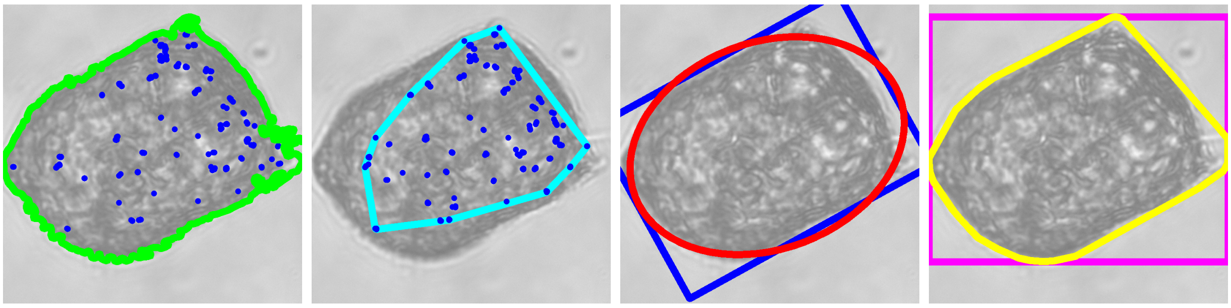
As inherent from this type of application, not all objects that were identified are actually organisms: they can be dirt particles, detritus, optical distortions and other types of noise, and thus need to be removed from the dataset through a filtering process. Therefore, we discard the contours which have an area smaller than a certain threshold ($t_{noise} = 3000px^2$), because very small objects are probably noise. Furthermore, it is important to make sure that the contour we found has information inside of it, otherwise it doesn't represent an actual informative object. For that, we use the algorithm Oriented FAST and Rotated BRIEF (ORB) [31] to detect the image keypoints, and then discard the objects which do not contain any keypoints.

We use ORB again to find the keypoints of the whole image ($\mathcal{K}_{full}$) as well as the detected object ($\mathcal{K}_{obj}$). We extracted 47 keypoint and shape-related features from each identified object, which are detailed in Table 1 and Figure 2. Most of the attributes were based on the work of [12, 32]. In particular, the three convexities are presented in [33], while Hu Moments and Haralick attributes were proposed respectively in [34] and [35].

Table 1: Features extracted from each ROI. The symbols are further detailed in Figure 2.

Feature	Description
Distance 1 to Distance 5	Five smallest distances between $\mathcal{K}_{obj}$
Mean Distance	Mean of all distances between $\mathcal{K}_{obj}$
Distance Standard Deviation	Std of all distances between $\mathcal{K}_{obj}$
Keypoint count	$\lvert\mathcal{K}_{obj}\rvert$
Keypoint hull area	Area of H_k
Full keypoint count	How many of $\mathcal{K}_{total}$ are inside C_{obj}
Rectangle mean intensity	Mean intensity of all pixels inside R
Ellipse mean intensity	Mean intensity of all pixels inside E
Aspect Ratio	R longest axis / R smallest axis
Area	C_{obj} area
Convex hull area	H_{obj} area
Extent	C_{obj} area / B area
Perimeter	C_{obj} perimeter
Convex Hull Perimeter	H_{obj} perimeter
Convexity 1	C_{obj} area / H_{obj} area
Convexity 2	H_{obj} perimeter / C_{obj} perimeter
Convexity 3	R perimeter / C_{obj} perimeter
Compactness	Circunference of a circle (C_{equiv}) with the same area as C_{obj}
Heywood Circularity	C_{obj} perimeter / Circularity
Waddel Circularity	C_{equiv} diameter
Rectangularity	C_{obj} area/R area
Eccentricity	$\frac{\sqrt{A_R^2-a_R^2}}{A_R}$, where A_R and a_R are the major and minor axis of R, respectively
Ellipse area	$\pi A_E a_E$, where A_E and a_E are the major and minor axis of E, respectively
Hu Moments 1 to 7	Translation, rotation and scale invariant features
Haralick 1 to 13	Texture features

All 790 images went through all steps, resulting in a dataset with 1,155 objects, to which we will refer as the "Santos" dataset. Figures 3a and 3b show that the dataset is considerably unbalanced, and the minority class has very few examples.



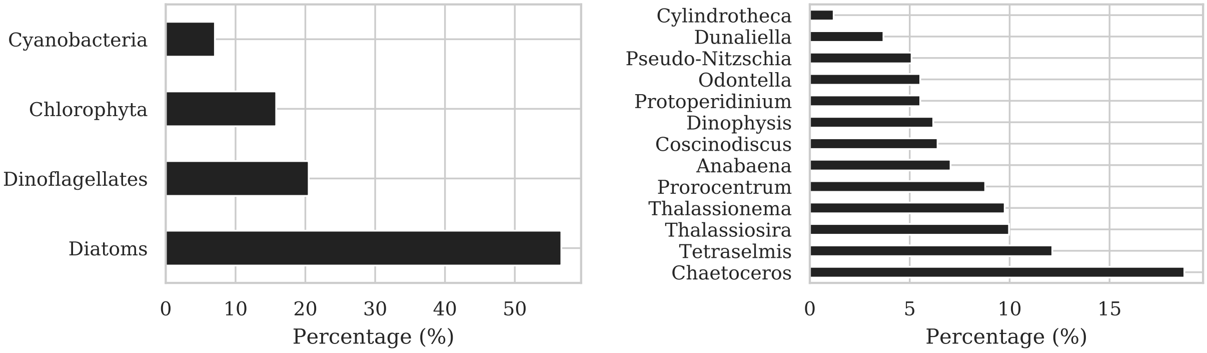
(a) Full image keypoints $\mathcal{K}_{full}$ (blue) and object contour C_{obj} (green)

(b) Object keypoints $\mathcal{K}_{obj}$ (blue) and convex hull of the keypoints H_k (red)

(c) Circumscribed ellipse E (red) and circumscribed rectangle R (blue)

(d) Object convex hull H_{obj} (yellow) and its bounding box B (magenta)

Figure 2: Visual representation of features.



(a) Generic class distribution.

(b) Specific class distribution.

Figure 3: Class distribution in the Santos dataset.

4. Data Assessment

To assess different classifiers on plankton classification, we consider the Santos dataset described in Section 3 as well as the dataset used in the Kaggle National Data Science Bowl Challenge [7]. This dataset was made public for the competition after it was collected by the Oregon State University's Hatfield Marine Science Center in the Straits of Florida using the In Situ Ichthyoplankton Imaging System (ISIIS) over an 18-day period. It consists of 30,336 grayscale images divided into 121 classes, and 130,400 images for performance evaluation of the competition. As the labels of the test dataset are not public, we consider only the training dataset. We discard all classes containing less than 100 images or classes for which our segmentation method identified less than 10 organisms, resulting in 21 classes. The main characteristics of each dataset are summarized on Table 2.

Table 2: Datasets main characteristics.

Dataset	Classes	Features	Objects
Santos-Generic	4	47	925
Santos-Specific	13	47	925
Kaggle	21	47	1744

Even though this dataset contains mostly ichthyoplankton, as opposed to phytoplankton, the used preprocessing methods should provide relevant information from any type of water-borne microorganisms, for which we wish to compare classifier performance.

We classify both datasets with 10-fold cross-validation, using the following classifiers: Naïve Bayes [36], kNN [36], Decision Tree (CART) [37], Logistic Regression [36], SVM [14] and Random Forest [38]. Random Forest and SVM's parameters were tuned with grid search, also with 10-fold cross-validation.

In the Santos dataset, for Random Forest, we considered 10, 50, 100, 500, and 1000 trees, and noted that past the 100-tree mark there was little to no difference in accuracy. For SVM, we considered $C \in \{1, 10, 100, 200\}$, and the linear, polynomial and RBF kernels, for which we obtained, with cross-validation, the best results with RBF and $C = 10$.

In the Kaggle dataset, we considered the same options for number of trees, C and kernels. The best results with cross-validation were obtained with 500 trees and $C = 10$ with RBF as kernel.

The average accuracy and F-measure obtained on each dataset are presented on Table 3. Random Forest and SVM outperformed all other classifiers, being very close to each other in all datasets. The performance difference between Generic and Specific classes was expected, as there are more classes, even though they contain the same objects and features. In the Kaggle dataset however, aside from the greater number of classes, the images are of a poorer quality.

Table 3: Average Accuracy and F-measure on 10-fold cross-validation in each dataset. Bold values indicate the best obtained for each dataset.

		Accuracy			F-Measure	
Classifier	Generic (4)	Specific (13)	Kaggle (21)	Generic (4)	Specific (13)	Kaggle (21)
Naïve Bayes	66.41 ± 11.88	64.55 ± 6.33	50.72±2.38	0.69 ± 0.09	0.60 ± 0.06	0.35±0.04
1NN	79.85 ± 13.63	68.50 ± 4.78	56.46±3.23	0.79 ± 0.12	0.63 ± 0.06	0.37±0.03
3NN	80.83 ± 13.92	69.10 ± 5.22	57.79±3.44	0.79 ± 0.12	0.61 ± 0.05	0.35±0.05
5NN	81.91 ± 13.03	72.02 ± 4.49	60.00±3.04	0.81 ± 0.11	0.64 ± 0.05	0.37±0.03
Log. Regression	79.50 ± 13.35	74.63 ± 3.07	70.27±2.14	0.78 ± 0.10	0.69 ± 0.06	0.51±0.03
Decision Tree	77.47 ± 15.01	66.71 ± 6.46	53.21±2.14	0.74 ± 0.13	0.62 ± 0.06	0.35±0.02
SVM	**85.42 ± 12.81**	76.24 ± 4.97	**68.64±2.48**	**0.84 ± 0.12**	0.71 ± 0.08	**0.52±0.04**
Random Forest	84.16 ± 13.84	**77.21 ± 4.35**	67.38±2.84	0.83 ± 0.12	**0.72 ± 0.08**	0.47±0.05

In order to better understand our problem, we assessed class similarity among the 13 Specific classes. This analysis is performed through the Random Forest, which was the best algorithm in this problem. Two objects are said to be similar if they end up on the same leaf on many trees of the Random Forest. Then, we can compute the average similarity between two classes by averaging the similarity between all pairs of objects of said two classes. The results of this calculation are shown in Figure 4.

It is noticeable that we have some pairs of classes with high similarities. Through visual inspection of the image data, we can see that these are classes that are indeed similar, and may be challenging to classify even through the eyes of an expert. A few examples of such images are shown in Figure 5 (5a vs 5b and 5c vs 5d).

To provide insight to which features most discriminate the presented classes in the Santos dataset, we performed two different feature analysis. We first analyze feature importance according to the Random Forest, which achieved the best results in the Santos dataset. We trained a Random Forest on the complete dataset and calculated the importance of each feature. The results of this analysis in the Santos dataset are shown in Table 4. We performed the same analysis on the Kaggle dataset. The results, however, were inconclusive, as the classifier had few features and all had very similar (low) importance values.

Overall, Haralick feature were very informative as they make up 4 of the 6 best features in both bases. Then we have features based on keypoints, such as the mean and standard deviation of the distances between them, or the number of keypoints. It can be noted that no distance between keypoints was used per se, only

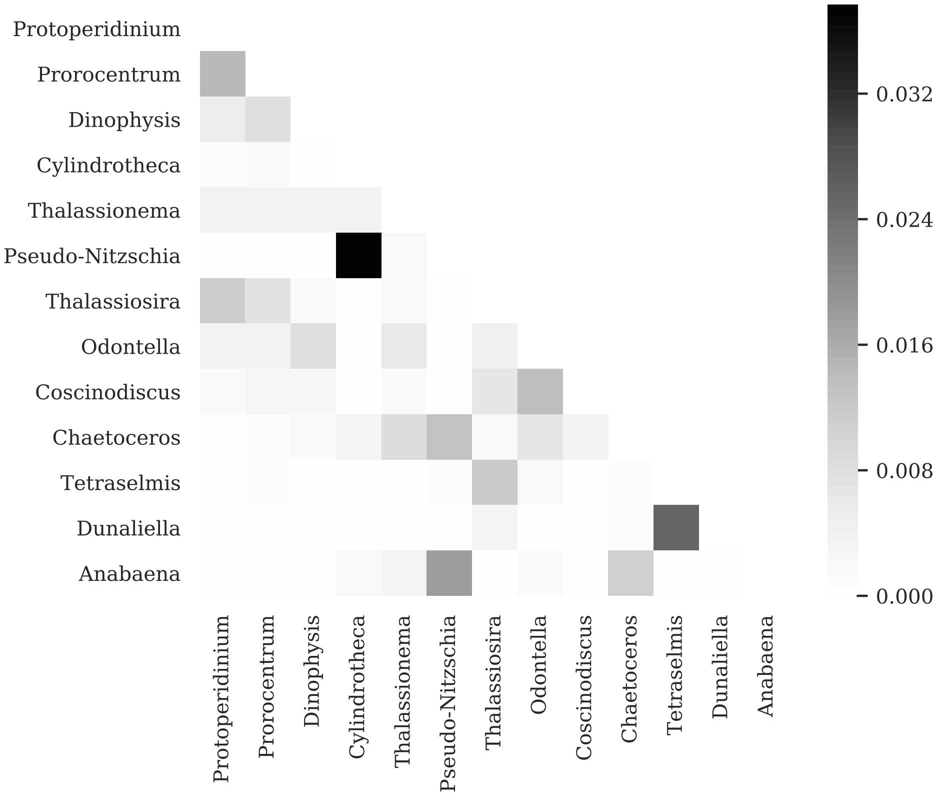
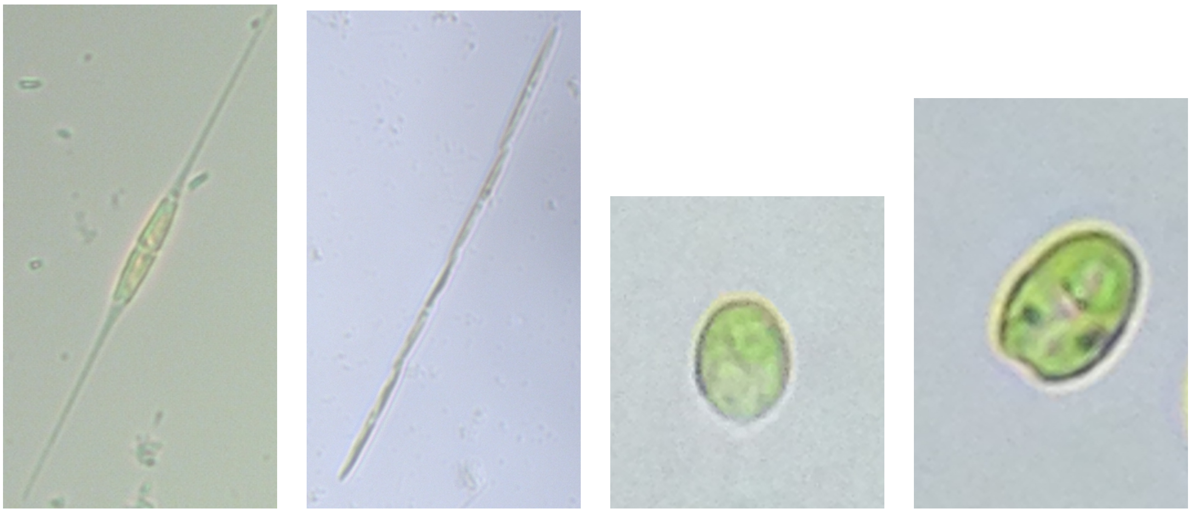


Figure 4: Class proximity between Specific classes.



(a) *Cylindrotheca* (b) *Pseudo-Nitzschia* (c) *Dunaliella* (d) *Tetraselmis*

Figure 5: Examples of images that may cause confusion.

their summarization, which shows that it was effective. Only then do we have some shape features, such as Average Rectangle Intensity and Extension. Thus, the analysis suggests that texture features, in general,

Table 4: Top 20 features by importance in the Generic dataset and their rank in the Specific dataset.

Feature	Generic	Specific
Haralick 4	1	2
Haralick 11	2	3
Distance Std.	3	5
Haralick 2	4	10
Haralick 12	5	6
Rectangle mean intensity	6	32
Keypoint Hull Area	7	7
Extent	8	8
Mean Distance	9	11
Full keypoint count	10	15
Haralick 5	11	19
Haralick 10	12	9
Haralick 1	13	16
Solidity	14	13
Haralick 9	15	27
Hu 0	16	21
Hu 1	17	14
Eccentricity	18	1
Aspect Ratio	19	4
Keypoint Count	20	12

are more informative than shape features. Some exceptions would be the Aspect Ratio, one of the simplest features in the dataset, and Eccentricity, as both appear among the 4 most important for Specific classes, although they have little use for Generic classes. Only the first two Hu Moments appear in the list, and only in the lower rankings.

We also employed a Naïve Bayes Wrapper Feature Subset Selection to find out which features are most relevant. We take a *backwards-search* approach [36], as we start with all features and remove them one by one until cross-validation accuracy is worsened.

For the generic classes, the Distances 1 to 4, Haralick 3, Area, Convex Hull Area, Perimeter, Compactness and Heywood Circularity were discarded, obtaining an accuracy of 75.12%, considerably higher than that achieved with all features. (Table 3). For the specific classes, only Distance 1 and Hu 0 were removed, reaching an accuracy of 65.72%. In the Kaggle dataset, this method kept 21 features with an overall accuracy of 57.76%, an improvement of 7%. The features it kept were the following: Distance 01, Distance 04, Distance Std, Keypoint Hull Area, Rectangle Mean, Ellipse Mean, Aspect Ratio, Circularity, Waddel Circularity, Rectangularity, Eccentricity, Ellipse Area, Convexity3, Hu 1, Hu 3, Hu 4, Hu 5, Haralick 1, Haralick 2, Haralick 3, Haralick 11.

Note that most *raw* distances have been removed, in accordance to the Random Forest analysis which indicates that summarizations are more effective than distances themselves. Also note that the Area and Hull Area were discarded, possibly because of redundancy when having features such as Convexity 1, which is the ratio between the two. The same goes for Perimeter, Compactness and Circularity, which can be similarly represented by other features. It can also be noted that the only discarded texture feature was

Haralick 3, which reinforces the observation that texture features are generally more important.

5. Open Source Tool

As described, the main objective of this study is to provide software for automatic plankton classification via image. The open-source tool we created is publicly available[1] under MIT license.

Its main purpose is to find and label plankton on a given image. The tool classifies the organism according to previously trained models. For convenience, we have already provided models that have been trained on the Santos dataset for this paper, but new models that best meet the needs of each application can be trained. The user can interact with the tool in two different ways: using a graphical interface (GUI) or with a web server. This section provides more details about the tool.

5.1. Training new models

The tool considers the same classifiers used in this paper: Naïve Bayes, *k*NN, Decision Tree (CART), Logistic Regression, SVM and Random Forest. The parameters of the Random Forest and SVM are the ones chosen by grid-search. Models that have been trained on the complete Santos dataset are provided with the tool to be used as desired.

However, each application is expected to have its own problem with different classes and therefore need different classifiers from ours. New models can be trained by selecting images whose labels are known and organizing them into directories according to those labels. Then, by executing a build command the software will identify plankton in the images and train different classifiers to replace the ones provided. The training process is the same described in this paper in Figure 1, with the same image preprocessing and attribute extraction techniques.

5.2. Interacting with the tool

Once the templates have been trained, using the software is simple. The first way to use it is through the GUI shown in Figure 6a. In this version, the classifier runs locally on each user's machine, and Python must be installed along with its dependencies. This version uses the Generic classes, with the SVM classifier, which our experiments show to be the most accurate.

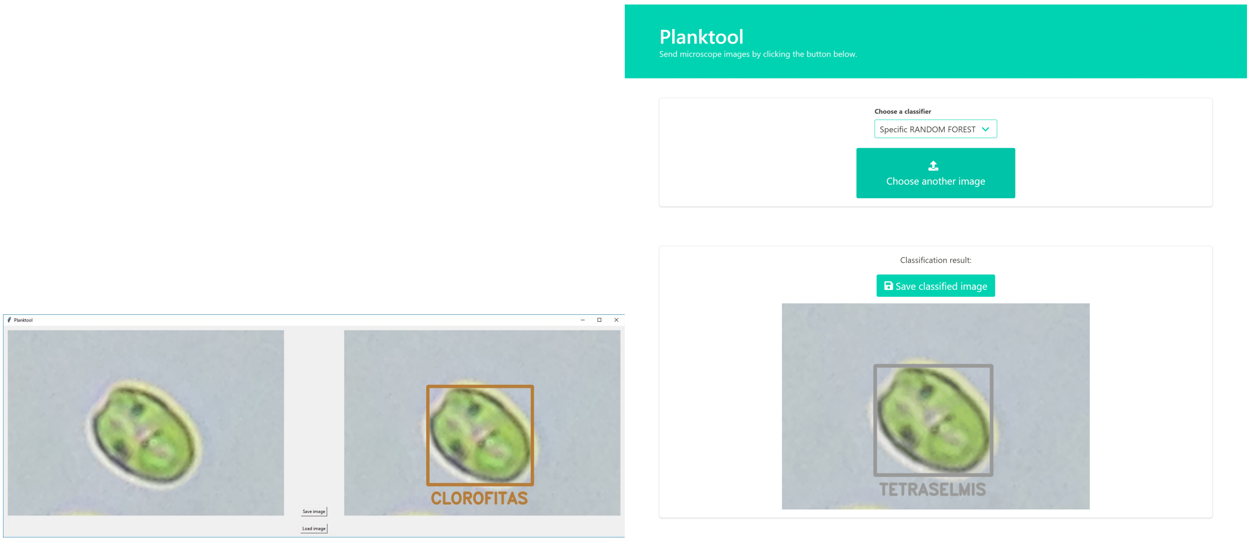
The second version is a web application, which starts a server that can receive and process images. A front-end interface (Figure 6b) is exposed to allow users to run classifiers over LAN or intranet, for example, without installing any additional end-user dependencies. Also, in this version, users can choose from all trained classifiers and see how their results differ.

6. Conclusion

This paper proposed an automatic image-based phytoplankton classification tool. We also presented a comparative study of different classifiers on the problem. After acquiring microscope phytoplankton images and manually classifying them, we were able to create an image set to perform classification analysis. We have employed computer vision techniques to identify cells in microscope images, and extract relevant yet simple features from their shape and texture. We then apply traditional classifiers to predict the phytoplankton class according to two different levels of specificity. We also show how our methods perform on the related problem of ichthyoplankton classification, and the results are consistent with the literature.

As future work, we intend to analyze different segmentation approaches, as well as assessing the performance of Active Learning in this problem. Also, for the specific classes of our dataset a hierarchical combination of classifiers is a promising future work.

[1] `https://github.com/mcf1110/planktool`



Choose a classifier
Specific RANDOM FOREST

(a) Graphical User Interface.

(b) Web Interface.

Figure 6: Examples of the developed interfaces.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

[1] H. W. Paerl, J. Dyble, P. H. Moisander, R. T. Noble, M. F. Piehler, J. L. Pinckney, T. F. Steppe, L. Twomey, L. M. Valdes, Microbial indicators of aquatic ecosystem change: current applications to eutrophication studies, FEMS Microbiology Ecology 46 (3) (2003) 233–246.

[2] S. H. Baek, S. Shimode, M.-S. Han, T. Kikuchi, Growth of dinoflagellates, ceratium furca and ceratium fusus in sagami bay, japan: the role of nutrients, Harmful algae 7 (6) (2008) 729–739.

[3] C. A. Heil, P. M. Glibert, C. Fan, Prorocentrum minimum (pavillard) schiller: a review of a harmful algal bloom species of growing worldwide importance, Harmful Algae 4 (3) (2005) 449–470.

[4] K. H. Mann, J. R. Lazier, Dynamics of marine ecosystems: biological-physical interactions in the oceans, John Wiley & Sons, 2013.

[5] A. Abonyi, M. Leitao, A. M. Lançon, J. Padisák, Phytoplankton functional groups as indicators of human impacts along the river loire (france), in: Phytoplankton responses to human impacts at different scales, Springer, 2012, pp. 233–249.

[6] R. K. Cowen, S. Sponaugle, K. Robinson, J. Luo, Planktonset 1.0: Plankton imagery data collected from f.g. walton smith in straits of florida from 2014–06-03 to 2014–06-06 and used in the 2015 national data science bowl (nodc accession 0127422), NOAA National Centers for Environmental Information (2015).

[7] Kaggle, National data science bowl.
URL `https://www.kaggle.com/c/datasciencebowl/data`

[8] J. Ellen, H. Li, M. D. Ohman, Quantifying california current plankton samples with efficient machine learning techniques, in: OCEANS 2015, 2015, pp. 1–9. `doi:10.23919/OCEANS.2015.7404607`.

[9] J. Dai, R. Wang, H. Zheng, G. Ji, X. Qiao, Zooplanktonet: Deep convolutional network for zooplankton classification, in: OCEANS 2016 - Shanghai, 2016, pp. 1–6. `doi:10.1109/OCEANSAP.2016.7485680`.

[10] R. Wang, J. Dai, H. Zheng, G. Ji, X. Qiao, Multi features combination for automated zooplankton classification, in: OCEANS 2016 - Shanghai, 2016, pp. 1–5. `doi:10.1109/OCEANSAP.2016.7485675`.

[11] C. Li, K. Wang, N. Xu, A survey for the applications of content-based microscopic image analysis in microorganism classification domains, Artificial Intelligence Review (2017) 1–70.

[12] M. G. Medeiros, Desenvolvimento de uma instrumentação de captura de imagens in situ para estudo da distribuição vertical do plâncton, Ph.D. thesis, Universidade de São Paulo (2017).

[13] V. P. Pastore, T. G. Zimmerman, S. K. Biswas, S. Bianco, Annotation-free learning of plankton for classification and anomaly detection, Scientific reports 10 (1) (2020) 1–15.

[14] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[15] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall PTR, 1998.

[16] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, `http://www.deeplearningbook.org`.

[17] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, arXiv preprint arXiv:1505.00853 (2015).

[18] X. Li, Z. Cui, Deep residual networks for plankton classification, in: OCEANS 2016, IEEE, 2016, pp. 1–4.

[19] J. Y. Luo, J.-O. Irisson, B. Graham, C. Guigand, A. Sarafraz, C. Mader, R. K. Cowen, Automated plankton image analysis using convolutional neural networks, Limnology and Oceanography: Methods 16 (12) (2018) 814–827.

[20] A. Lumini, L. Nanni, Ocean Ecosystems Plankton Classification, Springer International Publishing, Cham, 2019, pp. 261–280.

[21] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, Curran Associates Inc., 2012, pp. 1097–1105.

[22] H. Zheng, R. Wang, Z. Yu, N. Wang, Z. Gu, B. Zheng, Automatic plankton image classification combining multiple view features via multiple kernel learning, BMC bioinformatics 18 (16) (2017) 570.

[23] T. Luo, K. Kramer, D. B. Goldgof, L. O. Hall, S. Samson, A. Remsen, T. Hopkins, Active learning to recognize multiple types of plankton, Journal of Machine Learning Research 6 (Apr) (2005) 589–613.

[24] Python Core Team, Python: A dynamic, open source programming language, Python Software Foundation (2017).

[25] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools (2000).

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[27] N. Otsu, A threshold selection method from gray-level histograms, IEEE transactions on systems, man, and cybernetics 9 (1) (1979) 62–66.

[28] J. Canny, A computational approach to edge detection, IEEE Transactions on pattern analysis and machine intelligence (6) (1986) 679–698.

[29] R. C. Gonzalez, R. E. Woods, Processamento de imagens digitais, Edgard Blucher, 2000.

[30] S. Suzuki, et al., Topological structural analysis of digitized binary images by border following, Computer vision, graphics, and image processing 30 (1) (1985) 32–46.

[31] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: Computer Vision (ICCV), 2011 IEEE international conference on, IEEE, 2011, pp. 2564–2571.

[32] M. B. Blaschko, G. Holness, M. A. Mattar, D. Lisin, P. E. Utgoff, A. R. Hanson, H. Schultz, E. M. Riseman, M. E. Sieracki, W. M. Balch, et al., Automatic in situ identification of plankton, in: 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1, Vol. 1, IEEE, 2005, pp. 79–86.

[33] J. Zunic, P. L. Rosin, A new convexity measure for polygons, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (7) (2004) 923–934.

[34] M.-K. Hu, Visual pattern recognition by moment invariants, IRE transactions on information theory 8 (2) (1962) 179–187.

[35] R. M. Haralick, K. Shanmugam, I. H. Dinstein, Textural features for image classification, IEEE Transactions on systems, man, and cybernetics (6) (1973) 610–621.

[36] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison-Wesley, 2005.

[37] L. Breiman, J. Friedman, C. J. Stone, R. Olshen, Classification and Regression Trees, CRC Press, 1984.

[38] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.